Face Expression Transfer

Utku Saglam
TUM
utku.saglam@tum.de

Rachmadio Noval Lazuardi
TUM
rnl.lazuardi@tum.de

Ramandika Pranamulia
TUM
ramandika.pranamulia@tum.de

April 3, 2024

Abstract

We presented a method for facial expression transfer from a single source RGB image into a target RGB image in a photorealistic fashion. To realize this, two-fold optimizations on Basel Face Model (BFM) are utilized: sparse-term and dense-term. The sparse term refers to the optimization of BFM done by utilizing 68 facial landmarks. Specifically, the corresponding landmarks of the RGB image and the projected BFM are compared against each other to optimize for the shape and expression of the BFM. On the other hand, dense optimization is used to obtain the color information of the image into BFM. Finally, after the optimization is done on both source and target image, expression transfer is done by simply overwriting the expression weight from target BFM with source BFM.

1 Introduction

The face expression transfer has been worked on for a while, and there are good previous works that accomplish this very well. It has many use cases, such as creating facial expressions for fictional characters, dubbing, and virtual reality[11]. The [6] accomplishes face expression transfer in real-time. The [7] uses RGB image to perform Reenactment from source to target image.

In our work, we aimed to transfer facial expressions from one RGB image to another one in an offline setting. To achieve that, we followed the previous methods' footsteps and added updates to adapt them today. We first extracted the landmarks of the given images and used them as truth labels. Then, using BFM 2017, we got the model and its corresponding landmarks. As a third step, we applied sparse and dense optimizations that we talk about later on the align the face with the model in terms of spatial and color. Finally, we transferred the expression from one image to another.

2 Related Works

2.1 Facial Landmark Detection

An important initial step in the process of face expression transfer is detecting the facial landmark of the input image and target image. The locations of the fiducial facial landmark points around facial components and facial contour capture the rigid and non-rigid facial deformations due to head movements and facial expressions [9]. Numerous studies have focused on advancing the state-of-the-art in this area, employing various methodologies and technologies to enhance the precision and efficiency of landmark detection algorithms. We mainly focus to look for methods with high detection accuracy for this project since we aim for offline expression transfer not caring too much about the detection speed.

We consider some methods including the hour glass architecture [10], VGG net based architecture [1], and methods from survey papers[9]. Since we use Basel Face Model which has 68 landmarks in our intermediate step, we focus on methods outputting similar number of landmarks. For simplicity we use the ex-

isting trained network for detection and find a python library encapsulates most of those methods. Finally we choose Single Shot Scale-invariant Face Detector [12] resulting decent amount of accuracy for our needs and good reasonable speed.

2.2 Face Model

We use a face model in our steps to transfer the expression between the input and target faces. The face model is first used to mimic the input face and target face shape form by tuning the parameters. After having the desired face form, we can also learn the facial expression and trasfer them along from input to target face. We consider two face models in our project - the BFM and FLAME.

BFM or Basel Face Model is a parametric Face Model developed by university of Basel constructed by taken hundreds of faces as input. There are two versions of it, the first one is released in 2009 [5] and the later in 2017 [2]. The later one is richer in terms of age range and ethnic diversity. There is a mean model which depict a face with generalized features from those input faces which can be later tuned by adjusting the parameters to get the desired targeted face.

FLAME [4] itself is fundamentally similar to BFM but with more diverse data regarding races and expressiveness of input faces. It has also much lower dimension with only 100 dimension compared to 199 on BFM. The problem with Flame is that the tuning process of a face is way more complicated than BFM. In Flame, tuning process involves Advanced optimization techniques like non-linear solvers and gradient-based methods are commonly used. While in BFM it typically involves adjusting fewer parameters using linear methods like PCA fitting or least squares optimization. This makes it computationally less expensive. For that matter, we prefer BFM over FLAME in this project.

2.3 Face2Face

Face2Face [7] is a work in facial reenactment, enabling real-time manipulation of facial expressions in target videos using a live source actor. We consider some methods to be adapted to our project such as defining the optimization problem and seperating optimization problem into some different steps. The main difference between Face2Face and our project is that we don't operate on video as input and target. We are also focusing on offline reenactment as well.

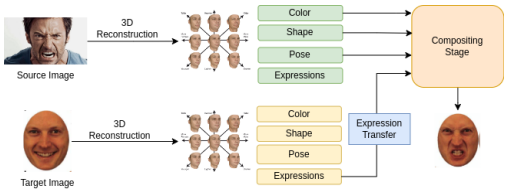


Figure 1: Overall Pipeline of Facial Expression Transfer.

3 Method

Our project's overall pipeline, depicted in Figure 1, involves two essential steps. Firstly, we must reconstruct the 3D face models from both the source and target RGB data. Once this reconstruction is optimized, we replace the expression parameter of the target model with that of the source model, effectively creating a composite 3D face model. The optimization process details are elaborated in the subsequent subsections. Section 3.1 delves into the 3D Morphable Model employed in our project, while Section **??** explains our approach to projection. Lastly, our optimization strategy is outlined in Section 3.2.

3.1 Synthesis of Facial Imagery

For our project, we utilized the Basel Face Model [3]. This model employs principal component analysis (PCA) to offer us mean shape, expression, and color data, along with basis functions and their associated standard deviations, facilitating the morphing of facial features. Specifically, denoting n for number of vertices, BFM models mean shape, expression, and color as a vector $\mu_{shape} \in R^{3n}$, $\mu_{exp} \in R^{3n}$, and $\mu_{color} \in R^{3n}$ respectively. Basis functions are defined as matrices of $E_{shape} \in R^{3n*199}$, $E_{exp} \in R^{3n*100}$, and

$E_{color} \in R^{3n*199}$ along with their corresponding variance $\sigma^2_{shape} \in R^{199}$, $\sigma^2_{exp} \in R^{100}$, and $\sigma^2_{color} \in R^{199}$. Hence, we parametrize face as:

$$\mathcal{F}_{shape} = \mu_{shape} + E_{shape} \times \sigma_{shape} \times \alpha \quad (1)$$

$$\mathcal{F}_{exp} = \mu_{exp} + E_{exp} \times \sigma_{exp} \times \beta \quad (2)$$

$$\mathcal{F}_{geo} = \mathcal{F}_{shape} + \mathcal{F}_{exp} \quad (3)$$

$$\mathcal{F}_{color} = \mu_{\text{color}} + E_{\text{color}} \times \gamma \quad (4)$$

where α, β, and γ are BFM parameters to be optimized. We used BFM'17 as our baseline. The model has 28K vertices and 56K faces.

3.2 Energy Formulation

We introduce two energy components for optimization, referred to as sparse and dense terms, inspired by the methodology outlined in [7]. The sparse term is employed for point-to-point optimization between landmarks identified in the input image and the projected BFM landmarks. Utilizing a sparse term solver, we optimize facial weights shape and expression (α, β) and camera parameters (R, t, and fov). Subsequently, we render the face using the optimized model and camera parameters. Then, the dense term is applied to perform point-to-point color optimization between the bilinearly interpolated BFM vertex on RGB image and the BFM color in order to optimize for the color weights (γ). Lastly, regularization term is applied to the face parameters to ensure the parameters remain close with the mean. Overall optimization pipeline is shown by figure **??**.

3.2.1 Sparse Energy Formulation

The goal of sparse optimization is to obtain camera parameters R, t, and fov, as well as BFM shape α and expression β weights. The optimization operates on 68 facial landmarks, instead of using all BFM vertices. We compute the L2-norm between the projected coordinates of landmarks derived from the BFM vertex v_j and the coordinates of the corresponding landmarks in the input image f_j, as shown by the equation 3.2.1

$$\sum_{j=1}^{68} \|\Pi(\phi(v_j)) - f_j\|_2^2 \quad (5)$$

Directly optimizing both camera and face parameters will make the optimization unstable. To alleviate this problem, the sparse optimization is divided into two steps. Firstly, camera parameters are optimized to ensure alignment between the BFM projection and the input image. Secondly, only the face parameters are optimized to gain the value of the shape and expression that matches with the image. As an initialization, we set rotation matrix as an identity matrix while the translation value is defined as $T = [0, 0, -400]$ with 60° field of view angle. During optimization, we found that the element of the rotation matrix $R[3, 3]$ often returns negative value. As a mitigation, we set a lower bound in the optimization process where $R[3, 3] \leq 1$.

3.2.2 Dense Energy Formulation

After camera parameters that aligns with the input image is obtained, we applied dense optimization to every BFM vertices to optimize for the face color weights γ. Such a process is done by first calculating both BFM vertex and BFM color as in equation 3 and 4 that corresponds to its vertex indices. For each of the BFM vertex, we find its corresponding color in the input image by using bilinear interpolation. Finally, the optimization is done by minimizing the L2-loss between the BFM color and the interpolated RGB color.

$$\sum_{j=0}^{vertices} \|\mathcal{F}_{color}(j) - I(j)\|_2^2 \quad (6)$$

3.2.3 Regularization

We additionally incorporated regularization cost functions to ensure that the parameters remain statistically aligned with the mean. These regulated

parameters encompass expression, shape, and color weights, each assigned distinct weights for regularization. Regularization is done by multiplying the corresponding weights with a square root of the regularization weights.

4 Results

4.1 Face Reconstruction

Results from our reconstruction pipeline can be seen in figure 2. Our pipeline successfully captured the images' shape, expression, and color of different faces. We see the original faces of different images in the first column. We tried to capture four different expressions. The first is neutral, the second is angry, the third one is shocked, and the last one is slightly happy.

In the second column, we see the reconstructions of the faces that we get by using our sparse and dense optimizations. Our pipeline can capture the expression very well in almost all the images. On the second one, Hugh Jackman's [8] expressions took much work to catch. However, our method also worked on that well and captured the expression closely. There is still room for improvement because of the facial wrinkles. We wanted to capture the shocked and slightly happy expressions on the third and last. Both of them were captured successfully.

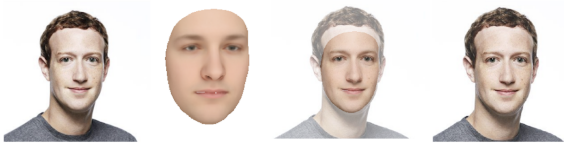
For the third column, we wanted to illustrate our reconstructed face on the transparent background image to show our geometric optimization result. We captured the geometric shapes for all images, and our reconstructed faces fit the images well without any problem.

The last column illustrates our color optimization result. The result is without discarding the estimated BFM color parameters. As we can see from the images, the colors of the original faces and the reconstructed faces are the same. The background seems slightly off because we still use a tiny amount of transparency to illustrate.

(a) First Reconstruction

(b) Second Reconstruction

(c) Third Reconstruction



(d) Fourth Reconstruction

Figure 2: Reconstruction Results

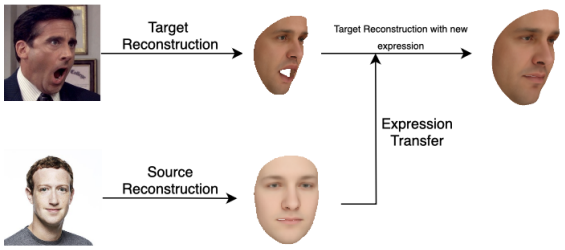
4.2 Expression Transfer

In figure 3, we illustrated our expression transfer results. We wanted to try a couple of combinations to see our overall performance.

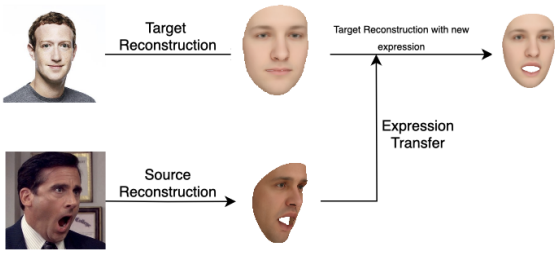
We elaborated the results of facial expression transfer in this section. In the first transfer, we transferred the expression from a slightly happy face to a shocked one, and as a result, we transferred the expression very well. Even tiny wrinkles on the slightly happy face transferred to the shocked face.

In the second one, we wanted to perform the reverse of the first expression transfer, which means transferring the expression from a shocked face to a slightly happy face. As a final result, we transformed the target's facial expression to shock.

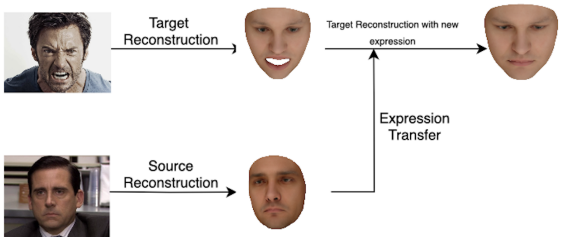
In the third one, we wanted to transform the expression of the target image from anger to a neutral expression. After we successfully reconstructed both images. We transformed the expression, and as can be seen in the figure, we made the model expression neutral from the angry one.



(a) First Transfer



(b) Second Transfer



(c) Third Transfer

Figure 3: Expression Transfer Results

5 Analysis

The face formation and expression transfer have been successfully accomplished. Unfortunately, our current implementation is still far from a real-time application. Given the input and target images, the process takes a couple of minutes to complete. The duration for each step can be seen in Table 1 below.

Steps	Time Taken
Landmark detection	17.2 s
Optimize $[R\|t]$	57 s
Calculate perspective projection	22 ms
Optimize shape + exp weight	1.3 s
Optimize color weight	132.1 s
Total construct + transfer time	526 s

Table 1: Time Taken for Various Processes

As we can see from the table above, the optimization parts, consisting of $[R|t]$ optimization and color weight, are steps with the longest running duration, summing up to a total of 189 seconds. This is followed by landmark detection with a run time of 17.2 seconds. Additionally, if we consider the total time for face construction + expression transfer, meaning from the beginning of the program execution until termination, it requires 526 seconds to complete. This is because many I/O executions take place in between the processes mentioned above. For example, the landmark detection happens in Python code, and we write the landmark detection output to a file for our C++ program to read later on.

6 Conclusion

We presented a method for facial expression transfer from a single source RGB image into a target RGB image in a photorealistic fashion. Following [7], we employed sparse and dense-optimization process to optimize for camera and face parameters, with regularization to control the behavior of the expression, shape, and color weights. We demonstrated that our work could capture the images' shape, expression, and color on different kind of faces, expressions, and orientation.

References

[1] Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. Style aggregated network for facial landmark detection. In *Proceedings of the IEEE conference on com-*

puter vision and pattern recognition, pages 379–388, 2018. 1

[2] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter. Morphable face models-an open framework. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 75–82. IEEE, 2018. 2

[3] IEEE. *A 3D Face Model for Pose and Illumination Invariant Face Recognition*, Genova, Italy, 2009. 2

[4] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 2

[5] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pages 296–301. Ieee, 2009. 2

[6] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. Real-time expression transfer for facial reenactment. *ACM Transactions on Graphics (TOG)*, 34(6), 2015. 1

[7] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016. 1, 2, 3, 5

[8] Wikipedia contributors. Hugh jackman — Wikipedia, the free encyclopedia, 2024. [Online; accessed 12-February-2024]. 4

[9] Yue Wu and Qiang Ji. Facial landmark detection: A literature survey. *International Journal of Computer Vision*, 127:115–142, 2019. 1

[10] Jing Yang, Qingshan Liu, and Kaihua Zhang. Stacked hourglass network for robust facial landmark localisation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 79–87, 2017. 1

[11] Juyong Zhang, Keyu Chen, and Jianmin Zheng. Facial expression retargeting from human to avatar made easy. *CoRR*, abs/2008.05110, 2020. 1

[12] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. S3fd: Single shot scale-invariant face detector. In *Proceedings of the IEEE international conference on computer vision*, pages 192–201, 2017. 2