©Copyright 2021
Bindita Chaudhuri

Deep Facial Expression Modeling and 3D Motion Retargeting from 2D Images

Bindita Chaudhuri

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2021

Reading Committee:

Linda Shapiro, Chair

Alex Colburn

Adriana Schulz

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Deep Facial Expression Modeling
and 3D Motion Retargeting from 2D Images

Bindita Chaudhuri

Chair of the Supervisory Committee:
Professor Linda Shapiro
Paul G. Allen School of Computer Science and Engineering

Facial expression modeling and motion retargeting, which involves estimating the 3D motion of a human face from a 2D image and transferring it to a 3D character, is an important problem in both computer graphics and computer vision. Traditional methods fit a 3D morphable model (3DMM) to the face, which requires an additional face detection step, does not ensure perceptual validity of the retargeted expression, and has limited modeling capacity (hence fails to generalize well to in-the-wild data). In this thesis, I present five deep learning based approaches to overcome these limitations: (1) a supervised network to jointly predict the bounding box locations and 3DMM parameters for multiple faces in a 2D image, (2) a self-supervised framework to jointly learn a personalized face model per user and per-frame facial motion parameters from in-the-wild videos of user expressions, (3) a multimodal approach that leverages both audio and video information to create a 4D facial avatar using dynamic neural radiance fields, (4) a semi-supervised multi-stage system that leverages a database of hand-animated character expressions to predict a character's rig parameters from a user's facial expressions, and (5) an unsupervised cycle-consistent generative adversarial network to directly predict the character's 3D geometry with retargeted expression. Experimental results have shown that these approaches outperform state-of-the-art methods in terms of retargeting accuracy. Applications of these approaches include avatar animation for visual storytelling or virtual conversation, motion capture films, and social AR/VR experience.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

Firstly, I would like to thank my advisor and committee chair, Prof. Linda Shapiro, for her immense support and valuable guidance throughout the Ph.D. years, both in professional and personal matters. I would also like to thank my co-advisor, Dr. Alex Colburn, for guiding me in my work with his expertise. Thanks to Prof. Adriana Schulz and Prof. John Kramlich for being a part of my graduation supervisory committee.

A part of this work was done during my internship at Microsoft (Redmond, WA) in collaboration with Dr. Baoyuan Wang and Sol Vesdapunt. I would like to express my sincere gratitude towards them as well as my other colleagues (Zeyu Chen, Muscle Wu, Pai Zhang, Xiang Yan, Wenbin Zhu) for their constant help with anything I needed. I would also like to thank my managers during my other internships (Dr. Oscar Nestares at Intel Labs and Dr. Nikolaos Sarafianos and Dr. Tony Tung at Facebook Reality Labs) for providing valuable technical insights that I could apply to my work in general.

Special thanks to Barbara Mones, Gary Faigin, Deepali Aneja and all other past and present members of the Facial Expression Research Group (FERG) for the valuable discussions. I am also grateful to Sachin Mehta, Nicholas Nuechterlein, Beibin Li, Shima Nofallah, Ezgi Mercan and other past and present members of the Multimedia research group for providing me a healthy environment to thrive. Last but not the least, thanks to my parents and my partner for being by my side through thick and thin and providing me the emotional support.

Chapter 1

INTRODUCTION

With the ubiquity of mobile phones, AR/VR headsets and video games, facial gestures have become an effective medium of non-verbal communication and it becomes more appealing when communication takes place through 3D animated characters. This has led to extensive research [15, 4, 58] in developing techniques to retarget human facial motion to 3D animated characters. Extracting 3D face information from monocular images is an ill-posed problem, so a typical solution is to leverage a parametric 3D morphable model (3DMM) [9] trained on a limited number of 3D face scans as prior knowledge [9, 108, 146, 84, 119, 48, 142, 32, 77]. A 3DMM represents the 3D face of a user as a linear combination of user-specific blendshapes. Retargeting involves predicting the blendshape weights and head pose that can fit the 3DMM to the input face image and then directly mapping the predicted weights to semantically equivalent blendshapes of the target 3D character model. However, the low-dimensional space of a 3DMM limits its modeling capacity as shown in [140, 145, 65] and scalability using more 3D scans is expensive. Similarly, the texture model of a generic 3DMM is learned in a controlled environment and does not generalize well to in-the-wild images. Tran et al. [145, 144] overcomes these limitations by learning a non-linear 3DMM from a large corpus of in-the-wild images. Nevertheless, these reconstruction-based approaches do not easily support facial motion retargeting.

Previous methods [15] formulate 3DMM fitting as an optimization problem and generally use decision trees or equivalent regression techniques to estimate the 3DMM parameters from the input image. However, these methods require significant manual effort because of pre-processing in terms of acquiring user-specific blendshapes, learning animation priors etc. or post-processing in terms of constraining the raw output. In addition, these methods use a two-step regression technique to predict the rigid deformation (head pose) and non-rigid deformation (facial expression)

and iterate over these two steps multiple times to achieve accurate results. With the advent of deep learning, recent works have shown remarkable accuracy by using deep convolutional neural networks to regress the 3DMM parameters from a 2D image. However, while 3DMM fitting with deep learning is frequently used in the state-of-the-art methods in related domains like 2D face alignment [179, 12], 3D face reconstruction [117, 42, 68, 24] etc., it is not yet a popular approach for facial motion retargeting. This is because (1) face alignment methods focus more on accurate facial landmark localization, while face reconstruction methods focus more on accurate 3D shape and texture reconstruction and on capturing the fine geometric details. In contrast, facial retargeting to an arbitrary 3D character only requires accurate transfer of facial expression and head pose. However, due to the ambiguous nature of this ill-posed problem, both facial expression and head pose are generally *sub-optimal* as they are not well decoupled from other information like identity. (2) Unlike alignment and reconstruction, retargeting often requires real-time tracking and transfer of the facial motion. However, existing methods for alignment and reconstruction are highly memory intensive and often involve complex rendering of the 3D model as intermediate steps, thereby making these methods difficult to deploy on light-weight hardware like mobile phones. Therefore, it is much desired to design a new 3DMM fitting system that is tailored specifically for face retargeting applications.

The importance of believable and accurate animated character facial expressions is readily demonstrated by films and games such as Polar Express [168] and Mass Effect: Andromeda [2]. In these examples, it is difficult for the audience to connect to the characters and broader storyline, because the characters do not exhibit clearly recognizable facial expressions that are consistent with their emotional state in the storyline [135, 109]. Animator-created character expressions can be expressive and clear but require expertise and hours of work. In order to speed up the animation process, animators often use human actors to control and animate a 3D stylized character using a facial performance capture system. These systems often lack the expressive quality and perceptual validity of animator-created animations, mainly due to their assumption that geometric markers are sufficient for expression transfer. The geometry-based methods and retargeting [95] based on hand-crafted descriptors may be unable to take into account the perception of the intended expression

when transferred onto a stylized character. We are unaware of any tools or methods that support animators by validating the perception of character expressions during creation. Despite recent advances in modeling capabilities, motion capture and control parameterization, current methods do not address the fundamental problem of creating clear expressions that humans recognize as the intended expression.

In this work, we present several semi-supervised and unsupervised approaches to overcome the limitations stated above. Our approaches can be grouped into three categories: 1) *blendshape-based unsupervised approaches*, in which retargeting takes place by representing human expressions in terms of character blendshapes and their weights, 2) *multimodal approaches*, in which different modalities (like video, audio and gaze direction) contribute to extracting the 3D face information required for retargeting, and 3) *example-based semi-supervised approaches*, in which retargeting takes place by learning human expression characteristics guided by character expression examples. Blendshape-based approaches generalize well to in-the-wild human expressions and can be easily applied to multiple new 3D characters (only semantically similar blendshapes as the 3DMM are required). On the other hand, example-based approaches generate more perceptually valid and geometrically consistent character expressions because of the additional guidance from hand-animated training examples. Multimodal approaches leverage the complementary information from multiple modalities for more accurate 3D facial animation.

1.1 Contributions and Thesis Organization

Our proposed blendshape-based (1-3) and example-based approaches (4-5) are as follows:

1. Chapter 3 presents a single end-to-end network to jointly predict the bounding box locations and 3DMM parameters for multiple faces. First, we design a novel multitask learning framework that learns a disentangled representation of 3DMM parameters for a single face. Then, we leverage the trained single face model to generate ground truth 3DMM parameters for multiple faces to train another network that performs joint face detection and motion retargeting for images with multiple faces.

2. Chapter 4 overcomes the limited modeling capacity of 3DMM by using an end-to-end framework that jointly learns a personalized face model (geometry and reflectance) per user and per-frame facial motion parameters (expression coefficients, head pose and scene illumination) from a large corpus of in-the-wild videos of user expressions. Specifically, we learn user-specific expression blendshapes and dynamic (expression-specific) albedo maps by predicting personalized corrections on top of a 3DMM prior.

3. Chapter 5 improves the face model personalization by leveraging 3D face information from the user video as well as dynamic lip motion from the user's speech. We design two parallel autoencoders - one for the input video frame and the other for the input speech segment at any given time frame. These encoders encode the facial motion into embeddings, which are then combined using learned weights and used as conditional input to drive dynamic neural radiance fields for 4D avatar reconstruction.

4. Chapter 6 introduces a multi-stage deep learning system that utilizes human and character expression recognition network features to learn a mapping from input human images to output 3D character rig parameters and also generalizes to multiple 3D characters including non-human characters.

5. Chapter 7 presents a unified framework that leverages cycle-consistent generative adversarial networks and position maps to learn the mapping from input human images to output 3D character vertex coordinates (geometry).

Finally, Chapter 8 concludes the thesis by summarizing the main contributions and proposing future research directions.

Chapter 2

RELATED WORK

2.1 Face Modeling

Methods like [151, 61, 91, 123, 81, 86, 143] leverage user images captured with varying parameters (e.g. multiple viewpoints, expressions etc.) at least during training to create a user-specific face model. Monocular video-based optimization techniques for 3D face reconstruction [47, 48] leverage the multi-frame consistency to learn the facial details. For single image based reconstruction, traditional methods [180] regress the parameters of a 3DMM and then learn corrective displacement [61, 69, 54] or normal maps [121, 118] to capture the missing details. Recently, several deep learning based approaches have attempted to overcome the limited representation power of 3DMM. Tran et al. [145, 144] proposed to train a deep neural network as a non-linear 3DMM. Tewari et al. [140] proposed to learn shape and reflectance correctives on top of the linear 3DMM. In [138], Tewari et al. learn new identity and appearance models from videos. However, these methods use expression blendshapes obtained by deformation transfer [131] from a generic 3DMM to their own face model and do not optimize the blendshapes based on the user's identity. In addition, these methods predict a single static albedo map to represent the face texture, which fail to capture adequate facial details. Optimization based methods like [82, 63, 18] have demonstrated the need to optimize the expression blendshapes based on user-specific facial dynamics. These methods alternately update the blendshapes and the corresponding coefficients to accurately fit some example poses in the form of 3D scans or 2D images. For facial appearance, existing methods either use a generic texture model with linear or learned bases or use a GAN [49] to generate a static texture map. But different expressions result in different texture variations. Nagano et al. [102] and Olszewski et al. [104] addressed this issue by using a GAN to predict the expression-specific texture maps given the texture map in neutral pose. However, the texture variations with expres-

sion also vary from person to person. Hence, hallucinating an expression-specific texture map for a person by learning the expression dynamics of other persons is not ideal. Besides, these methods requires fitted geometry as a preprocessing step, thereby limiting the accuracy of the method by the accuracy of the geometry fitting mechanism.

2.2 2D Face Alignment and 3D Face Reconstruction

Early methods like [74] used a cascade of decision trees or other regressors to directly regress the facial landmark locations from a face image. Recently, the approach of regressing 3DMM parameters using CNNs and fitting the 3DMM to the 2D image has become popular. While Jourabloo et al. [70] use a cascade of CNNs to alternately regress the shape (identity and expression) and pose parameters, Zhu et al. [179, 178] perform multiple iterations of a single CNN to regress the shape and pose parameters together. These methods use large networks and require 3DMM in the network during testing, thereby requiring large memory and execution time. Regressing 3DMM parameters using CNNs is also popular in face reconstruction [143, 117, 54, 141]. Richardson et al. [118] uses a coarse-to-fine approach to capture fine details in addition to face geometry. However, reconstruction methods also regress texture and focus more on capturing fine geometric details. For joint face alignment and reconstruction, [42] regresses a position regression map from the image and [145] regresses the parameters of a nonlinear 3DMM using an unsupervised encoder-decoder network. For joint face detection and alignment, recent methods either use a mixture of trees [111] or a cascade of CNNs [169]. In [31], separate networks are trained to perform different tasks like proposing regions, classifying and regressing the bounding boxes from the regions, predicting the landmark locations in those regions etc. In [112], region proposals are first generated with a selective search algorithm, and bounding box and landmark locations are regressed for each proposal using a multitask learning network. In contrast, we propose a single end-to-end network to do joint face detection and 3DMM fitting for face retargeting purposes.

2.3 Performance-Based Facial Animation

Traditional performance capture systems (using either depth cameras or 3D scanners for direct mesh registration with depth data) [3, 10, 150] require a complex hardware setup that is not readily available. Among the methods which use 2D images as input, PCA-based linear modeling [29, 96] and the blendshape interpolation technique [15, 124] are most popular. However, these methods require dense correspondence of facial points [116] or user-specific adaptations [83, 16] to estimate the blendshape weights. Recent CNN-based approaches either require depth input [81, 55] or regress character-specific parameters with several constraints [11, 38]. The authors of [164] leveraged an expression recognition task for retargeting purposes, but the method is limited only to transferring expressions to 2D cutout animations.

Existing joint face tracking and retargeting methods [150, 10, 83] generally optimize the face model parameters with occasional correction of the expression blendshapes using depth scans. Recent deep learning based tracking frameworks like [142, 151, 22, 76] either use a generic face model and fix the model during tracking, or alternate between tracking and modeling until convergence. We propose to perform joint face modeling and tracking with novel constraints to disambiguate the tracking parameters from the model. Commercial marker-based and markerless facial motion capture software products like Faceshift [39], Faceware [40], Mixamo [100], Dynamixyz [35] and Optitrack [105] perform real-time retargeting but with poor expression accuracy [4]. The marker-based products map some predefined marker points on the human face to the corresponding points on the 3D character rigs enabling live tracking and retargeting, but the limited number of marker positions often fail to capture the intended expression. The markerless systems rely on blendshape-based retargeting. However, all these methods require a significant amount of manual effort in terms of setting up a new character, mapping the expressions from the existing character to a new one, and refining the generated expressions for accurate tracking.

2.4 Face Detection and Expression Recognition

In the literature of multiple object detection and classification, Fast RCNN [115] and YOLO [113] are the two most popular methods with state-of-the-art performance. While [115] uses a region proposal network to get candidate regions before classification, [113] performs joint object location regression and classification. Keypoint localization for multiple objects is popularly used for human pose estimation [78, 19] or object pose estimation [137]. In case of faces, landmark localization for multiple faces can be done in two approaches: *top-down approach* where landmark locations are detected after detecting face regions and *bottom-up approach* where the facial landmarks are initially predicted individually and then grouped together into face regions. In our method, we adopt the top-down approach.

Convolutional Neural Networks have shown great improvement in facial expression recognition tasks [101, 13, 88, 71, 34, 72] and there are a number of fusion algorithms to boost the recognition performance [161, 167, 133, 164]. Many systems are trained on a single facial expression database, which makes them sensitive to the lighting and particular poses present in that database. These methods focus on engineered features, which lack the generalizability to perform in-the-wild expression recognition. To overcome this limitation, we combine human databases from different sources including a dataset collected in the wild for our training step in order to improve the robustness of our trained model. In our work, we use CNNs to learn perceptual features pertaining to facial expressions and combine them with geometric features, since this approach has been shown to perform better in expression recognition and retargeting than geometric features alone [5]. Note that Aneja et al. [5] proposed a retrieval method to identify the closest 2D character expression image from the existing database to a given human image, whereas we propose a method that generates a 3D character expression for a given human image.

2.5 Audio-driven Animation

Audio-driven animation can be classified into two categories - a) methods which synthesize a 2D face video from audio input, and b) methods which animate a 3D face model based on audio.

Fan et al. [41] uses bidirectional LSTM to generate talking face frames. Suwajanakorn et al. [132] and Kumar et al. [80] both generated talking head for specifically Barack Obama using RNN with compositing techniques and time-delayed LSTM with Pix2Pix respectively. Jalalifar et al. [67] uses RNN with conditional GAN and Vougioukas et al. [147] uses Temporal GAN to synthesize talking faces. Chen Song et al. [130] uses conditional RNN to enforce accurate lip synchronization across video frames for talking face generation. Chen et al. [23] employs optical flow between frames to improve photo-realism in talking heads. Mittal et al. [99] proposed using disentangled audio representations based on variational autoencoders to generate talking heads. Recently, Zakharov et al. [166] suggested a method with few shot learning capability to generate a talking head model for an unseen image. A major challenge for these methods is to generate photo-realistic face images. Although many of these methods have been able to achieve high photo-realism, these approaches still do not exhibit adequately high temporal synchronization of audio and generated video. Some works [173, 99] have attempted to address the problem using disentangled representations but there is no approach that learns a shared representation across multiple modalities to decouple just the content information to drive the animation.

Karras et al. [73] performs speech-driven 3D facial animation by mapping the input waveforms to 3D vertex coordinates of a face model and simultaneously using an emotional state representation to disambiguate the variations in facial pose for a given audio. Yang et al. [174] introduces a deep learning based approach to map the audio features directly to the parameters of the JALI model [36]. Taylor et al. [136] uses a sliding window approach to animate a parametric face model from phoneme labels. Recently, [30] introduced a model called Voice Operated Character Animation (VOCA) which takes as input a speech segment in the form of its corresponding DeepSpeech [56] features and a one-hot encoding over training subjects to produce offsets for 3D face mesh for subject template registered using FLAME [84] model. Their approach is for 3D facial animation which allows altering speaking style, pose and shape, but cannot adapt completely to an unseen identity. Moreover, none of the previous methods utilize multimodal learning to disentangle the content from the audio-visual modalities.

2.6 Multimodal Learning

Several multimodal representation learning based approaches have been suggested over the years, an overview of which is presented in [6]. Some of them aim to propose sparse representations as encoding the mutual information [20, 75] for applications such as biometrics recognition [126] and audio/video reconstruction [103]. These approaches combine the signal coming from different modalities via late fusion to learn a shared representation that can encode and bring the two modalities close to each other in terms of variations. [57] suggests incorporating attention to fuse multiple modalities for video description, while [152] proposes a fusion based method for weakly-supervised learning. Multimodal approaches for face based applications such as facial animation include [89, 173]. However, these approaches have either limited themselves to the 2D image domain or have not fully utilized the visual information as part of the second modality. We propose to use audio-visual modality to correlate the content information from the two modalities and discard any identity specific information for the task of speaker-agnostic 3D face animation.

2.7 Neural Scene Representation

Recently neural scene representation networks have been widely used in neural rendering and reconstruction approaches, as given in [139]. While neural scene representation networks were first introduced by Sitzmann et al. [129], Mildenhall et al. [98] extend this idea to store radiance fields in a neural network (NERF), where they assume the availability of images of the 3D object or scene from multiple views. Follow-up work improves upon them by using different positional encodings [134] and in-the-wild training data including appearance interpolation [94]. Neural Sparse Voxel Fields [87] employ an Octree to cull empty space and speed up rendering. The limitation of these methods is the assumption of a static object, whereas 3D face in our case changes dynamically. In [45], the authors warp the input coordinates of NERF using 3DMM fitted rigid head pose and train a regular NERF to create a 3D portrait from input 2D image. Gafni et al. [44] have attempted to create a 4D reconstruction of a user given the user video frames as input, conditioned on the head pose and expression parameters. However, these methods utilize only one modality (video input).

Chapter 3

JOINT FACE DETECTION AND FACIAL MOTION RETARGETING

3.1 Introduction

Traditional methods for facial motion retargeting address the problem of fitting 3DMM to a single face in the input image. For multiple faces in a single images, a straightforward approach is to run a face detector on the image to detect the face regions and perform the desired operations on each face individually. However, this approach requires additional execution time for face detection and the computational complexity increases linearly with the number of faces in the input image. In the literature of joint face detection and alignment, existing methods either use a mixture of trees to predict the face bounding boxes and the facial landmarks or adopt an iterative two-step approach of generating region proposals and predicting the landmark locations in the proposed regions. These methods are only good for directly regressing the landmark locations and not 3DMM parameters, and they do not run in real time. Methods like Fast R-CNN [51] and YOLO [113] have shown remarkable results in joint object detection and classification while achieving real time performance. Similar methods have been applied to regress the 6D poses of multiple objects or humans in a single image, but to the best of our knowledge, no method has attempted to perform multiple 3DMM fitting to a single image using a single network.

To this end, we divide our work into two parts. In the first part, we propose a multitask learning framework to directly regress the 3DMM parameters from a 2D image of a single human face. The 3DMM parameters are grouped into: a) identity parameters that contain the face shape information, b) expression parameters that captures the facial expression, c) pose parameters that include the 3D rotation and 3D translation of the head and d) scale parameters that link the 3D face with the 2D image. We have observed that pose and scale parameters require global information while identity and expression parameters require local information, so we propose to emphasize the high level

image features for pose and scale and the low level image features for identity and expression. Our network architecture is designed such that different layers embed image features at different scales, and we use different combinations of these multi-scale features for each group of parameters. This also helps in disentangling the parameter groups from each other by allowing the network to learn different features for different groups. Finally, we add an optional second stage to our network that further refines the expression parameters from the first stage by training on the eye and mouth regions of the input image separately. Our Single Face Network (SFN) performs in real-time even on regular mobile devices and requires negligible memory while achieving state-of-the-art performance. Qualitative evaluation also shows that our method is robust to poses, illuminations and occlusions.

In the second part of our work, we propose a single end-to-end trainable network to jointly detect the face bounding boxes and regress the 3DMM parameters for multiple faces in a single image. Inspired by YOLO, we design our Multiple Face Network (MFN) architecture similar to YOLOv2 that takes a 2D image as input and predicts the centroid positions and dimensions of the face bounding boxes as well as the 3DMM parameters for the faces in those boxes. In human pose estimation problems, this approach is called a *top-down* approach of multiple keypoint prediction. However, existing publicly available image datasets with multiple faces only have ground truth for face bounding boxes and are generally used for face detection. We leverage our SFM to generate the ground truth for 3DMM parameters for each face box and then use both the ground truths to train our MFN. Experimental results show that our MFN not only performs well for multi-face retargeting but also improves the accuracy of face detection.

Our main contributions can be summarized as follows:

1. We present a novel top-down approach using an end-to-end trainable network to jointly learn the face bounding boxes and the 3DMM parameters from an image having multiple faces with different poses and expressions. The ground truth 3DMM parameters for multiple faces required to train our network is generated by a semi-supervised method.

2. We design a multitask learning framework that combines image features at different scales to

disentangle and accurately predict different groups of 3DMM parameters. We also design an optional second stage for our network to refine the expression parameters while decoupling the eye and mouth expression parameters.

3. Our system is easy to deploy into practical applications without requiring separate face detection for pose and expression retargeting. Our joint network can be run in real-time on mobile devices without engineering level optimization, e.g. only 39ms on Google Pixel 2.

3.2 Methodology

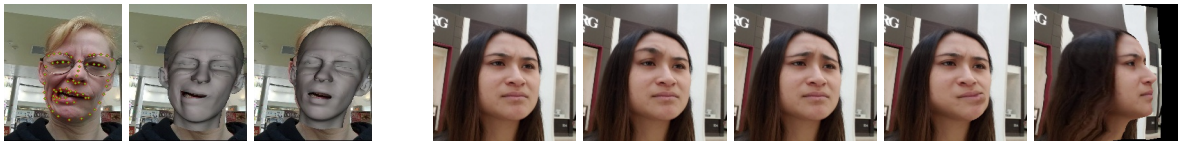
The 3D mesh of a human face can be represented by a multilinear 3D Morphable Model (3DMM) as

$$\mathcal{M} = \mathcal{V} \times \mathrm{b_{id}} \times \mathrm{b_{exp}} \tag{3.1}$$

where $\mathcal{V}$ is the mean neutral face, $\mathrm{b_{id}}$ are the identity bases and $\mathrm{b_{exp}}$ are the expression bases. We use the face tensor provided by FacewareHouse [17] as 3DMM, where $\mathcal{V} \in \mathbb{R}^{11510\times3}$ denotes $11,510$ 3D co-ordinates of the mesh vertices, $\mathrm{b_{id}}$ denotes 50 shape bases obtained by taking PCA over 150 identities and $\mathrm{b_{exp}}$ denotes 47 bases corresponding to 47 blendshapes (1 neutral and 46 micro-expressions). To reduce the computational complexity, we manually mark 68 vertices in $\mathcal{V}$ as the facial landmark points based on [111] and create a reduced face tensor $\hat{\mathcal{M}} \in \mathbb{R}^{204\times50\times47}$ for use in our networks. Given a set of identity parameters $w_{\mathrm{id}} \in \mathbb{R}^{50\times1}$, expression parameters $w_{\mathrm{exp}} \in \mathbb{R}^{47\times1}$, 3D rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$, 3D translation parameters $\mathbf{t} \in \mathbb{R}^{3\times1}$ and a scale parameter (focal length) f, we use weak perspective projection to get the 2D landmarks $\mathbf{P_{lm}} \in \mathbb{R}^{68\times2}$ as:

$$\mathbf{P_{lm}} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \end{bmatrix} [\mathbf{R} * (\hat{\mathcal{M}} * w_{\mathrm{id}} * w_{\mathrm{exp}}) + \mathbf{t}] \tag{3.2}$$

where $w_{\mathrm{exp}}[1] = 1 - \sum_{i=2}^{47} w_{\mathrm{exp}}[i]$ and $w_{\mathrm{exp}}[i] \in [0,1], i = 2,\ldots,47$. We use a unit quaternion $\mathbf{q} \in \mathbb{R}^{4\times1}$ [178] to represent 3D rotation and convert it into rotation matrix for use in equation 3.2. Please note that, for re-targeting purposes, we omit the learning of texture and lighting in the 3DMM fitting model.



(a) (b)

Figure 3.1: (a) Left: landmark projection from both meshes are exactly the same, middle: mesh with maximum jaw left, right: mesh without jaw left, but larger roll angle, (b) Synthesized images for regularization.

3.2.1 Multi-scale Representation Disentangling

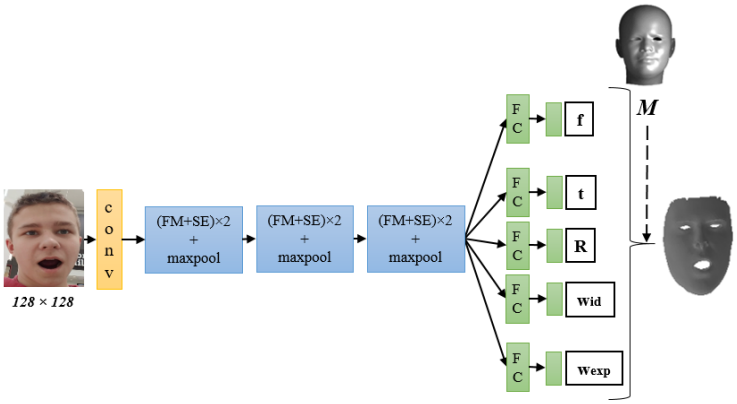
A straightforward way of learning all the 3DMM parameters would be to simply cast this as a regular regression network that holistically outputs all the parameters together through a fully connected layer on top of one shared representation. However, this will not be optimal particularly for our problem where each group of parameters has strong semantic meanings. Intuitively speaking, head pose learning does not require detailed local face representations since it is fundamentally independent of skin texture and subtle facial expressions, as observed in recent work on pose estimation [172]. However, for identity learning, a combination of both local and global representation would be necessary to discriminate among different persons. For example, it is common to see someone with relatively small eyes but fat cheek compared with others with big eyes and thin cheek, so both the local features around the eyes and the overall face silhouette would be important to approximate the identity shape. Similarly, face expressions learning possibly requires even fine-grained granularity of different scales of representations. Single eye winking, mouth grin and big laugh clearly require three different levels of representations to discriminate them from other expressions. Another observation is, given the 2D landmarks of an image, there exist multiple combinations of 3DMM parameters that can minimize the landmark loss. This ambiguity causes additional challenges to the learning to favor the semantically more meaningful combinations. Motivated by these observations, we designed a novel network structure that is specifically tailored for facial retargeting applications. For example, as shown in Fig. 3.1a, we can still minimize the 2D landmark loss by rotating the head and using different identity coefficients to accommodate the jaw

left even without a strong jaw left expression coefficient. Motivated by both the multi-scale prior and the ambiguity nature of this problem, we designed a novel network structure that is specifically tailored for facial retargeting applications as illustrated in Fig. 3.2b, where pose is only learned through the final global features, while expression learning depends on the concatenation of multi-layer representations.

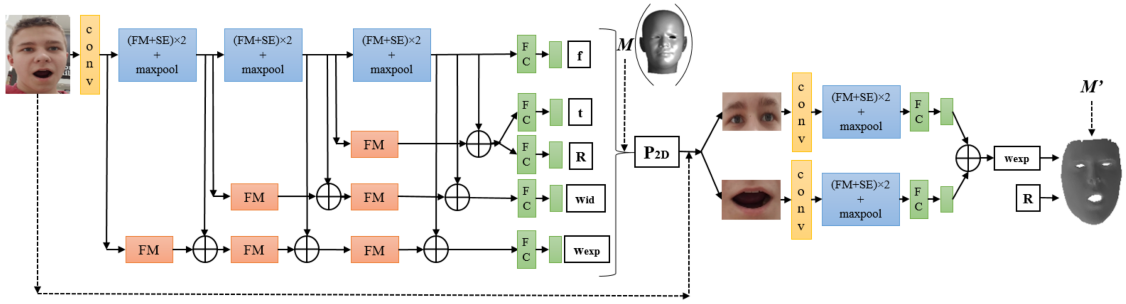
In addition to the above network design, we add a few regularization during the training to further enforce the disentangled representation learning. For example, for each face image, we can augment it by random translation/rotation perturbation to ask their resulting output to have the same identity and expression coefficients. Using image warping technique, we can re-edit the face image to slightly change the facial expression without hurting the pose and identity. Fig. 3.1b shows a few such synthesized examples where their identity parameters need to be the same.

3.2.2 Single Face Retargeting Network

When the face bounding box is given, we can train a single face retargeting network to output 3DMM parameters for each cropped face image using a multitask learning framework as shown in Fig. 3.2b. Fortunately, many public datasets [122, 17, 60, 179] already provide bounding boxes along with 68 facial landmark points. To reduce the ambiguity issue mentioned above, we first fit 3DMM parameters for each cropped single face image using existing optimization method [12], and then treat them as ground-truth into our network loss function by mean absolute error, in additional to the 2D landmark loss. As each so called "ground-truth" was optimized for each image locally by [12], the risk of over-fitting could make the "ground-truth" a little bit noisy. However, a deep neural network is trained by overseeing a large number of data that can intrinsically combat the noisy labelling issue to focus more on the global common patterns. Therefore, the training starts with a large weight on the L1 loss with respect to the "ground-truth", then gradually decays this weight to trust more on the 2D landmarks loss, as shown in the following loss function:



(a) Single scale SFN.



(b) Multi-scale SFN.

Figure 3.2: Our Single Face Network (SFN) architecture. FM denotes Fire Module [62], SE denotes squeeze-excite block [59] and FC denotes Fully Connected layer. Each convolution layer is followed by a batch normalization layer and a ReLU activation layer.

$$\tau * \left\{ \frac{1}{50} \sum_{i=1}^{50} |w_{\text{id}_i} - w_{\text{id}_i}^g| + \frac{1}{46} \sum_{i=1}^{46} |w_{\text{exp}_i} - w_{\text{exp}_i}^g| + \frac{1}{4} \sum_{i=1}^{4} |\mathbf{R}_i - \mathbf{R}_i^g| \right\} + \sqrt{\frac{1}{68} \sum_{i=1}^{68} (\mathbf{P}_{\text{lm}_i} - \mathbf{P}_{\text{lm}_i}^g)^2} \tag{3.3}$$

where τ denotes decay parameter with respect to epoch. We choose τ = 10/epoch across all experiments. g denotes ground truth obtained from [12]. Note that, although we drop the 3D translation and scale ground truth loss to allow 2D translation and scaling augmentation, the translation and scale parameters can still be learned by the 2D landmark loss.

We also add an optional second stage which is a light-weight extension that takes the eye and mouth regions of the image as input and refines the expression parameters predicted by the first stage. To get the input images for the second stage, we first generate the 2D landmarks using the predicted parameters in equation 3.2 and use the landmarks to crop the eye and the mouth regions from the input image. The resized eye and mouth images are then fed to two separate branches to predict the residual values of the eye and mouth expression parameters respectively. Among the 46 blendshapes in our 3DMM, the first 19 blendshapes are for the eyes and the remaining 27 are for the mouth. The predicted values in the second stage are then added to the predicted values in the first stage to get the final expression parameters. We initially train the first stage, use the trained weights of first stage to initialize the weights of the second stage, train the second stage keeping the weights of the first stage fixed, and finally fine-tune both the stages in an end-to-end manner. The loss function for the second stage is given by

$$L_1^{w_{\text{exp}}} + L_2^{\mathbf{p}_{\text{eye}}} + L_2^{\mathbf{p}_{\text{mouth}}} \tag{3.4}$$

where $\mathbf{p}_{\text{eye}} = \mathbf{p}[i,:], i = 37, \ldots, 48$ and $\mathbf{p}_{\text{mouth}} = \mathbf{p}[i,:], i = 49, \ldots, 68$ and L_1 and L_2 denote mean absolute error and mean square error respectively.

Our network takes 128x128 resized image as input. In the first layer, we use a 7×7 convolution layer with 64 filters and stride 2 followed by a 2×2 maxpooling layer to capture the fine details in the image. The following layers are made up of Fire Modules(FM) of SqueezeNet [62] (with 16 and 64 filters in squeeze and expand layers respectively) and squeeze-and-excite modules(SE) of [59] in order to compress the model size and reduce the model execution time without compromising the accuracy. At the end of the network, we use a global average pooling layer followed by fully connected (FC) layers to generate the parameters. The penultimate FC layers each has 64 units with ReLU activation and sigmoid activation is used at the end of the last expression branch's FC layer to restrict the values between 0 and 1. To realize the multiscale prior and the disentangled learning, we concatenate the features at different scales and form separate branches for each group of parameters. The extra branches are built with the same blocks as the main branch, but we reduce the channel size by half to restrict the extra computation cost. The input images in the second

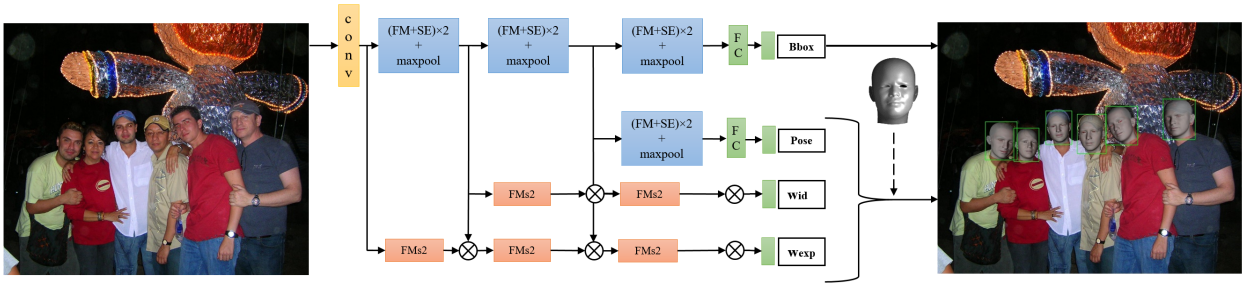


Figure 3.3: Our Multi Face Network (MFN) architecture. The building blocks are Fire Module (FM) [62] and squeeze-and-excitation (SE) [59] which are designed for real-time application. The multi-scale branch uses multiple slim FM with stride 2 on the last FM (FMs2) to allow concatenation. The multiplication of $Pose$, w_{id}, w_{exp} with 3DMM generates a 3D human mesh for every bounding box.

stage are both resized to 64×64 pixels. The architecture of our single scale single face retargeting network is shown in Fig. 3.2a. The resolution (scale) of the image feature maps is reduced by 2 after every block, and the pose, expression and identity parameters are learned from the same feature map, hence the term single scale. We designed our multi-scale single face retargeting network to learn different groups of parameters from separate branches that represent image features at different scales.

3.2.3 Joint Face Detection and Retargeting

As discussed before, it is less efficient to run face detection first and then run the single face retargeting network (see above Sec.3.2.2) individually for each face region, especially for images with multiple faces. This is because the computational complexity increase linearly with the number of faces, also when customized for the case of multiple face tracking, such straightforward approach quickly becomes complicated when people move in and out from the frame or occlude each other. Therefore, our goal is to save computation cost by predicting both tasks at the time. The network could potentially also benefit from the cross domain knowledge, especially for detection task, where introducing 3DMM gives the prior on how the face should look in 3D space, which is complementary to the 2D features in the separate face detection framework.

Inspired by YOLO [113], our joint network can be trained by extending the Single Face Retargeting Network to let it output face bounding boxes as well. Specifically, we ask the network to predict 3DMM parameters for each anchor point in additional to bounding box displacement and objectness. We divide the input image into 9×9 grid and predict a vector of length $4 + 1 + (50 + 46 + 4 + 3 + 1) = 109$ for a bounding box in each grid cell. Here, 4 denotes the 2D co-ordinates of the centroid, width and height of the face bounding box, 1 denotes the confidence score for the presence of a face in that cell and the rest are the 3DMM parameters for the face in the cell. We also adopt the method of starting with 5 anchor boxes as bounding box priors. The x, y co-ordinates of the centroid and the width and height of a bounding box are calculated in the same manner as in [113] and we use the same loss functions for these values.

Since there are no publicly available datasets that contain multiple faces in each image and provide their corresponding landmark annotations, for proof-of-concept, we obtain the 3DMM ground truth by running our single face retargeting network on each face separately and then train the joint network. Each bounding box b predicted by the network has the following co-ordinates: t_x, t_y, t_w, t_h, t_o and $t_{v_{1-104}}$. The final outputs (b_x, b_y - x, y co-ordinates of the box centroid, b_w, b_h - width and height of the box, b_o - objectness score, $b_{\text{id}}, b_{\text{exp}}, b_{\mathbf{R}}, b_{\mathbf{t}}, b_{\text{f}}$ - 3DMM parameters and b_{lm} - corresponding 2D landmarks) are then given by:

$$b_x = \sigma(t_x) + c_x;\ b_y = \sigma(t_y) + c_y;\ b_w = p_w * e^{t_w};\ b_h = p_h * e^{t_h}$$

$$b_o = Pr(\text{face}) * IOU(b, \text{face}) = \sigma(t_o)$$

$$b_{\text{id}} = t_{v_{1-50}};\ b_{\text{exp}} = \sigma(t_{v_{51-97}})$$

$$b_{\mathbf{R}} = t_{v_{98-101}};\ b_{\mathbf{t}} = t_{v_{102-104}};\ b_{\text{f}} = \sigma(t_{v_{105}})$$

$$b_{\text{lm}_x} = b_x + b_w * b_{\hat{\text{lm}}_x};\ b_{\text{lm}_y} = b_y + b_h * b_{\hat{\text{lm}}_y}$$

where σ denotes sigmoid function, (c_x, c_y) is the offset of the cell containing b from the top left corner of the image, (p_w, p_h) are the dimensions of the bounding box prior, $b_{\hat{\text{lm}}}$ are the initial landmarks obtained using equation 3.2 and IOU denotes intersection over union. As evident from the equations, the landmark loss puts additional constraints on the bounding box locations and

dimensions, thereby improving the accuracy of face detection in the joint training compared to simple face detection. Our final loss function is the summation of Eq. 3.3 across all grids and anchors, as shown in the following equation:

$$\tau * \Bigg\{ \frac{1}{50} \sum_{j=1}^{9^2} \sum_{k=1}^{5} \sum_{i=1}^{50} \mathbb{1}_{ijk} |w_{\text{id}_{ijk}} - w^{g}_{\text{id}_{ijk}}| + \frac{1}{46} \sum_{j=1}^{9^2} \sum_{k=1}^{5} \sum_{i=1}^{46} \mathbb{1}_{ijk} |w_{\text{exp}_{ijk}} - w^{g}_{\text{exp}_{ijk}}| + \frac{1}{4} \sum_{j=1}^{9^2} \sum_{k=1}^{5} \sum_{i=1}^{4} \mathbb{1}_{ijk} |\mathbf{R}_{ijk} - \mathbf{R}^{g}_{ijk}| \Bigg\} + \sqrt{\frac{1}{68} \sum_{j=1}^{9^2} \sum_{k=1}^{5} \sum_{i=1}^{68} \mathbb{1}_{ijk} (\mathbf{P}_{\text{lm}_{ijk}} - \mathbf{P}^{g}_{\text{lm}_{ijk}})^2} \quad (3.5)$$

where $\mathbb{1}_{ijk}$ denotes whether a kth bounding box predictor in cell j contains a face.

Our joint detection and retargeting network architecture, shown in Fig. 3.3, is similar to Tiny DarkNet [113] with the final layer changed to predict a tensor of size $9 \times 9 \times 5 \times 109$. However, since we only have one object class (face) in our problem, we reduce the number of filters in each layer to a quarter of their original values. We extend the multi-scale backbone for single face retargeting by changing the input image size to 288×288 and the output of each branch to accommodate grid output. The pose branch outputs change from 4 (R) + 3 (T) + 1 (f) = 8 to $9 \times 9 \times 5 \times 8$. The expression branch outputs change from 46 to $9 \times 9 \times 5 \times 46$, and identity branch outputs change from 50 to $9 \times 9 \times 5 \times 50$. One extra branch is also added to output objectness and bounding box location ($9 \times 9 \times 5 \times (4 + 1)$).

3.3 Experimental Setup

3.3.1 Datasets

For single face retargeting, we combine multiple datasets to have a good training set for accurate prediction of each 3DMM parameters. 300W-LP contains many large poses and Facewarehouse is a rich dataset for expressions. The ground truth 68 2D landmarks provided by these datasets are used to obtain 3DMM ground truth by [12]. LFW and AFLW2000-3D are used as test sets for static images and 300VW is used as test set for tracking on videos. For multiple face retargeting, AFW has ground truth bounding box, pose angles and 6 landmarks and is used as a test set for static images, while FDDB and WIDER only provide bounding box ground truth and are therefore

Table 3.1: Number of images or videos and faces for each dataset used in training and testing of our networks.

Dataset		#images	#faces
SFN	300W-LP [122, 179]	61225	61225
	FacewareHouse [17]	5000	500
	LFW [60]	12639	12639
	AFLW2000-3D [179]	2000	2000
	300VW [127]	114 (videos)	218K
MFN	FDDB [66]	2845	5171
	WIDER [158]	11905	56525
	AFW [111]	205	1000
	Music videos [171]	8 (videos)	-

used for training (WIDER test set is kept separate for testing). Music videos dataset is used to test our MFN performance on videos. We remove all images with more than 20 faces and also remove faces whose bounding box dimensions are $<2\%$ of the image dimensions from both the training and test sets. This is because determining the facial expressions for such small faces is ambiguous even for human eyes. More details about the datasets are summarized in Table 3.1. We use an 80-20 split of the training set for training and validation. To measure the performance of expression accuracy, we manually collect an expression test set by selecting those extreme expression images. The number of images in each of the expression categories are: eye close: 185, eye wide: 70, brow raise: 124, brow anger: 100, mouth open: 81, jaw left/right: 136, lip roll: 64, smile: 105, kiss: 143, total: 1008 images.

3.3.2 Implementation Details

We use Keras [25] with Tensorflow backend for our implementation. Both networks are trained using Adam optimizer with batch size 32. The initial learning rate is set to 10^{-3} and 10^{-4} for SFN

and MFN respectively and is decreased by 10 times when the validation loss does not change over 5 consecutive epochs, until the learning rate reaches 10^{-6}. Training takes about a day on a Nvidia GTX 1080 for each network. For data augmentation, we use random scaling in the range [0.8,1.2], random translation of 0-10%, color jitter and in-plane rotation. These augmentation techniques, apart from improving the performance of SFN, help in generating more accurate ground truth for individual faces for MFN.

3.3.3 Evaluation Metrics

We use 4 metrics: 1) average precision (AP) with different intersection-over-union thresholds as defined in [114] to evaluate our MFN performance for face detection, 2) normalized mean error (NME) defined as the Euclidean distance between the predicted and ground truth 2D landmarks averaged over 68 landmarks and normalized by the bounding box dimensions, 3) area under the curve (AUC) of the Cumulative Error Distribution curve for landmark error normalized by the diagonal distance of ground truth bounding box [31], and 4) expression metric defined as the mean absolute distance between the predicted expression parameters with respect to the ground truth, which is 1 in our case following the practice of [17].

3.3.4 Face Tracking for SFN

For a video, we perform retargeting on a frame-by-frame basis. We start by feeding the entire image into the model and keep shrinking the bounding box over time until it fits the face. Then we set the bounding box of the current frame to be same as the bounding box obtained from the boundaries of the 2D landmarks of the previous frame. Hence, unlike other methods, our method does not require a face detection module at every frame or regular intervals or for the first frame.

3.4 Results

3.4.1 Importance of Multi-Scale Representation

Our multi-scale network design reduces the load on the network to learn complex features by allowing the network to concentrate on different image features to learn different parameters unlike the single scale design. In Fig. 3.6, we see that single scale network learns generic facial features that combines the representations for identity, expression and pose. On the other hand, multi-scale network learns different levels of representation (pixel-level detailed features for expression, region level features for identity and global aggregate features for pose). We have randomly chosen only 25 filter outputs at level 3 of our SFN for clearer visualization. Table 3.2 shows that our multi-scale design not only reduces NME for single face images using SFN but also improves the performance of MFN in terms of both normalized mean error (by generating a better weakly supervised ground truth) and average precision for detection. Clearly, different feature representations are crucial to accurately learn different groups of parameters. By reducing the network load, this design also allows model compression so that multi-scale networks can be of comparable size with respect to single scale networks while having better accuracy. Fig. 3.7a shows that the multi-scale design predicts more accurate expression parameters (first row has correct landmarks for closed eyes) and identity parameters (second row has correct landmarks that fit the face shape) while being robust to large poses (second row), illumination (first row) and occlusion (third row).

3.4.2 Importance of Joint Training

Joint regression of both face bounding box locations and 3DMM parameters forces the network to learn exclusive facial features that characterize face shape, expression and pose in addition to differentiating face regions from the background. This helps in more precise face detection in-the-wild by leveraging both 2D information from bounding boxes and 3D information from 3DMM parameters. Table 3.2 shows that average precision (AP) is improved by a large margin with joint training compared to when the same network is trained to only regress bounding box locations. The retargeting accuracy for MFN is also comparable to that of SFN and the slight decrease in NME

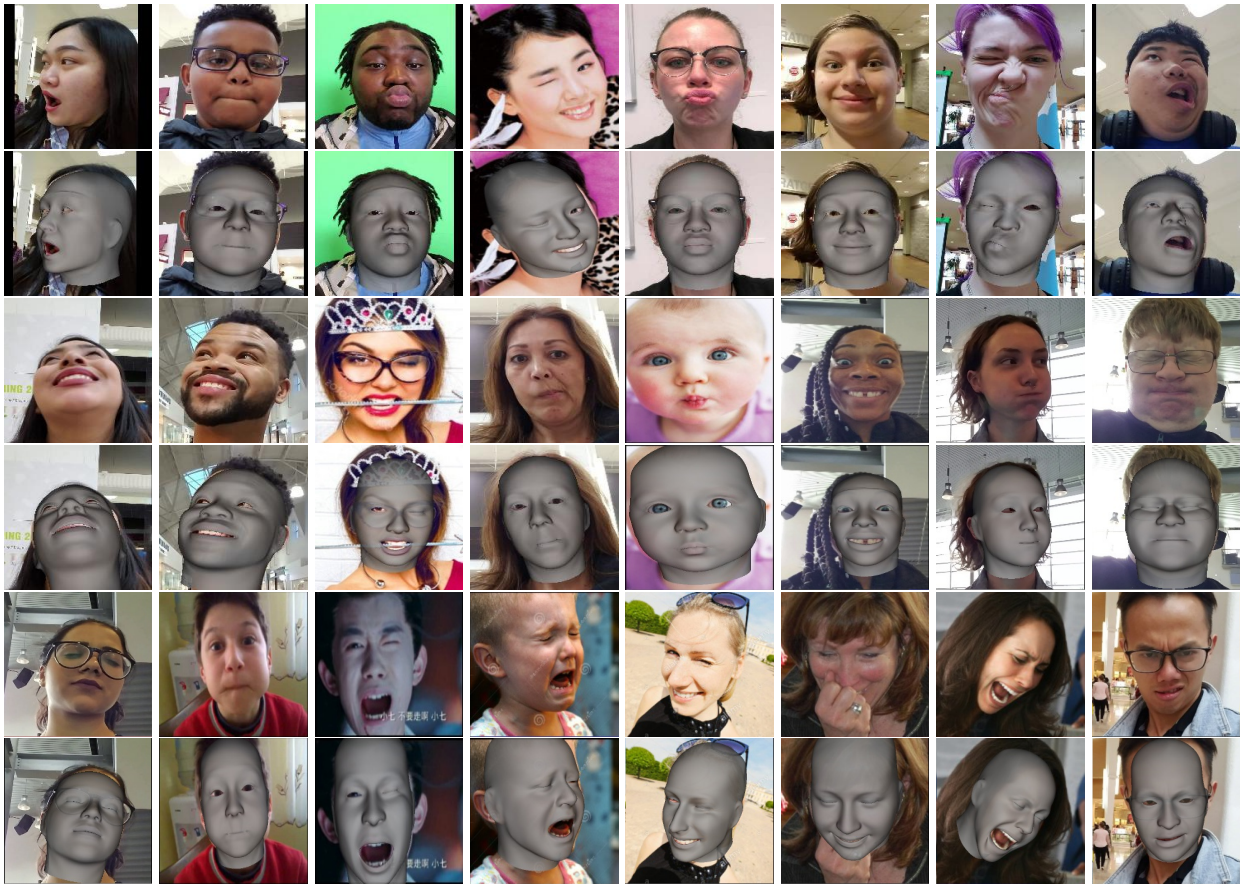


Figure 3.4: A few results from our own expression testing set using our single face retargeting network.

is because of training MFN on multi-face images and testing on single face images. Nevertheless, we observe improved performance in terms of both NME and AP by using better ground truth generated by multi-scale model. Results of our MFN on multi-face images are illustrated in Fig. 3.5.

3.4.3 Importance of Two Stage Refining

The optional second stage added to our single face retargeting network enables the network to learn exclusive eye and mouth region features independent of each other to correct the expression parameter values predicted by the second stage. Fig. 3.7b shows some examples where the second stage predict more accurate expression parameters compared to the first stage. We would like to

Figure 3.5: Testing results of our joint detection and retargeting model on AFW and WIDER. We show both the predicted bounding boxes in one row and the corresponding 3D meshes constructed from 3DMM parameters in the subsequent row. Our method can handle any number of faces of any shape.

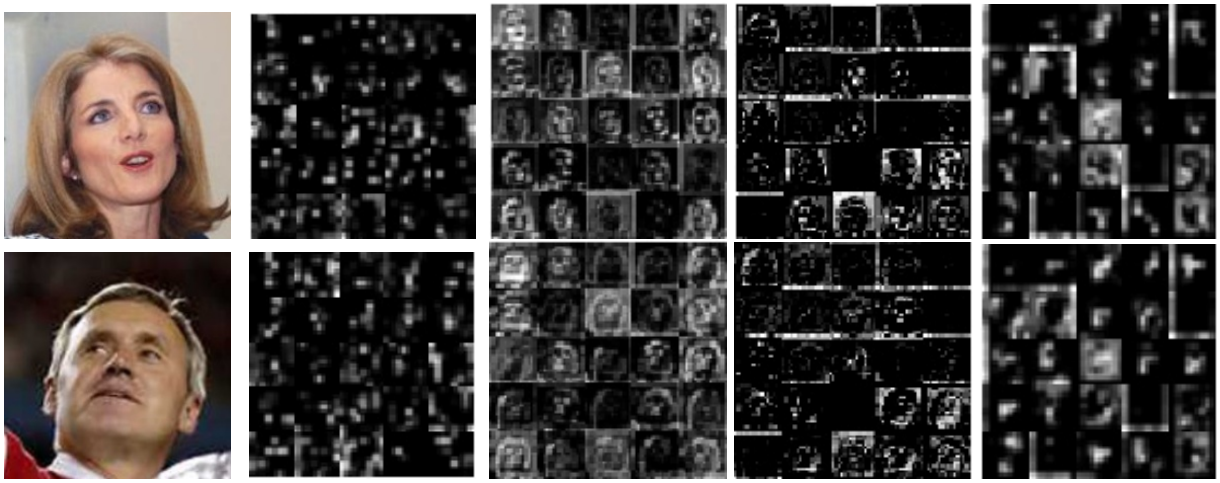


Figure 3.6: Visualization of learned features. From left to right in each row: input image, features for single scale SFN, features for expression branch of multi-scale SFN, features for identity branch of multi-scale SFN, features for pose branch of multi-scale SFN.

point out that in some cases, our prediction is more accurate compared to the provided ground truth as evident from the figure. We also observe a decrease of NME by ˜3% on AFLW2000-3D by the two-stage network. The improvement is mainly because it is impossible to have a balanced distribution of different combinations of eye and mouth expressions in the training data, and the two-stage training alleviates the need for such balanced distribution.

3.4.4 Performance of Face Detection

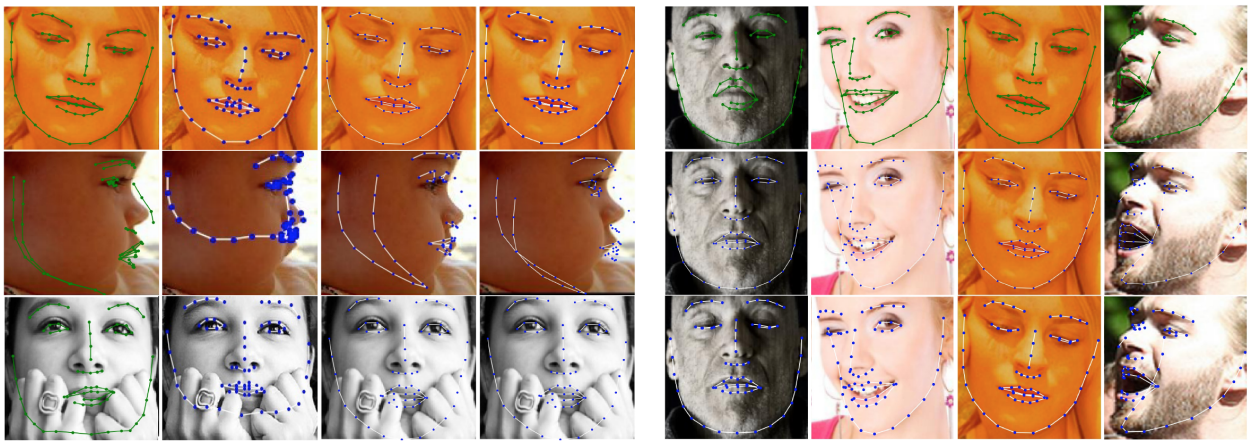
Our network can detect multiple small faces of reasonable size even though it is not trained on images more than 20 faces. Our detection accuracy (mAP: 98.8%) outperforms Hyperface [112] (mAP: 97.9%) and Faceness-Net [157] (mAP: 97.2%) on the entire AFW dataset when compared under the same settings. Figure 3.8 shows our network outputs for an image with more than 20 faces in the AFW dataset. The blue rectangles denote the predicted bounding boxes and the red points denote the 2D landmarks (for better viewing) projected from the predicted 3DMM parameters.

Table 3.2: Quantitative evaluation of our SFN and MFN. SS-MFN and MS-MFN denote single scale and multi-scale MFN respectively. NME values are calculated for LFW (single faces) and AP values are calculated for AFW.

Model	Evaluation			
	NME	Multi Face		
	(%)	AP	AP50	AP75
(1) MFN (detection only)	-	92.1	99.2	94.3
(2) Single scale SFN	2.16	-	-	-
(3) Multi-scale SFN	1.91	-	-	-
(4) SS-MFN + GT from (2)	2.89	97.5	99.8	98.1
(5) SS-MFN + GT from (3)	2.65	98.2	100	98.9
(6) MS-MFN + GT from (3)	2.23	98.8	100	99.3

3.4.5 Comparison with 2D Alignment Methods

Even though we aim to predict the 3DMM parameters for retargeting applications, our model can naturally serve the purpose for 2D face alignment (via 3D) as well. Therefore, we can evaluate the model from the performance of 2D alignment perspective. Table 3.3 compares the performance of our single scale and multi-scale SFN with state-of-the-art 2D face alignment methods. The images of AFLW2000 are divided into 3 groups based on yaw angles. As can be seen, our model achieves much smaller errors compared to most of the methods that are dedicated for precise landmark locations. While PRN [42] has lower NME, its network size is 160MB compared to only 2MB of ours and it doesn't perform in real time on mobile or even on a common PC CPU. In addition to evaluations on static images, we also measure the face tracking performance in a video using our SFN. Table 3.4 compares the AUC values on 300VW dataset for three scenarios categorized by the dataset. Our method performs significantly better than other methods (about 6% and 9% improvement over the second best method for Scenarios 1-2 and 3 respectively) with negligible failure rate because extensive data augmentation helps our tracking algorithm to quickly recover



(a) (b)

Figure 3.7: (a) 2D Face Alignment results for AFLW2000-3D. Column 1: original image with ground truth landmarks, Column 2: results using [12], Column 3: our single scale SFN, Column 4: our multi-scale SFN. (b) Evaluation of our two-stage SFN performance on AFLW200-3D. Row 1: Input images with ground truth landmarks, Row 2: Output of 1st stage of SFN, Row 3: Output of 2nd stage of SFN. The eye landmarks are corrected by the second stage in the first 3 columns and the mouth landmarks are corrected in the last column.

from failures.

3.4.6 Evaluation of Expressions

Our expression evaluation results in Table 4.4 emphasizes the improvement of multi-scale design on SFN. MS-MFN performs much better than SS-SFN for all expressions except the eye expressions. This is because eye patches are usually small compared to the entire image for MFN whereas they are zoomed in on cropped images for SFN. Attention network for emphasizing small eyes region could be a future work for our MFN. However, MS-MFN shows less accuracy compared to MS-SFN because it is being tested on single face images while being trained on multi-face images. For the multi-person test set images, we found similar visual results by applying MS-MFN on the whole image and by applying MS-SFN on each face individually cropped from the image. This is expected because MFN is trained with ground truth from SFN. The performance of MS-SFN on

Figure 3.8: Network output for an image with multiple small faces from the AFW dataset.

our expression test set is shown in Fig. 3.4. We also conducted live performance capture experiments to evaluate the efficiency our system in retargeting facial motion from one or more faces to one or more 3D characters. Fig. 3.9 shows some screenshots recorded during the experiments. The performance of our networks on face videos is shown on the project webpage[1]. In the first half of the video, we show the results of retargeting from a single face video to a generic 3D human face model using our single face retargeting network. The face bounding box for the current frame is obtained from the boundaries of the 2D landmarks predicted in the previous frame. In the second half of the video, we show the retargeting results with videos having multiple faces using our multi-face retargeting network (only frames with multiple faces are shown).

3.4.7 Computational Complexity

Excluding the IO time, SFN can run at 15ms/frame on Google Pixel 2 (assuming single face and excluding face detector runtime). Face detection with our compressed detector model is 34ms, so

[1] https://homes.cs.washington.edu/~bindita/multifaceretargeting.html

Figure 3.9: Retargeting from face(s) to 3D character(s).

separate face detection and retargeting requires 49ms for 1 face, 109ms for 5 faces and 184ms for 10 faces. On the other hand, our proposed MFN performs joint face detection and retargeting at 39ms on any number of faces. The model sizes for compressed face detector is 11.5 MB and SFN is 2 MB, so the combination is 13.5 MB, while our MFN is only 7.8 MB. Hence our joint network reduces both memory requirement and execution time.

Table 3.3: Comparison of NME(%) for AFLW2000-3D (68 landmarks). 3DDFA2 refers to 3DDFA+SDM [179].

Method	[0°,30°]	[30°,60°]	[60°,90°]	Mean
SDM [154]	3.67	4.94	9.67	6.12
3DDFA [179]	3.78	4.54	7.93	5.42
3DDFA2 [179]	3.43	4.24	7.17	4.94
Yu et al. [163]	3.62	6.06	9.56	6.41
3DSTN [8]	3.15	4.33	5.98	4.49
DFF [68]	3.20	4.68	6.28	4.72
PRN [42]	2.75	3.51	4.61	3.62
SS-SFN (ours)	3.09	4.27	5.59	4.31
MS-SFN (ours)	2.91	3.83	4.94	3.89

Table 3.4: Landmark localization performance of our method on videos in comparison to state-of-the-art face tracking methods. The values are reported in terms of Area under the Curve (AUC) for Cumulative Error Distribution of the 2D landmark error for 300VW test set.

Method	Scenario 1	Scenario 2	Scenario 3
Yang et al. [156]	0.791	0.788	0.710
Xiao et al. [153]	0.760	0.782	0.695
CFSS [176]	0.784	0.783	0.713
MTCNN [169]	0.748	0.760	0.726
MHM [31]	0.847	0.838	0.769
SFN (ours)	**0.901**	**0.884**	**0.842**

Chapter 4

PERSONALIZED FACE MODELING FOR IMPROVED FACE TRACKING AND RETARGETING

4.1 Introduction

In order to perform tracking for retargeting, blendshape interpolation technique is usually adopted where the users' blendshapes are obtained by deformation transfer [131], but this alone cannot reconstruct expressions realistically as shown in [48, 82]. Another popular technique is to use a multilinear tensor-based 3DMM [146, 15, 14], where the expression is coupled with the identity implying that same identities should share the same expression blendshapes. However, we argue that facial expressions are characterized by different skin deformations on different persons due to difference in face shape, muscle movements, age and other factors. This kind of user-specific local skin deformations cannot be accurately represented by a linear combination of predefined blendshapes. For example, smiling and raising eyebrows create different cheek folds and forehead wrinkle patterns respectively on different persons, which cannot be represented by simple blendshape interpolation and require correcting the corresponding blendshapes. Some optimization-based approaches [82, 48, 63, 116] have shown that modeling user-specific blendshapes indeed results in a significant improvement in the quality of face reconstruction and tracking. However, these approaches are computationally slow and require additional preprocessing (e.g. landmark detection) during test time, which significantly limits real-time applications with in-the-wild data on the edge devices. The work [22] trains a deep neural network instead to perform retargeting in real-time on typical mobile phones, but its use of predefined 3DMM limits its face modeling accuracy. Tewari et al. [138] leverage in-the-wild videos to learn face identity and appearance models from scratch, but they still use expression blendshapes generated by deformation transfer.

Moreover, existing methods learn a single albedo map for a user. The authors in [52] have

shown that skin reflectance changes with skin deformations, but it is not feasible to generate a separate albedo map for every expression during retargeting. Hence it is necessary to learn the static reflectance separately, and associate the expression-specific dynamic reflectance with the blendshapes so that the final reflectance can be obtained by interpolation similar to blendshape interpolation, as in [102]. Learning dynamic albedo maps in addition to static albedo map also helps to capture the fine-grained facial expression details like folds and wrinkles [104], thereby resulting in reconstruction of higher fidelity.

To address these issues, we introduce a novel end-to-end framework that leverages a large corpus of in-the-wild user videos to jointly learn personalized face modeling and face tracking parameters. Specifically, we design a modeling network which learns geometry and reflectance corrections on top of a 3DMM prior to generate user-specific expression blendshapes and dynamic (expression-specific) albedo maps. In order to ensure proper disentangling of the geometry from the albedo, we introduce the face parsing loss inspired by [177]. Note that [177] uses parsing loss in a fitting based framework whereas we use it in a learning based framework. We also ensure that the corrected blendshapes retain their semantic meanings by restricting the corrections to local regions using attention maps and by enforcing a blendshape gradient loss. We design a separate tracking network which predicts the expression blendshape coefficients, head pose and scene lighting parameters. The decoupling between the modeling and tracking networks enables our framework to perform reconstruction as well as retargeting (by tracking one user and transferring the facial motion to another user's model). Our main contributions are:

1. We propose a deep learning framework to learn user-specific expression blendshapes and dynamic albedo maps that accurately capture the complex user-specific expression dynamics and high-frequency details like folds and wrinkles, thereby resulting in photorealistic 3D face reconstruction.

2. We bring two novel constraints into the end-to-end training: face parsing loss to reduce the ambiguity between geometry and reflectance and blendshape gradient loss to retain the semantic meanings of the corrected blendshapes.

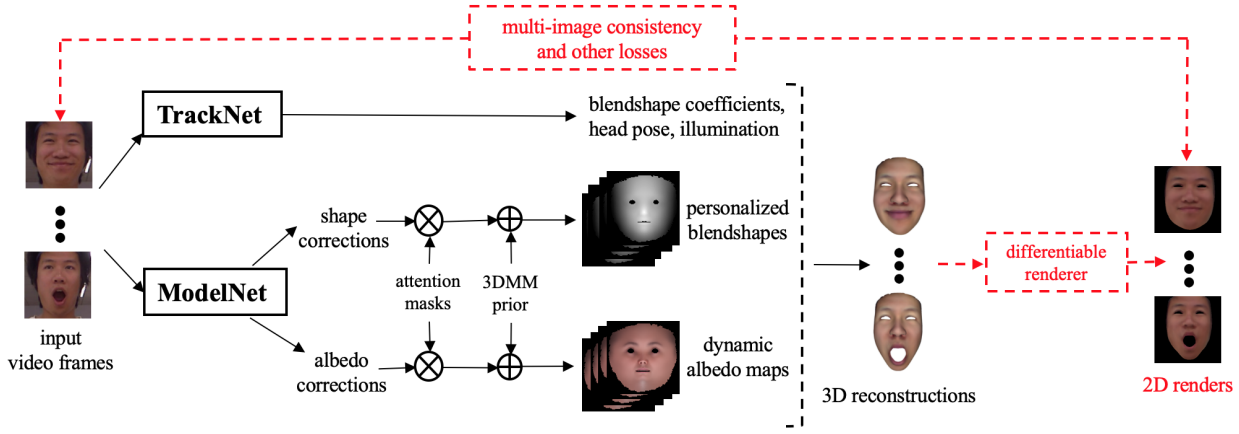


Figure 4.1: Our end-to-end framework. Our framework takes frames from in-the-wild video(s) of a user as input and generates per-frame tracking parameters via the *TrackNet* and personalized face model via the *ModelNet*. The networks are trained together in an end-to-end manner (marked in red) by projecting the reconstructed 3D outputs into 2D using a differentiable renderer and computing multi-image consistency losses and other regularization losses.

3. Our framework jointly learns a user-specific face model and user-independent facial motion in disentangled form, thereby supporting motion retargeting.

4.2 Methodology

Our network architecture, as shown in Fig. 4.1, has two parts: 1) *ModelNet* which learns to capture the user-specific facial details and 2) *TrackNet* which learns to capture the user-independent facial motion. The networks are trained together in an end-to-end manner using multi-frame images of different identities, i.e., multiple images $\{I_1, \ldots, I_N\}$ of the same person sampled from a video in each mini-batch. We leverage the fact that the person's facial geometry and appearance remain unchanged across all the frames in a video, whereas the facial expression, head pose and scene illumination change on a per-frame basis. The *ModelNet* extracts a common feature from all the N images to learn a user-specific face shape, expression blendshapes and dynamic albedo maps (Section 4.2.1). The *TrackNet* processes each of the N images individually to learn the image-

specific expression blendshape coefficients, pose and illumination parameters (Section 4.2.2). The predictions of *ModelNet* and *TrackNet* are combined to reconstruct the 3D faces and then projected to the 2D space using a differentiable renderer in order to train the network in a self-supervised manner using multi-image photometric consistency, landmark alignment and other constraints. During testing, the default settings can perform 3D face reconstruction. However, our network architecture and training strategy allow simultaneous tracking of one person's face using *TrackNet* and modeling another person's face using *ModelNet*, and then retarget the tracked person's facial motion to the modeled person or an external face model having similar topology as our face model.

4.2.1 Learning Personalized Face Model

Our template 3D face consists of a mean (neutral) face mesh S_0 having 12K vertices, per-vertex colors (converted to 2D mean albelo map R_0 using UV coordinates) and 56 expression blendshapes $\{S_1, \ldots, S_{56}\}$. Given a set of expression coefficients $\{w_1, \ldots, w_{56}\}$, the template face shape can be written as $\bar{S} = w_0 S_0 + \sum_{i=1}^{56} w_i S_i$ where $w_0 = (1 - \sum_{i=1}^{56} w_i)$. Firstly, we propose to learn an identity-specific corrective deformation Δ_0^S from the identity of the input images to convert $\bar{S}$ to identity-specific shape. Then, in order to better fit the facial expression of the input images, we learn corrective deformations Δ_i^S for each of the template blendshapes S_i to get identity-specific blendshapes. Similarly, we learn a corrective albedo map Δ_0^R to convert R_0 to identity-specific static albedo map. In addition, we also learn corrective albedo maps Δ_i^R corresponding to each S_i to get identity-specific dynamic (expression-specific) albedo maps.

In our *ModelNet*, we use a shared convolutional encoder E^{model} to extract features F_n^{model} from each image $I_n \in \{I_1, \ldots, I_N\}$ in a mini-batch. Since all the N images belong to the same person, we take an average over all the F_n^{model} features to get a common feature F^{model} for that person. Then, we pass F^{model} through two separate convolutional decoders, D_S^{model} to estimate the shape corrections Δ_0^S and Δ_i^S, and D_R^{model} to estimate the albedo corrections Δ_0^R and Δ_i^R. We learn the corrections in the UV space instead of the vertex space to reduce the number of network parameters and preserve the contextual information.

User-specific expression blendshapes A naive approach to learn corrections on top of template blendshapes based on the user's identity would be to predict corrective values for all the vertices and add them to the template blendshapes. However, since blendshape deformation is local, we want to restrict the corrected deformation to a similar local region as the template deformation. To do this, we first apply an attention mask over the per-vertex corrections and then add it to the template blendshape. We compute the attention mask A_i corresponding to the blendshape S_i by calculating the per-vertex Euclidean distances between S_i and S_0, thresholding them at 0.001, normalizing them by the maximum distance, and then converting them into the UV space. We also smooth the mask discontinuities using a small amount of Gaussian blur following [102]. Finally, we multiply A_i with Δ_i^S and add it to S_i to obtain a corrected S_i. Note that the masks are precomputed and then fixed during network operations. The final face shape is thus given by:

$$S = w_0 S_0 + \mathcal{F}(\Delta_0^S) + \sum_{i=1}^{56} w_i [S_i + \mathcal{F}(A_i \Delta_i^S)] \tag{4.1}$$

where $\mathcal{F}(\cdot)$ is a sampling function for UV space to vertex space conversion.

User-specific dynamic albedo maps We use one static albedo map to represent the identity-specific neutral face appearance and 56 dynamic albedo maps, one for each expression blendshape, to represent the expression-specific face appearance. Similar to blendshape corrections, we predict 56 albedo correction maps in the UV space and add them to the static albedo map after multiplying the dynamic correction maps with the corresponding UV attention masks. Our final face albedo is thus given by:

$$R = R_0^t + \Delta_0^R + \sum_{i=1}^{56} w_i [A_i \Delta_i^R] \tag{4.2}$$

where R_0^t is the trainable mean albedo initialized with the mean albedo R_0 from our template face similar to [138].

Table 4.1: Architecture of our networks. $s\#$ refers to stride $\#$. (a) Architecture of the shared encoder of our modeling network. The outputs of the encoder for each input image in a mini-batch are average pooled to obtain a single $(7, 7, 512)$ feature that becomes the input to both the decoders. (b) Architecture of each of the decoder of our modeling network. Note that the output of the last Deconv2D layer goes into both the last 2 Conv2D layers.

(a)

Layers	Input Shape	Output Shape
Conv2D (7×7, s2)	(224,224,3)	(112,112,64)
Maxpool (3×3, s2)	(112,112,64)	(56,56,64)
Conv2D (3×3, s1)	(56,56,64)	(56,56,128)
Conv2D (3×3, s2)	(56,56,128)	(28,28,128)
Conv2D (3×3, s1)	(28,28,128)	(28,28,256)
Conv2D (3×3, s2)	(28,28,256)	(14,14,256)
Conv2D (3×3, s1)	(14,14,256)	(14,14,512)
Conv2D (3×3, s2)	(14,14,512)	(7,7,512)

(b)

Layers	Input Shape	Output Shape
Deconv2D (4×4, s2)	(7,7,512)	(16,16,512)
Deconv2D (4×4, s2)	(16,16,512)	(32,32,256)
Deconv2D (4×4, s2)	(32,32,256)	(64,64,128)
Deconv2D (4×4, s2)	(64,64,128)	(128,128,64)
Conv2D (1×1, s1)	(128,128,64)	(128,128,3)
Conv2D (1×1, s1)	(128,128,64)	(128,128,56*3)

4.2.2 Joint Modeling and Tracking

The *TrackNet* consists of a convolutional encoder E^{track} followed by multiple fully connected layers to regress the tracking parameters $\mathbf{p}_n = (\mathbf{w}_n, \mathbf{R}_n, \mathbf{t}_n, \gamma_n)$ for each image I_n. The encoder and fully connected layers are shared over all the N images in a mini-batch. Here $\mathbf{w}_n = (w_0^n, \ldots, w_{56}^n)$ is the expression coefficient vector and $\mathbf{R}_n \in SO(3)$ and $\mathbf{t}_n \in \mathbb{R}^3$ are the head rotation (in terms of Euler angles) and 3D translation respectively. $\gamma_n \in \mathbb{R}^{27}$ are the 27 Spherical Harmonics coefficients (9 per color channel) following the illumination model of [138].

Training Phase: We first obtain a face shape S_n and albedo R_n for each I_n by combining S (equation 4.1) and R (equation 4.2) from the *ModelNet* and the expression coefficient vector $\mathbf{w}_n$ from the *TrackNet*. Then, similar to [49, 138], we transform the shape using head pose as $\tilde{S}_n = \mathbf{R}_n S_n + \mathbf{t}_n$ and project it onto the 2D camera space using a perspective camera model $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$. Finally,

we use a differentiable renderer $\mathcal{R}$ to obtain the reconstructed 2D image as $\hat{I}_n = \mathcal{R}(\tilde{S}_n, \mathbf{n}_n, R_n, \gamma_n)$ where $\mathbf{n}_n$ are the per-vertex normals. We also mark 68 facial landmarks on our template mesh which we can project onto the 2D space using Φ to compare with the ground truth 2D landmarks.

Testing Phase: The *ModelNet* can take a variable number of input images of a person (due to our feature averaging technique) to predict a personalized face model. The *TrackNet* executes independently on one or more images of the same person given as input to *ModelNet* or a different person. For face reconstruction, we feed images of the same person to both the networks and combine their outputs as in the training phase to get the 3D faces. In order to perform facial motion retargeting, we first obtain the personalized face model of the target subject using *ModelNet*. We then predict the facial motion of the source subject on a per-frame basis using the *TrackNet* and combine it with the target face model. It is important to note that the target face model can be any external face model with semantically similar expression blendshapes.

4.2.3 Loss Functions

We train both the *TrackNet* and the *ModelNet* together in an end-to-end manner using the following loss function:

$$L = \lambda_{\text{ph}} L_{\text{ph}} + \lambda_{\text{lm}} L_{\text{lm}} + \lambda_{\text{pa}} L_{\text{pa}} + \lambda_{\text{sd}} L_{\text{sd}} + \lambda_{\text{bg}} L_{\text{bg}} + \lambda_{\text{reg}} L_{\text{reg}} \tag{4.3}$$

where the loss weights λ_* are chosen empirically and their values are chosen empirically as $\lambda_{\text{ph}} = 200; \lambda_{\text{lm}} = 0.1; \lambda_{\text{pa}} = 50; \lambda_{\text{sd}} = 2.5; \lambda_{\text{bg}} = 1.5; \lambda_{\text{reg}} = 10^{-3}; \lambda_{\gamma} = 0.02$.

Photometric and Landmark Losses: We use the $l_{2,1}$ [144] loss to compute the multi-image photometric consistency loss between the input images I_n and the reconstructed images $\hat{I}_n$. The loss is given by

$$L_{\text{ph}} = \sum_{n=1}^{N} \frac{\sum_{q=1}^{Q} ||M_n(q) * [I_n(q) - \hat{I}_n(q)]||_2}{\sum_{q=1}^{Q} M_n(q)} \tag{4.4}$$

where M_n is the mask generated by the differentiable renderer (to exclude the background, eyeballs and mouth interior) and q ranges over all the pixels Q in the image. In order to further improve the quality of the predicted albedo by preserving high-frequency details, we add the image (spatial)

gradient loss having the same expression as the photometric loss with the images replaced by their gradients. Adding other losses as in [49] resulted in no significant improvement. The landmark alignment loss L_{lm} is computed as the l_2 loss between the ground truth and predicted 68 2D facial landmarks.

Face Parsing Loss: The photometric and landmark loss constraints are not strong enough to overcome the ambiguity between shape and albedo in the 2D projection of a 3D face. Besides, the landmarks are sparse and often unreliable especially for extreme poses and expressions which are difficult to model because of depth ambiguity. So, we introduce the face parsing loss given by:

$$L_{\text{pa}} = \sum_{n=1}^{N} ||I_n^{\text{pa}} - \hat{I}_n^{\text{pa}}||_2 \tag{4.5}$$

where I_n^{pa} is the ground truth parsing map generated using the method in [85] and $\hat{I}_n^{\text{pa}}$ is the predicted parsing map generated as $\mathcal{R}(\tilde{S}_n, \mathbf{n}_n, T)$ with a fixed precomputed UV parsing map T.

Shape Deformation Smoothness Loss: We employ Laplacian smoothness on the identity-specific corrective deformation to ensure that our predicted shape is locally smooth. The loss is given as:

$$L_{\text{sd}} = \sum_{v=1}^{V} \sum_{u \in \mathcal{N}_v} ||\Delta_0^S(v) - \Delta_0^S(u)||_2^2 \tag{4.6}$$

where V is the total number of vertices in our mesh and $\mathcal{N}_v$ is the set of neighboring vertices directly connected to vertex v.

Blendshape Gradient Loss: Adding free-form deformation to a blendshape, even after restricting it to a local region using attention masks, can change the semantic meaning of the blendshape. However, in order to retarget tracked facial motion of one person to the blendshapes of another person, the blendshapes of both the persons should have semantic correspondence. We introduce a novel blendshape gradient loss to ensure that the deformation gradients of the corrected blendshapes are similar to those of the template blendshapes. The loss is given by:

$$L_{\text{bg}} = \sum_{i=1}^{56} ||\mathbf{G}_{S_0 \to (S_i + \Delta_i^S)} - \mathbf{G}_{S_0 \to S_i}||_2^2 \tag{4.7}$$

where $\mathbf{G}_{a \to b}$ denotes the gradient of the deformed mesh b with respect to original mesh a. Details about how to compute $\mathbf{G}$ can be found in [82].

Tracking Parameter Regularization Loss: We use sigmoid activation at the output of the expression coefficients and regularize the coefficients using l_1 loss (L^w_{reg}) to ensure sparse coefficients in the range $[0, 1]$. In order to disentangle the albedo from the lighting, we use a lighting regularization loss given by:

$$L^{\gamma}_{\text{reg}} = ||\gamma - \gamma_{\text{mean}}||_2 + \lambda_{\gamma}||\gamma||_2 \tag{4.8}$$

where the first term ensures that the predicted light is mostly monochromatic and the second term restricts the overall lighting. We found that regularizing the illumination automatically resulted in albedo consistency, so we don't use any additional albedo loss. Finally, $L_{\text{reg}} = L^w_{\text{reg}} + L^{\gamma}_{\text{reg}}$.

4.3 Experimental Setup

4.3.1 Datasets

We train our network using two datasets: 1) VoxCeleb2 [27] and 2) ExpressiveFaces. We set aside 10% of each dataset for testing. VoxCeleb2 has more than 140k videos of about 6000 identities collected from internet, but the videos are mostly similar. So, we choose a subset of 90k videos from about 4000 identities. The images in VoxCeleb2 vary widely in pose but lack variety in expressions and illumination, so we add a custom dataset (ExpressiveFaces) to our training, which contains 3600 videos of 3600 identities. The videos are captured by the users using a hand-held camera (typically the front camera of a mobile phone) as they perform a wide variety of expressions and poses in both indoor and outdoor environments. We sample the videos at 10fps to avoid multiple duplicate frames, randomly delete frames with neutral expression and pose based on a threshold on the expression and pose parameters predicted by [22], and then crop the face and extract ground truth 2D landmarks using [22]. The cropped faces are resized to 224×224 and grouped into mini-batches, each of N images chosen randomly from different parts of a video to ensure sufficient diversity in the training data. We set $N = 4$ during training and $N = 1$ during testing (unless otherwise mentioned) as evaluated to work best for real-time performance in [138].

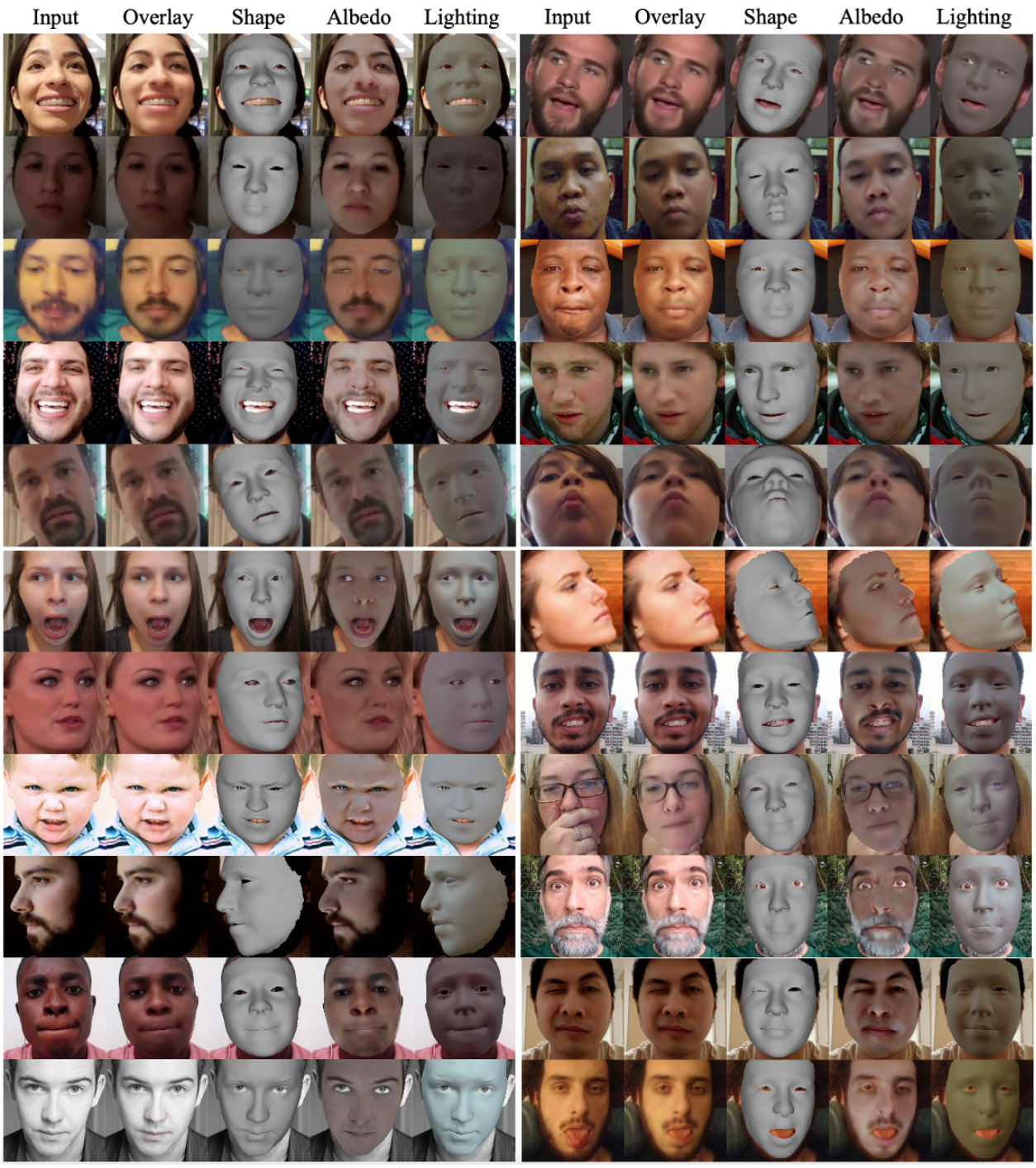


Figure 4.2: Qualitative results of our method. Our modeling network accurately captures high-fidelity facial details specific to the user, thereby enabling the tracking network to learn user-independent facial motion. Our network can handle a wide variety of pose, expression, lighting conditions, facial hair and makeup etc.

4.3.2 Implementation Details

We implemented our networks in Tensorflow and used TF mesh renderer [50] for differentiable rendering. During the first stage of training, we train both *TrackNet* and *ModelNet* in an end-to-end manner using equation 4.3. During the second stage of training, we fix the weights of *TrackNet* and fine-tune the *ModelNet* to better learn the expression-specific corrections. The fine-tuning is done using the same loss function as before except the tracking parameter regularization loss since the tracking parameters are now fixed. This training strategy enables us to tackle the bilinear optimization problem of optimizing the blendshapes and the corresponding coefficients, which is generally solved through alternate minimization by existing optimization-based methods. Besides, our method of obtaining the user-specific face shape and albedo from multiple frames helps in learning the static shape and albedo corrections separately from the expression-specific shape and albedo variations. As a result, our framework can produce photorealistic expression-specific deformations on a new user during testing. For training, we use a batch size of 8, learning rates of 10^{-4} (10^{-5} during second stage) and Adam optimizer for loss minimization. Training takes $\sim$20 hours on a single Nvidia Titan X for the first stage, and another $\sim$5 hours for the second stage. Since our template mesh contains 12264 vertices, we use a corresponding UV map of dimensions 128×128. The architectures of the encoder and decoders of our modeling network are given in Table 4.1a and Table 4.1b. Each Conv2D and Deconv2D layer is followed by batch normalization which is then followed by ReLU activation. Our end-to-end network has a size of 240 MB and takes 15.4 ms to execute 1 image and 37.5 ms to execute 4 images on a Titan X GPU on average.

4.4 Results

We evaluate the effectiveness of our framework using both qualitative results and quantitative comparisons. Fig. 4.2 shows the personalized face shape and albedo, scene illumination and the final reconstructed 3D face generated from monocular images by our method. Learning a common face shape and albedo from multiple images of a person separately from the image-specific facial motion helps in successfully decoupling the tracking parameters from the learned face model. As

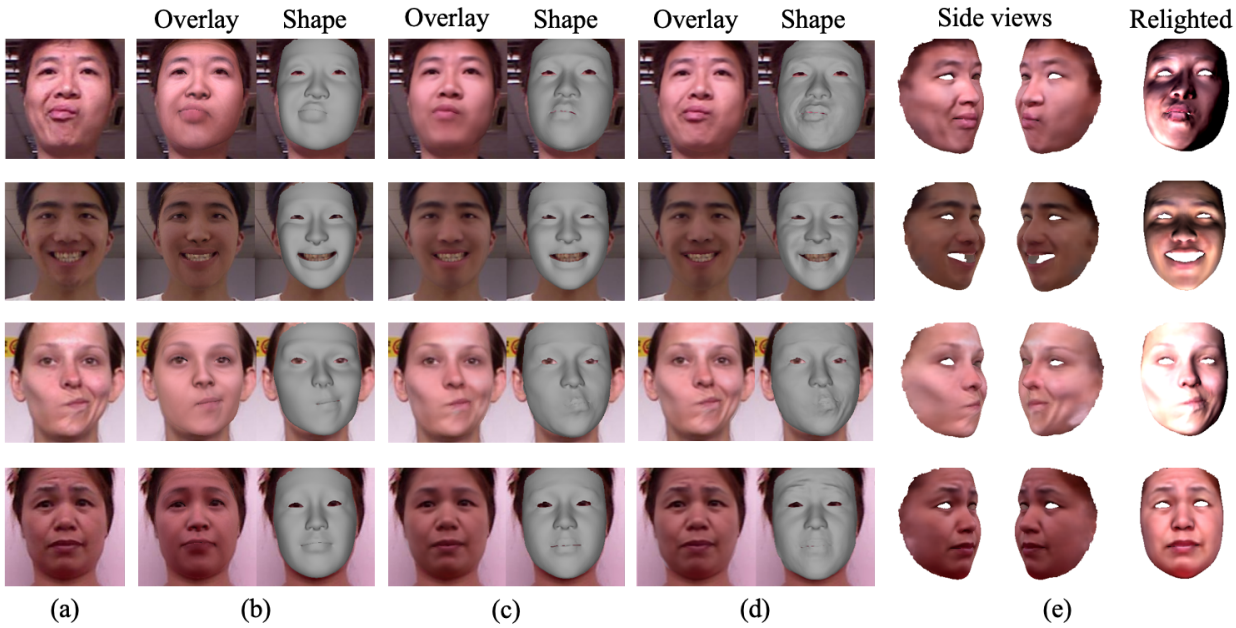


Figure 4.3: Importance of personalization. (a) input image, (b) reconstruction using 3DMM prior only, (c) reconstruction after adding only identity-based corrections, i.e. Δ_0^S and Δ_0^R in eq. (2) and (3) respectively, (d) reconstruction after adding expression-specific corrections, (e) results of (d) with different viewpoints and illumination.

a result, our tracking network have the capacity to represent a wide range of expressions, head pose and lighting conditions. Moreover, learning a unified model from multiple images help to overcome issues like partial occlusion, self-occlusion, blur in one or more images. Fig. 4.3 shows a gallery of examples that demonstrate the effectiveness of personalized face modeling for better reconstruction. Fig. 4.7a shows that our network can be efficiently used to perform facial motion retargeting to another user or to an external 3D model of a stylized character in addition to face reconstruction.

Fig. 4.2 shows 3D face reconstruction results using our method on our test data. It can be noted that our method can reconstruct faces accurately even under conditions like unusual lighting, uncommon face shapes (e.g. baby face), extreme poses, occlusion and extreme expressions. However, similar to [138], our method embeds eye glasses into the albedo. We would like to

point out that videos of ExpressiveFaces are captured by hand-held cameras and have resolution of 1920×1080, from which we crop faces resized to size 224×224. On the other hand, videos in Voxceleb2 [27] are scraped from Youtube and hence have a very different distribution (lower resolution) than the videos of ExpressiveFaces. The performance of our method on face videos is shown on the project webpage[1]. We show three applications of our method: a) personalized reconstruction, b) retargeting to a different user's face model, and c) retargeting to an external 3D puppet. To process the input video, we detect the face bounding box using [22] for the first frame only. For the subsequent frames, the bounding box of each frame is obtained from the boundaries of the 2D landmarks predicted in the previous frame. This technique helps in reducing the jitter in the results due to inconsistent bounding box selection if done on a per-frame basis. However, some temporal smoothing as a post-processing step will produce better results.

4.4.1 Importance of Personalized Face Model

Importance of personalized blendshapes: Modeling the user-specific local geometry deformations while performing expressions enable our modeling network to accurately fit the facial shape of the input image. Fig. 4.4a shows examples of how the same expression can look different on different identities and how the corrected blendshapes capture those differences for more accurate reconstruction than with the template blendshapes. In the first example, the extent of opening of the mouth in the *mouth open* blendshape is adjusted according to the user expression. In the second example, the mouth shape of the *mouth funnel* blendshape is corrected.

Importance of dynamic textures: Modeling the user-specific local variations in skin reflectance while performing expressions enable our modeling network to generate a photorealistic texture for the input image. Fig. 4.4b shows examples of how personalized dynamic albedo maps help in capturing the high-frequency expression-specific details compared to static albedo maps. In the first example, our method accurately captures the folds around the nose and mouth during smile

[1] https://homes.cs.washington.edu/~bindita/personalizedfacemodeling.html

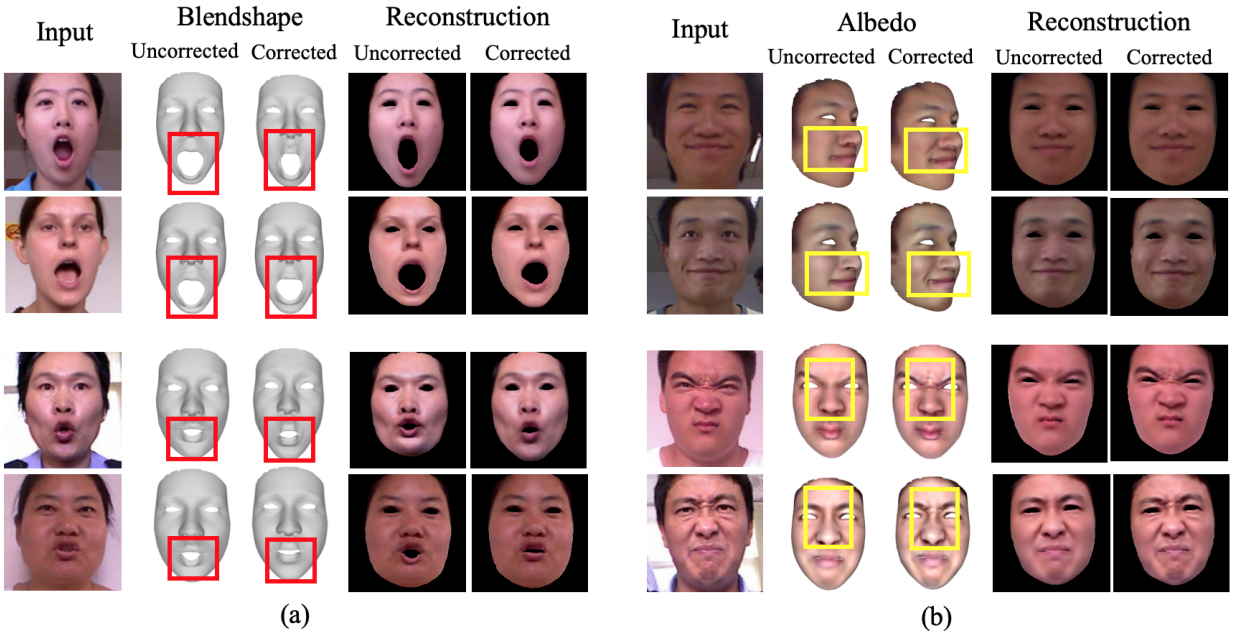


Figure 4.4: Visualization of corrected blendshapes and albedo. The corrections are highlighted. (a) Learning user-specific blendshapes corrects the mouth shape of the blendshapes, (b) Learning user-specific dynamic albedo maps captures the high-frequency details like skin folds and wrinkles.

expression. In the second example, the unique wrinkle patterns between the eyebrows of the two users are correctly modeled during a disgust expression.

4.4.2 Importance of Novel Training Constraints

Importance of parsing loss: The face parse map ensures that each face part of the reconstructed geometry is accurate as shown in [177]. This prevents the albedo to compensate for incorrect geometry, thereby disentangling the albedo from the geometry. However, the authors of [177] use parse map in a geometry fitting framework which, unlike our learning framework, does not generalize well to in-the-wild images. Besides, due to the dense correspondence of parse map compared to the sparse 2D landmarks, parsing loss (a) provides a stronger supervision on the geometry, and (b) is more robust to outliers. We demonstrate the effectiveness of face parsing loss in Fig. 4.5a. In the first example, the kiss expression is correctly reconstructed with the loss, since the 2D land-

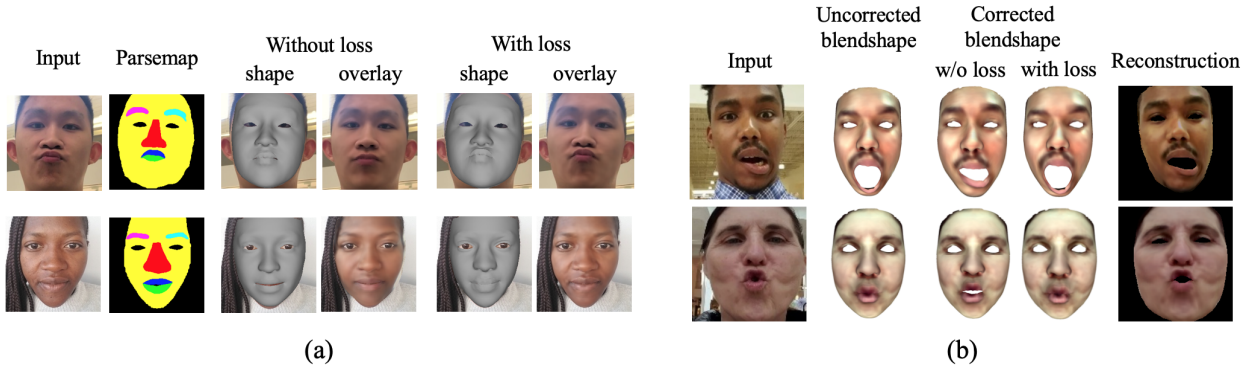


Figure 4.5: Importance of novel training constraints. (a) importance of face parsing loss in obtaining accurate geometry decoupled from albedo, (b) importance of blendshape gradient loss in retaining the semantic meaning of *mouth open* (row 1) and *kiss* (row 2) blendshapes after correction.

marks are not enough to overcome the depth ambiguity. In the second example, without parsing loss the albedo tries to compensate for the incorrect geometry by including the background in the texture. With loss, the nose shape, face contour and the lips are corrected in the geometry, resulting in better reconstruction.

Importance of blendshape gradient loss: Even after applying attention masks to restrict the blendshape corrections to local regions, our method can distort a blendshape such that it loses its semantic meaning, which is undesirable for retargeting purposes. We prevent this by enforcing the blendshape gradient loss, as shown in Fig. 4.5b. In the first example, without gradient loss, the *mouth open* blendshape gets combined with *jaw left* blendshape after correction in order to minimize the reconstruction loss. With gradient loss, the reconstruction is same but the *mouth open* blendshape retains its semantics after correction. Similarly in the second example, without gradient loss, the *kiss* blendshape gets combined with the *mouth funnel* blendshape, which is prevented by the loss.

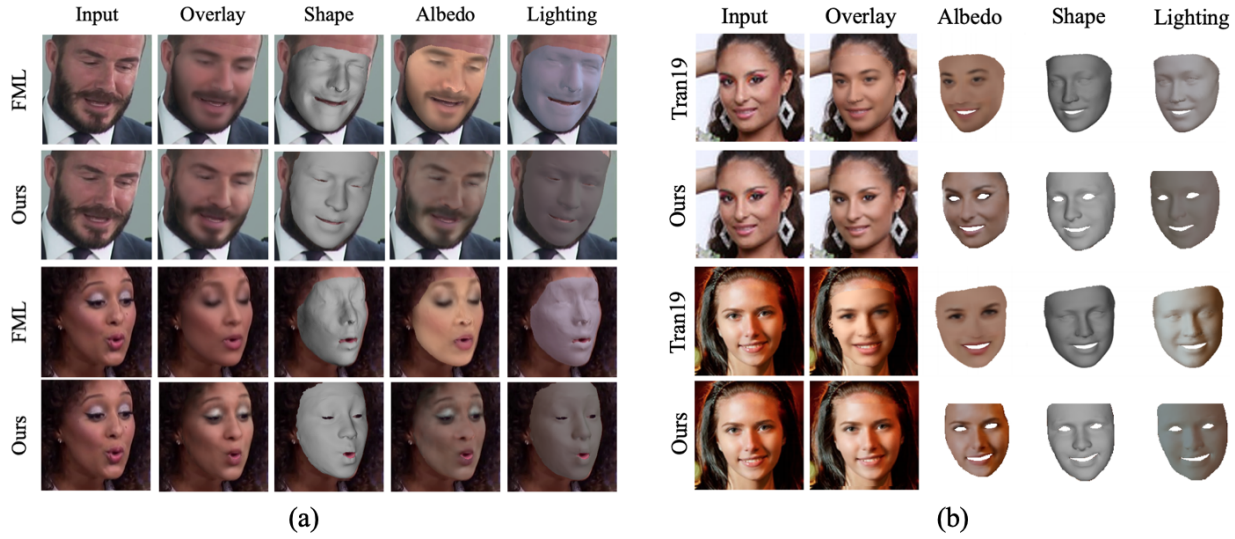


Figure 4.6: Visual comparison with (a) FML [138], (b) Non-linear 3DMM [144].

4.4.3 Visual Comparison with State-of-the-art Methods

3D face reconstruction: We test the effectiveness of our method on VoxCeleb2 test set to compare with FML results [138] as shown in Fig. 4.6a. In the first example, our method captures the mouth shape and face texture better. The second example shows that our personalized face modeling can efficiently model complex expressions like kissing and complex texture like eye shadow better than FML. We also show the visual comparisons between our method and Non-linear 3DMM [144] on the AFLW2000-3D dataset [179] in Fig. 4.6b. Similar to FML, Non-linear 3DMM fails to accurately capture the subtle facial details.

Face tracking and retargeting: By increasing the face modeling capacity and decoupling the model from the facial motion, our method performs superior face tracking and retargeting compared to a recent deep learning based retargeting approach [22]. Fig. 4.7a shows some frames from a user video and how personalization helps in capturing the intensity of the expressions more accurately.

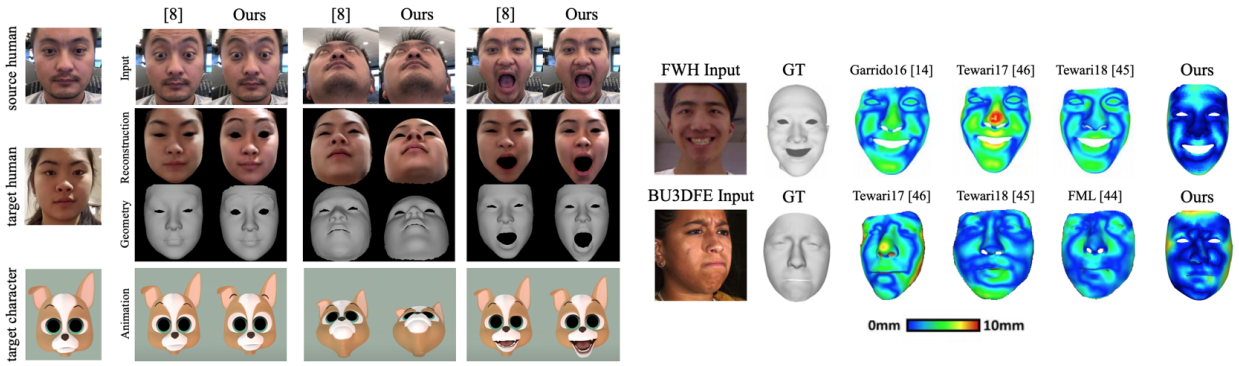


Figure 4.7: (a) Tracking comparison with [22]. (b) 3D reconstruction error maps.

4.4.4 Quantitative Evaluation

3D face reconstruction: We compute 3D geometry reconstruction error (root mean squared error between a predicted vertex and ground truth correspondence point) to evaluate our predicted mesh on 3D scans from BU-3DFE [160] and Facewarehouse (FWH) [16]. For BU-3DFE we use both the views per scan as input, and for FWH we do not use any special template to start with, unlike Asian face template used by FML. Our personalized face modeling and novel constraints together result in lower reconstruction error compared to state-of-the-art methods (Table 4.3a and Fig. 4.7b). The optimization-based method [48] obtains 1.59mm 3D error for FWH compared to our 1.68mm, but is much slower (120s/frames) compared to our method (15.4ms/frame). We also show how each component of our method helps in improving the overall output in Table 4.2. For photometric error, we used 1000 images of CelebA [90] (referred as CelebA*) dataset to be consistent with [138].

Face tracking: We evaluate the tracking performance of our method using two metrics: 1) Normalized Mean Error (NME), defined as an average Euclidean distance between the 68 predicted and ground truth 2D landmarks normalized by the bounding box dimension, on AFLW2000-3D dataset [179], and 2) Area under the Curve (AUC) of the cumulative error distribution curve for 2D landmark error [31] on 300VW video test set [127]. Table 4.3b shows that we achieve lower

landmark error compared to state-of-the-art methods although our landmarks are generated by a third-party method. We also outperform existing methods on video data (Table 4.3c). For video tracking, we detect the face in the first frame and use the bounding box from previous frame for subsequent frames similar to [22]. However, the reconstruction is performed on a per-frame basis to avoid inconsistency due to the choice of random frames.

Facial motion retargeting: In order to evaluate whether our tracked facial expression gets correctly retargeted on the target model, we use the expression metric defined as the mean absolute error between the predicted and ground truth blendshape coefficients as in [22]. Our evaluation results in Table 4.4 emphasize the importance of personalized face model in improved retargeting, since [22] uses a generic 3DMM as its face model.

Table 4.2: Ablation study. Evaluation of different components of our proposed method in terms of standard evaluation metrics. Note that B and C are obtained with all the loss functions other than the parsing loss and the gradient loss.

Method	**3D error (mm)**				**NME**	**AUC**	**Photo error**
	BU-3DFE		**FWH**		**AFLW2000-3D**	**300VW**	**CelebA***
	Mean	SD	Mean	SD	Mean	Mean	Mean
3DMM prior (A)	2.21	0.52	2.13	0.49	3.94	0.845	22.76
A + blendshape corrections (B)	2.04	0.41	1.98	0.44	3.73	0.863	22.25
B + albedo corrections (C)	1.88	0.39	1.85	0.41	3.68	0.871	20.13
C + parsing loss (D)	1.67	0.35	1.73	0.37	3.53	0.883	19.49
D + gradient loss (final)	1.61	0.32	1.68	0.35	3.49	0.890	18.91

Table 4.3: Quantitative evaluation with state-of-the-art methods. (a) 3D reconstruction error (mm) on BU-3DFE and FWH datasets, (b) NME (%) on AFLW2000-3D (divided into 3 groups based on yaw angles), (c) AUC for cumulative error distribution of the 2D landmark error for 300VW test set (divided into 3 scenarios by the authors). Note that higher AUC is better, whereas lower value is better for the other two metrics.

(a)

Method	**BU-3DFE**		**FWH**	
	Mean	SD	Mean	SD
[140]	1.83	0.39	1.84	0.38
[141]	3.22	0.77	2.19	0.54
[138]	1.74	0.43	1.90	0.40
Ours	**1.61**	**0.32**	**1.68**	**0.35**

(b)

Method	[0-30°]	[30-60°]	[60-90°]	Mean
[179]	3.43	4.24	7.17	4.94
[8]	3.15	4.33	5.98	4.49
[42]	2.75	3.51	4.61	3.62
[22]	2.91	3.83	4.94	3.89
Ours	**2.56**	**3.39**	**4.51**	**3.49**

(c)

Method	Sc. 1	Sc. 2	Sc. 3
[156]	0.791	0.788	0.710
[169]	0.748	0.760	0.726
[31]	0.847	0.838	0.769
[22]	0.901	0.884	0.842
Ours	**0.913**	**0.897**	**0.861**

Table 4.4: Quantitative evaluation of retargeting accuracy (measured by the expression metric) on [22] expression test set. Lower error means the model performs better for extreme expressions.

Model	Eye Close	Eye Wide	Brow Raise	Brow Anger	Mouth Open	Jaw L/R	Lip Roll	Smile	Kiss	Avg
(1) Retargeting [22]	0.117	0.407	0.284	0.405	0.284	0.173	0.325	0.248	0.349	0.288
(2) Ours	0.140	0.389	0.259	0.284	0.208	0.394	0.318	0.121	0.303	**0.268**

Chapter 5

JOINT AUDIO-VIDEO DRIVEN FACIAL EXPRESSION RETARGETING

5.1 Introduction

The goal of this work is to learn a shared embedding that encodes the intrinsic content representation from both audio and video modalities for efficient 3D face animation, especially the lip/mouth movement. In other words, we care about improving the speech driven animation through leveraging such a content representation. Guided by this design principle, we propose an end-to-end framework to harvest the disentangled representations through both unsupervised learning (via auto-encoder) and cross-modality self-supervised learning (through minimizing the disagreement between the two modalities). However, merely using the content representation as a latent space feature in existing 3D face reconstruction networks like [21] will result in an incomplete face without eyes, mouth interior, hair and background. Inspired by [44, 45], we leverage the recent advancements in neural radiance fields (NERFs) to represent the geometry and appearance of a 3D face as a neural network that can be sampled at points in space. A ray-marching approach together with a volumetric integration scheme similar to [98] is used to sample from the neural network to render the reconstructed face. The dynamic NERF is conditioned on the learned content representation in addition to the sample position and view direction of standard NERF. During each training iteration, our end-to-end framework takes a few frames of a user's video and the corresponding speech segment as input and creates separate embeddings for the audio and video modalities. The two embeddings are then linearly combined using learnable weights to create a shared representation that controls the 3D face of the user at the output of the dynamic NERF. As is evident from our current pipeline, we need to re-train our network for every new user, which is inefficient and time consuming. To overcome this limitation, we propose to utilize model-agnostic meta learning (MAML) [43] to quickly adapt the network parameters for any new user. Recently

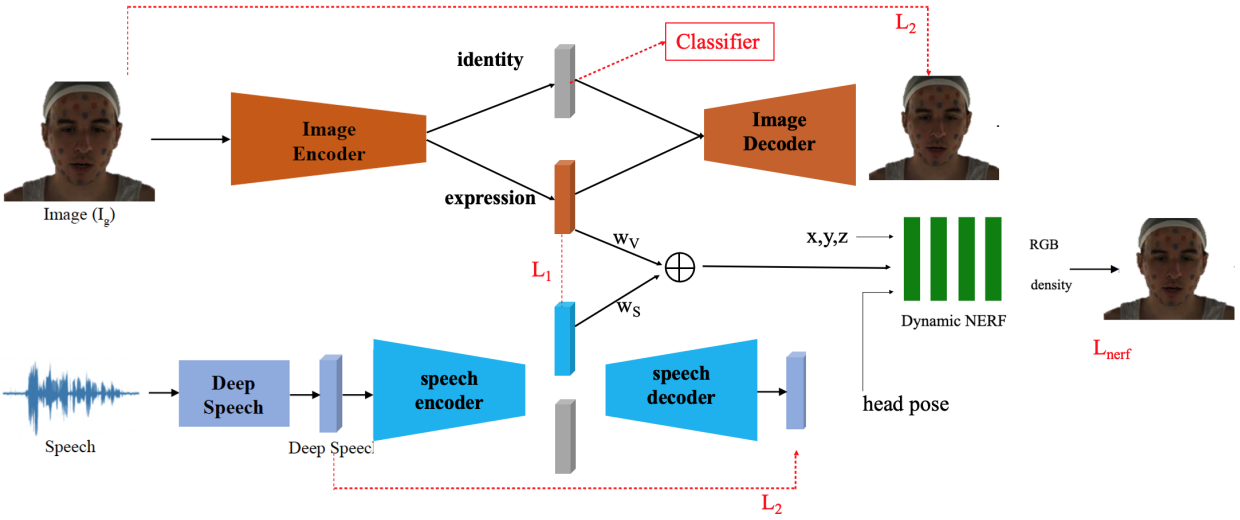


Figure 5.1: End-to-end framework of our method. Our network comprises of two parallel autoencoders which encode the input audio and video into content embeddings. A weighted average of the embeddings is then taken to create a shared embedding that becomes a conditional input to a neural radiance field. We use meta learning updates to adapt the network parameters for every new user.

MAML has been successfully used in tasks like object tracking [148], face recognition [53], and face antispoofing [110]. Our main contributions are:

1. We propose a novel end-to-end framework that extracts a compact representation from both audio and video modalities from talking user videos and uses it as a conditional input to a dynamic neural radiance field to create 4D user avatars.

2. Our architecture efficiently disentangles the facial motion information from the identity using cross-modality self-supervised learning.

5.2 Methodology

Fig. 5.1 shows the end-to-end framework of our approach and consists of three major components: (a) a *cross-modality shared embedding*, which consists of two parallel autoencoders encoding the identity and content information from the audio and video modalities, to eventually create a shared

embedding of the facial motion, (b) a *neural face representation*, which is a multilayer perceptron that takes the shared embedding as a conditional input in addition to other standard NERF inputs to create a 3D face at the output, and (c) a *meta learning setup*, which considers our end-to-end network trained on a single subject as a task and performs meta updates of the network for every new user. We now explain each of these components in more detail.

5.2.1 Cross-modality Shared Embedding

Our first aim is to disentangle the identity from the facial motion for each of the modalities, since the identity information is irrelevant for facial motion retargeting purposes. To achieve this, we utilize standard autoencoders, which can be trained in an unsupervised manner. Given an image I_g, the image encoder would output both the content feature embedding (denoted as V_c) and visual feature embedding (denoted as V_r). To avoid information loss, the image decoder takes both V_r and V_c as input and outputs another image I_p, which is supposed to reconstruct the input I_g. Likewise, for the speech modality, the encoder outputs both the content feature embedding (denoted as S_c) as well as the other speech feature embedding (denoted as S_r). The goal is also to reconstruct the input signal S_g from the speech decoder output denoted as S_p. Therefore, we have to minimize the reconstruction loss L_{rec} for both modalities, given by:

$$L_{\text{rec}} = (I_p - I_g)^2 + (S_p - S_g)^2 \tag{5.1}$$

The speech signal S_g is generated by passing the input speech segment through DeepSpeech [56] as in VOCA [30]. We then take a weighted average of the content embeddings of both modalities to get the shared embedding as:

$$E_c = \alpha_V * V_c + \alpha_S * S_c \tag{5.2}$$

where the weights α_V and α_S sum to 1 and are learned along with the entire network during the training.

However, an autoencoder alone cannot guarantee the disentanglement of identity from facial motion. Hence we add a separate classifier R to ensure that V_r embeds the identity information. The classifier is essentially a state-of-the-art face recognition network that is first trained on all

identities in our training data and then fixed during the training of our framework. The loss L_{cl} between the classifier output identity P_R and the ground truth identity P_g is the standard cross-entropy loss of face recognition tasks. Interestingly, such a classifier cannot be used for S_r, since S_r embeds more information than just identity, like background noise etc. Hence, we try to ensure that S_c encodes only the necessary content information. We perform this by using an L1 loss between V_c and S_c:

$$L_{\text{cross}} = ||V_c - S_c||_1 \tag{5.3}$$

5.2.2 Neural Face Representation

The 3D information of the input user face is represented here by a dynamic radiance field, which in turn is represented by a multi-layer perceptron (MLP) f_θ as:

$$f_\theta(\mathbf{p}, \overrightarrow{d}, E_c) = (RGB, \sigma) \tag{5.4}$$

where θ contains the MLP parameters, $\mathbf{p}$ is the position (x,y,z coordinates), $\overrightarrow{d}$ is the viewing direction, and E_c is the shared embedding vector from eq. 5.2. We use positional encoding for $\mathbf{p}$ and $\overrightarrow{d}$ as in [44]. In order to get the viewing direction, we estimate the rigid head pose of the input video frame using our method described in Chapter 3 and then use it to transform the camera space coordinates to the world coordinates of the head.

Volumetric rendering is used to render the learned 3D face. Specifically, we cast rays through each pixel of the input image, sample random points along these rays, and accumulate the predicted RGB values and density at the sampled points to get the pixel color C as:

$$C(\overrightarrow{r}; \theta, \mathbf{R}, \mathbf{t}, E_c) = \int_{z_1}^{z_2} \sigma_\theta(\overrightarrow{r}) \cdot RGB_\theta(\overrightarrow{r}, \overrightarrow{d}) \cdot \tau(t) dt \tag{5.5}$$

where $\overrightarrow{r}$ is the camera ray, $\mathbf{R}$ and $\mathbf{t}$ are the head rotation and translation, σ_θ and RGB_θ are predicted by eq. 5.4, and $\tau(t)$ is the transmittance accumulated from point z_1 to t along the ray:

$$\tau(t) = \exp(-\int_{z_1}^{t} \sigma_\theta(\overrightarrow{r}(s)) ds) \tag{5.6}$$

The NERF is trained on photometric reconstruction loss given by:

$$L_{\text{nerf}} = \sum_{j} ||C(\overrightarrow{r}; \theta, \mathbf{R}, \mathbf{t}, E_c) - I_g[j]||^2 \tag{5.7}$$

where the sum is taken over all pixels j in the image. The total loss function for the end-to-end network training is thus:

$$L = L_{\text{rec}} + L_{\text{cl}} + L_{\text{cross}} + L_{\text{nerf}} \tag{5.8}$$

5.2.3 Meta Learning Setup

As discussed earlier, at every training iteration, our network takes a particular user's video as input and learns to generate a 3D face for that user. This means the network needs to be re-trained for every new user, which is highly inefficient. In order to automatize this process to some extent, we consider the network learning for a single user as a *task* in a model-agnostic meta learning setup. For each such task, the support set is a set of frontal view frames exhibiting different facial expressions, and the query set is the set of all other views. Hence for a new user, it is enough to update the trained network parameters by capturing a few frontal views of the user expressions. The basic algorithm contains two loops:

- The inner loop computes $\nabla_{\hat{\theta}} L_u(\cdot)$ for user u over the support set and updates $\hat{\theta}^n$ at each meta-training iteration n, where $\hat{\theta}$ are the parameters of the end-to-end network.

- The outer loop computes $\nabla_{\hat{\theta}} \sum_u L_u(\cdot)$ over the query set and updates the optimal network parameters $\hat{\theta_*}^n$ at each meta-testing iteration n.

5.3 Experimental Setup

5.3.1 Datasets

We train our network using two datasets: 1) VoxCeleb2 [27] and 2) the VOCA dataset collected by Cudeiro et al. [30]. VoxCeleb2 has more than 140k videos of about 6000 identities collected from the Internet, but the videos are mostly similar. So, we choose a subset of 90k videos from

about 4000 identities. We sample the videos at 10fps to avoid multiple duplicate frames, randomly delete frames with neutral expression and pose based on a threshold on the expression and pose parameters predicted by [22], and then crop the face and extract ground truth 2D landmarks using [22]. The VOCA dataset consists of 4D face scans captured at 60 fps with the corresponding speech and RGB images. There are 12 identities and each speaks 40 sentences of roughly 3-5 seconds each. All the face scans are aligned to a common template mesh making the scans in correspondence. We sample the VOCA videos similarly and all cropped faces are finally resized to 224×224. We train only on VoxCeleb2, and VOCA is used as an additional test dataset to evaluate the performance of our method on 3D meshes.

5.3.2 Implementation Details

The inputs to our network during an iteration is a pair consisting of a video frame and the corresponding synchronized raw speech segment. The images are channel-wise standardized using the mean and standard deviation values of the training set. At the same time, each raw speech window corresponding to a video frame is first transformed into a DeepSpeech [56] frame of size 16×29 using a pretrained DeepSpeech network. The image encoder and decoder consist of standard convolutional and transpose convolutional layers respectively like U-Net [120], with the bottleneck linearizing the 2D feature into a vector for the embedding. The speech encoder and decoder consist of fully-connected layers. All layers are followed by batch normalization and ReLU activation. The dynamic NERF contains 8 fully-connected layers of 256 neurons each up to the prediction of density, following which there are 4 fully-connected layers of 128 neurons each to predict the RGB values. We pass 1000 rays through the image pixels, and sample 100 points along each ray to compute the aggregate. We implemented our network using PyTorch [107] and trained it for 200K iterations using the Adam optimizer with a constant learning rate of 10^{-4}.

5.4 Results

In this section, we present the study of the importance of each of our components, and qualitative and quantitative comparisons of our method with state-of-the-art approaches.

Table 5.1: Ablation study for the importance of each component of our network.

Method	**PSNR**
Audio only	35.74 $\pm$ 0.12
Video only	37.83 $\pm$ 2.83
Audio + Video	38.54 $\pm$ 2.97
Audio + Video + L_{cross}	38.91 $\pm$ 3.03
Audio + Video + L_{cross} + L_{cl}	39.14 $\pm$ 2.46

5.4.1 Ablation Study

Our initial hypothesis regarding the advantage of using both audio and video modalities in improving the face reconstruction quality is proved in Table 5.1. It is important to note that our learned weights in eq. 5.2 help the network to automatically decide the importance of the two modalities at any particular instant. For example, during instants when the speech is ambiguous but the facial movement is minimal, α_S is greater than α_V to ensure proper lip synchronization. On the other hand, during instants when the user is not speaking but expressing a strong emotion like laughter or surprise, α_V is greater than α_S in order to capture the expression-specific facial deformations, since the speech signal is negligible. The regularization losses L_{cross} and L_{cl} also contribute towards the improvement in the results by ensuring proper disentanglement of the user-independent facial motion from the user-dependent information.

The meta learning setup for training the network significantly reduces the computational complexity when a new test user video is considered. However, the results are slightly sensitive to the choice of support and query sets for the meta learning iterations. It is desirable to have a wide variety of facial expressions in the support set for the network to succeed reasonably on the query set. Automatically choosing keyframes from an input user video is an interesting direction to explore in future. We also observed that unlike [44] which relies on sparse expression blendshape coefficients predicted by a third party method, our approach of learning the expression embedding in an end-to-end fashion captures the subtleties of the facial expressions better.

Table 5.2: Quantitative evaluation of our method in comparison with state-of-the-art approaches.

Method	$L_1 \downarrow$	**PSNR** $\uparrow$	**SSIM** $\uparrow$	**LPIPS** $\downarrow$
FOMM [128]	0.037	25.13	0.90	0.18
DVP [76]	0.031	26.41	0.91	0.17
NERF-face [44]	0.022	27.34	0.93	0.13
Ours	**0.018**	**28.87**	**0.94**	**0.11**

5.4.2 Quantitative Comparison

In addition to face reconstruction, facial retargeting is also a desired application of our method. In order to retarget the facial motion from a source actor to a target actor, we need to provide the video frames of the source actor to the image encoder to encode the motion, but the input to the dynamic NERF will be video frames of the target actor. Hence, we can compare our method's performance with state-of-the-art face reenactment methods like First-Order Motion Models (FOMM) [128], Deep video portraits (DVP) [76] and NERF-face [44]. As evaluation metrics, we use the standard ones used by other methods - a) L_1 distance, b) Peak Signal to Noise Ration (PSNR), b) Structural Similarity Index (SSIM) [149] and Learned Perceptual Image Patch Similarity (LPIPS) [170]. Table 5.2 shows that our method outperforms the existing methods in all the metrics, which attributes to the fact that the complementary information from dual modality helps to capture finer details in the facial deformations.

5.4.3 Qualitative Evaluation

Fig. 5.2 shows the reconstructed faces for two users rendered with different head poses and expression references. The head pose and expression references are taken from another user video randomly selected from the dataset. The results show that our network reconstruct the face with high quality and effectively uses the head pose and facial motion as conditional inputs to make the desired changes in the results during retargeting. We also compare our results with that of VOCA [30], which is audio-only, in Fig. 5.3. We first register the ground truth neutral mesh of the user to

Figure 5.2: Renders of the 3D faces reconstructed by our network given sample frames of the user video and corresponding audio as input. (a) user 1, (b) user 2. Our reconstructed faces are consistent across multiple views and expression variations.

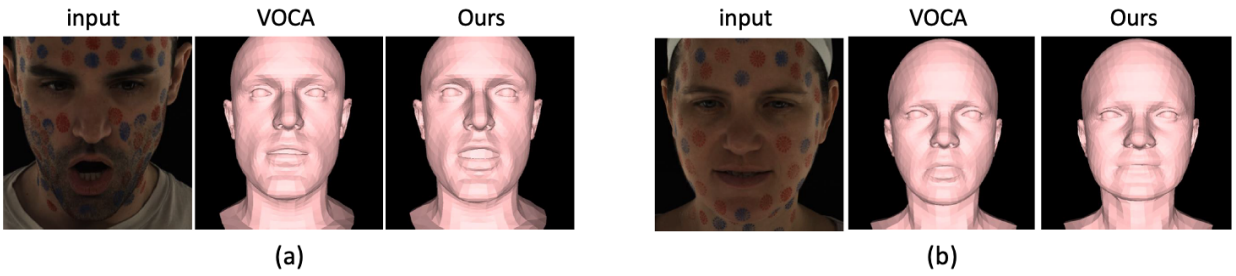


Figure 5.3: Comparison of our method on 3D meshes with respect to VOCA [30]. (a) user 1, (b) user 2.

our network-predicted density map using the iterative closest point algorithm [7] and then compute the point-to-point error between the 3D vertices. We observe that the additional visual information from the video frames solve the ambiguity of complex speech segments. Also in the majority of cases, the improvement is mostly in the lip area movement as expected. The point-to-point error of VOCA is 1.68 ± 0.2 whereas for our method is 1.53 ± 0.3.

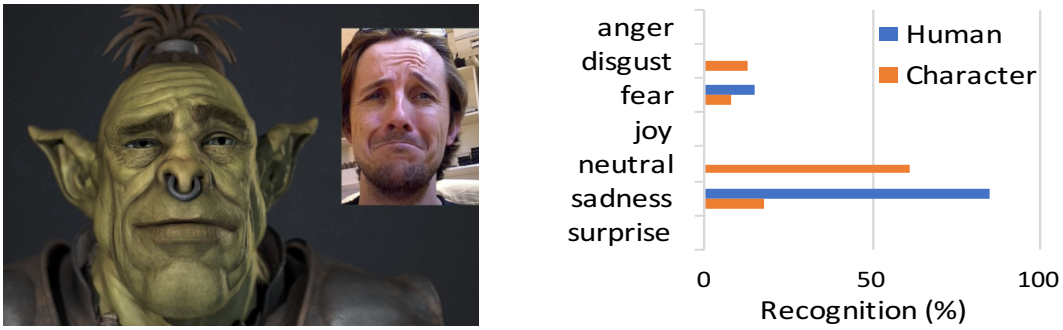
Chapter 6

LEARNING TO GENERATE 3D STYLIZED CHARACTER EXPRESSIONS FROM HUMANS

6.1 Introduction

Characters must have *perceptually valid* expressions, that are clearly perceived by humans to be in the intended expression class. Fig. 6.1 shows a concrete example of a perceptually invalid expression, in which the human expression did not transfer correctly to the character when tested on Mechanical Turk (MT) for expression clarity with 30 test subjects. Our goal is to learn 3D stylized character expressions from humans in a *perceptually valid* and *geometrically consistent* manner. To this end, we propose an end-to-end system, ExprGen, that takes a 2D image of a human and predicts the 3D rig parameters of a character. This is a challenging goal because there is no existing dataset mapping 2D images of human expressions to 3D character rig parameters. Further, it is prohibitively expensive to manually create a dataset with explicit human image to 3D rig parameter labeling. To address this challenge, we leverage publicly available human and character expression datasets with 2D images and expression labels [33, 92, 106, 93, 97, 5]. Each 2D character image in the dataset is rendered from its 3D facial rig (which has associated rig control parameters). In our work, the system learns from the six broad categories (anger, disgust, fear, joy, sadness, surprise) [37] and neutral, since there is agreement on their recognition within the facial expression research community, and these seven expressions occur in a wide range of intensities and can blend with each other to create additional expressions.

Our approach is to learn the correspondence between 2D images of human and character expressions and use this correspondence to map human expression images to 3D character rig parameters. We start with recognizing facial expressions for both humans and characters to induce a perceptual metric and constraints for expression generation and matching. Our system then learns



(a) (b)

Figure 6.1: Example of inaccurate expression transfer. (a) Expression transfer from human (top right) to a character [39]. (b) Mechanical Turk testers perceive the human expression as sadness, while the character expression is perceived as neutral and a mixture of other expressions. The character expression has neither expression clarity nor geometric consistency.

a joint embedding to map human expressions to character expressions. This mechanism also accounts for geometric consistency, so that any mapping is both perceptually valid and geometrically consistent. At this point, we can take any human expression image and find similar or dissimilar images in a database of character images. Using this similarity analysis, we train a regression network, 3D-CNN, from a human expression onto the parameters of a specific or *primary* character 3D rig. Finally, a lightweight mechanism, *Character Multi-Layer Perceptron* (C-MLP), transfers character expressions to other characters. This enables re-use of a primary character rig trained in the previous steps to drive *secondary* characters and eliminates the need to individually train each rig with the full pipeline. Fig. 6.2 depicts an overview of our system at run time. Images of human facial expressions are initially processed to detect the face and 49 geometric landmarks, and then fed into the 3D-CNN to generate expression specific parameters for the primary character rig. The C-MLP uses the generated expression parameters to produce the expression on other secondary 3D stylized characters. Both qualitative and quantitative results detailed in Sec. 6.3 illustrate the accurate, plausible, and perceptually valid expression transfer from humans to 3D stylized characters. We hope that our method will enable the creation of new perceptually driven tools that

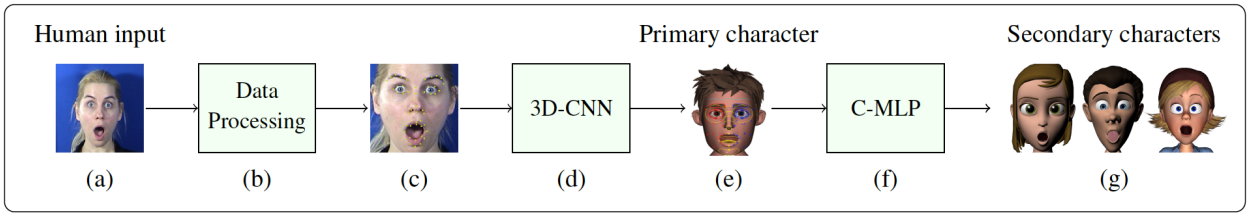


Figure 6.2: Overview of our multi-stage expression transfer system ExprGen. 2D images (a) of human facial expressions are preprocessed (b, c) and used to train a CNN (d), which generates rig parameters corresponding to the human expression for primary characters (e). A separate neural network (f) performs character-to-secondary-character expression transfer (g).

help animators create clear and accurate character expressions and ultimately successful animated stories. The main contributions of our work are:

1. We propose a novel perceptually valid method to map 2D human face images to 3D stylized character rig controls.
2. We use both geometric consistency and perceptual validity rather than pure geometry-based mathematical operations to generate 3D characters with clear unambiguous facial expressions.
3. We present a semi-supervised method as a light-weight extension to enable expression transfer between multiple characters.

6.2 Methodology

In order to build a system that can transfer human expressions to multiple 3D characters, we need several components to handle the human-to-character transfer in 2D, produce parameters for a primary character expression in 3D including both perceptual and geometric similarity, and transfer the expression of a primary character to multiple secondary characters. We build a multi-stage deep learning system ExprGen with two major components: Training from 2D Datasets and 3D Expression transfer.

6.2.1 Training from 2D Datasets

We combine five publicly available labeled facial expression databases to create the Human Expression Database (HED): (a) Static Facial Expressions in the Wild (SFEW) database [33], (b) Extended Cohn-Kanade database (CK+) [92], (c) MMI database [106], (d) Karolinska Directed Emotional Faces (KDEF) [93], and (e) Denver Intensity of Spontaneous Facial Actions (DISFA) database [97]. The HED consists of approximately 100K labeled images; the number of samples for each class is balanced to avoid bias towards a particular expression. Specifically, we undersampled the neutral class, so that its distribution is the same as the other expression classes. To preprocess our human input as shown in Fig. 6.2(a-c), we extract 49 facial landmarks [155] to register a face to an average frontal face via an affine transformation and use the landmarks to extract the geometric features including the following measurements: left/right eyebrow height (vertical distance between top of the eyebrow and center of the eye), left/right eyelid height (vertical distance between top of an eye and bottom of the eye), nose width (horizontal distance between leftmost and rightmost nose landmarks), mouth width (left mouth corner to right mouth corner distance), closed mouth measure (vertical distance between the upper and the lower lip), and left/right lip height (vertical distance between the lip corner from the lower eyelid). Each distance is normalized by the bounding box of the face. We extract the geometric features for the character images in the same manner. Once the faces are cropped and registered, the images are re-sized to 256 $\times$ 256 pixels for input to our network training. For the Character Expression Database (CED), we use FERG-DB [5] which consists of 55,767 labeled face images of six stylized characters (‘Mery’, ‘Aia’, ‘Bonnie’, ‘Jules’, ‘Malcolm’ and ‘Ray’) for training. In addition to FERG-DB, we add three new characters (‘Tuna’ [159], ‘Mathilda’ and ‘Cody’ [1]) for validation. An animator created 10 key poses per expression for each new character and labeled them using Mechanical Turk (MT) with 70% agreement among 50 test subjects. We also obtained the stylized 3D rigs modeled using the Autodesk®MAYA software for all the characters and used the control parameters associated with them in our work.

The goal of this step is to learn a joint embedding between human and primary character ex-

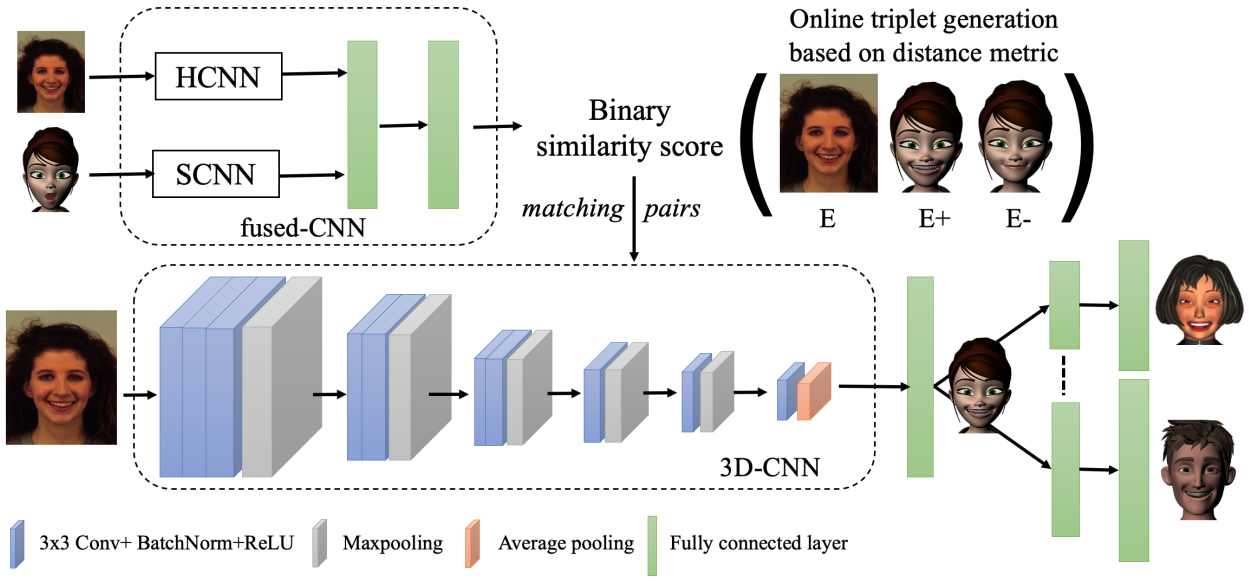


Figure 6.3: Network architecture of our proposed approach. We first learn a shared embedding space between human and character expression images. Once we obtain matching pairs from this space, we train a CNN to take a human face image as input and generate primary rig parameters as output. We can then simply use multilayer perceptrons to transfer the input expression from a primary to a seconday character.

pressions based on perception and geometry. We first train the Human CNN (HCNN) on the HED to classify an input human face image into one of the seven expression classes. Then, we initialize the weights of the Shared CNN (SCNN) with those of HCNN except for the Fully Connected (FC) layers and fine-tune the SCNN on FERG-DB by transfer learning [162] to create a shared embedding feature space. The network architecture for the HCNN and SCNN is similar to AlexNet [79]. After they are trained independently, we concatenate the outputs from their average pooling layers and send it to a network of two FC layers to form a Pseudo-Siamese network [26, 165] called the fused-CNN (f-CNN) as shown in Fig. 6.3. To train the f-CNN, we introduce a similarity measure based on the distance between two image encodings as follows. After the HCNN predicts the perceptual expression label of the human input image, the FERG-DB is searched to retrieve the character images having the same predicted label. Then, the Euclidean distance between the geometric feature vector of the human image and those of all the retrieved character images are

computed and ordered based on the distance to the human image. Note that we did not always find perfect matches, but our dataset is large enough to enable the CNNs to learn generalizable matching representations between human and character images. To solve the issue of incorrect geometry match within the same expression class, triplets $(E, E+, E-)$ of training images are created where E is a reference human expression image, $E+$ is a character image similar to the reference human expression image (best geometry match in the search), and $E-$ is another image that is not geometrically and/or perceptually similar to the reference human expression image. For example, if E is an open mouth joy human expression, then character anger retrieval would be incorrect perceptually and closed mouth human joy would be incorrect geometrically and both will be $E-$.

The f-CNN takes a human expression image and primary character expression image and produces a similarity score by minimizing a loss function consisting of a hinge-based loss term and a squared *L2*-norm regularization term [125]:

$$\min_w \sum_{i=1}^{N} \max(0, 1 - l_i y_i^{net}) + \frac{\lambda}{2} \|w\|_2 \tag{6.1}$$

where y_i^{net} is the network output for the i^{th} training sample, $l_i \in \{-1, 1\}$ is the corresponding label (with +1 and -1 denoting a non-matching and a matching pair, respectively) and w are the weights of the neural network. The hinge loss minimizes the distance between E and $E+$ (matching both geometry and perception) and maximizes the distance between E and $E-$ (mismatching the geometry and/or perception). Similar to the approach described in [125], triplets are generated online by selecting the hard positive/negative exemplars from within a mini-batch for our training. The softmax layer at the end of f-CNN converts the similarity score to binary classification (similar or dissimilar).

6.2.2 3D Expression Transfer

This step generates perceptually valid 3D characters from human expression images. It is divided into two stages: expression transfer from human to a primary character rig and expression transfer from primary to secondary character rigs. The stages are described as follows:

Human to Character Transfer We aim to control the 3D primary rig by predicting rig parameters given an input human image. So we train another CNN called the 3D-CNN which has the same configuration as HCNN or SCNN except for the dimensionality of the FC layers. Instead of seven probabilities for classifying seven expression classes, the final output is the parameters for the primary character. We initialize the weights of the 3D-CNN by trained HCNN weights so that we can transfer the knowledge learned from the HED, and the model does not overfit the 3D-parameters dataset. The pairs of a human input image and the 3D-parameters corresponding to the 2D character image with similar expression (as obtained at the output of f-CNN) are used for training the 3D-CNN (Fig. 6.2(d)). All the networks are trained end-to-end using the Torch framework [28] using stochastic gradient descent with hyper parameters (momentum of 0.9, weight decay of 0.0005 and a batch size of 50) on a single NVIDIA GTX-1080 GPU. In order to make sure that the pre-trained weights are not drastically changed, the learning rate for the SCNN, f-CNN and 3D-CNN is set lower (0.0001) than that of the HCNN (0.001). The learning rate was dropped by a factor of 10 after every 10 epochs of training. We used Batch normalization [64] and ReLU activation after every convolutional layer. To avoid overfitting, our training data is augmented by horizontal flipping, rotating, and random cropping followed by scaling. We used an 80:10:10 split for training, validation and test sets, and performed 5-fold cross validation.

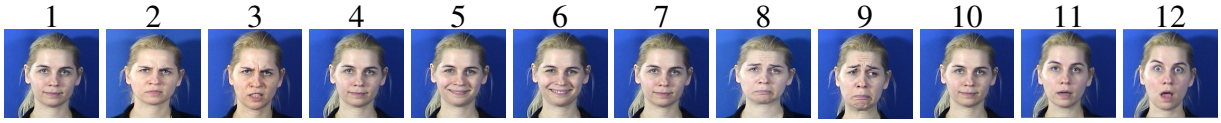
Character to Character Transfer ExprGen is trained for a primary character rig, and we propose a lightweight alternative to training a different network for each new secondary character as shown in Fig 6.2(e-g). Due to the absence of one-to-one correspondence between the facial control points on different rigs, manual mapping of the rig parameters is often not possible. Our character-to-character expression transfer model aims at automatically learning a function to map the 3D-parameters of the primary character to the secondary characters. For each secondary character we create a separate multilayer perceptron (MLP), which is a one-hidden-layer neural network with M input nodes, N output nodes and $\frac{1}{2}(M+N)$ hidden nodes with *tanh* activation, where M and N are the number of 3D-parameters of the primary and the secondary characters respectively. Gradient descent is used with a mini-batch size of 10 and a learning rate of 0.005 to minimize the square

loss between the input and output parameters. These networks (together called C-MLP) are trained in parallel and then augmented at the end of the 3D-CNN to map the input human expression simultaneously on multiple stylized characters.

We obtained pairs of training examples for the C-MLP by using a combination of two distance measures: d_{geometry} and $d_{\text{perception}}$. $d_{\text{geometry}} = ||f_p^g - f_s^g||_2$ is the Euclidean distance between the geometric feature vectors of the primary (f_p^g) and secondary character (f_s^g) image pairs, while $d_{\text{perception}} = ||f_p^p - f_s^p||_2$ is the Euclidean distance between the perceptual feature vectors (f_p^p and f_s^p) of the image pairs. The perceptual features are obtained by extracting the output of the last FC layer of the SCNN and normalizing it by the softmax weight as done in [5]. Given a primary character with an expression to find on a secondary character in FERG-DB (on which our SCNN is trained), all secondary character images in the CED having the same perceptual label as the primary character image are retrieved and ordered by the smallest value of d_{geometry}; the image with smallest distance value is returned. If the secondary character is not in FERG-DB, based on empirical evidence, the images of the secondary character for the perceptual labels having the two highest probabilities are retrieved and the combined function $\frac{1}{2}(d_{\text{perception}} + d_{\text{geometry}})$ is used to order them for retrieval. This methodology produces a set of matching (primary character, secondary character) pairs, for which we have both images *and* the 3D parameters that can be used to generate the 3D meshes from which those images are derived. The pairs of corresponding parameters are used to train the C-MLP. Once trained, the C-MLP transforms the 3D parameters of a primary character into the corresponding 3D parameters of a secondary character.

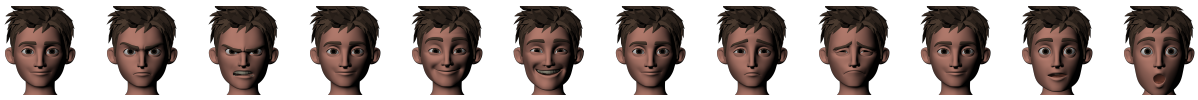
6.3 Results

We evaluated the performance of our system by computing the expression recognition accuracies of the HCNN and SCNN independently, testing the human-to-character expression transfer perceptual accuracy and comparing our results with Faceware (commercial product). In all the subsequent figures and tables, we show the 2D rendered images of 3D character rigs and use the following notation for the expression classes - A: anger, D: disgust, F: fear, J: joy, N: neutral, Sa: sadness, Su: surprise. The performance of our method on a video and the access to our database can be

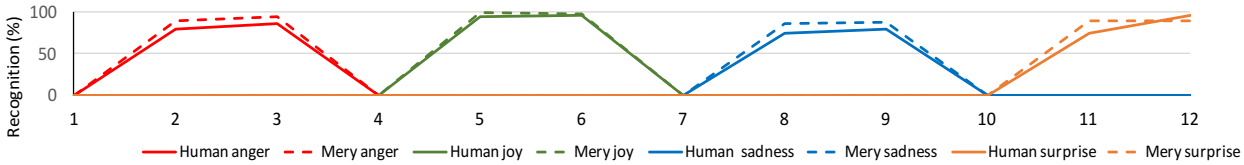


(a) Human expression sequence

(b) Primary character 'Mery'



(c) Primary character 'Ray'



(d) Plot showing percentage of 30 MT test subjects recognizing the correct expression on human and the character 'Mery'

Figure 6.4: Human to primary character expression transfer for human expression transition from neutral to joy, from neutral to surprise, from neutral to sadness, and from neutral to anger based on both perceptual and geometric similarity. (a) Human input expression frames (1-12), (b) Mapped expressions on 'Mery', and (c) Mapped expressions on 'Ray', (d) Expression recognition results between human (solid lines) and transferred expressions on 'Mery' (dashed lines) for different expressions.

Table 6.1: Average (%) expression recognition accuracy for 2D images of human and stylized character expressions when compared with the ground truth labels respectively. Note that the characters have higher expression clarity than humans due to their simpler geometry.

Class	A	D	F	J	N	Sa	Su
Human	76.27	63.81	68.47	94.31	78.03	72.95	92.26
Character	90.45	72.89	79.16	96.39	84.38	79.44	94.87

Perceived human expression (%)	Perceived character expression (%)						
	A	D	F	J	N	Sa	Su
A	71.32	16.28	5.43	1.55	3.10	0.78	1.55
D	14.29	67.35	4.08	1.02	4.08	8.16	1.02
F	2.88	6.47	64.03	2.16	3.60	3.60	17.27
J	0.92	1.83	0.92	90.83	1.83	0.92	2.75
N	1.09	3.26	2.17	4.35	76.09	10.87	2.17
Sa	1.80	3.60	2.70	1.80	18.02	71.17	0.90
Su	0.52	1.04	7.77	1.55	0.52	0.52	88.08

Figure 6.5: Confusion matrix for perceived transferred expression recognition (%) for seven expression classes.

found on the project webpage[1]. Video retargeting is performed frame-by-frame, and the predicted parameters are temporally smoothed using a Kalman filter.

6.3.1 Expression Recognition Accuracy

We first evaluated the HCNN and SCNN for the expression recognition task using the ground truth labels of HED and CED respectively in a 10-fold cross-validation setting. The HCNN and SCNN obtained average accuracies of 89.71% and 96.82%, respectively (Table 6.1). We note that our classification networks perform better than the prior networks trained for a similar classification task [5] because of training the HCNN on an additional dataset to learn the features in the wild and because of applying average pooling instead of max pooling after every convolution layer. We also observe that the character expression recognition accuracies are higher than humans, since the characters have simpler geometry and stylization can make the expressions relatively easier to perceive. Surprise and joy show high accuracy, while disgust and fear are more difficult for humans to both perceive and act out.

[1] http://grail.cs.washington.edu/projects/deepexpr/ferg-3d-db.html

6.3.2 Human to Character Expression Transfer

To evaluate our results for clarity in expression recognition and perceptual accuracy of the transferred expression, we asked 30 MT test subjects to recognize the input human expression and the generated primary character expression (output of 3D-CNN) for 1000 expression transfer results (approx. 150 for each expression class) on different stylized characters. We used the perceived label (as perceived by MT subjects) of the human as the ground truth and the perceived label of the character as the predicted output in its human-character-transfer pair. Fig. 6.5 shows the confusion matrix for transferred expression recognition for each expression class. For a given row (e.g. anger), the columns represent the percentage (averaged over all the perceived human anger expressions) of MT subjects agreeing on the corresponding expression classes for the transferred character expressions. The values show that ExprGen results in accurate transfer of expressions for most of the classes with an average correct perceptual recognition rate of 75.55%. The most common errors are confusion between disgust and anger, between fear and surprise, and between neutral and sadness. These errors are intuitively reasonable since the confused expressions have similar-looking geometric configurations. The least accurate expression transfer was for disgust and fear but as Table 2 shows, these expressions are difficult to recognize for both human and character images.

ExprGen generates expressions for multiple characters with high perceptual validity. The expression transfer results from a human to two primary stylized characters are shown in Fig. 6.4, which shows the generalizability of our algorithm in generating the same expressions on different characters having annotated training data. We tested the expression recognition on input human expressions and transferred expressions using 30 MT test subjects. The plot shown in Fig. 6.4d shows high correlation between MT agreement for recognized expressions on 'Mery' (Fig. 6.4b), which confirms the accurate perception of the intended expression transfer. We obtained a very similar plot for 'Ray' (Fig. 6.4c).

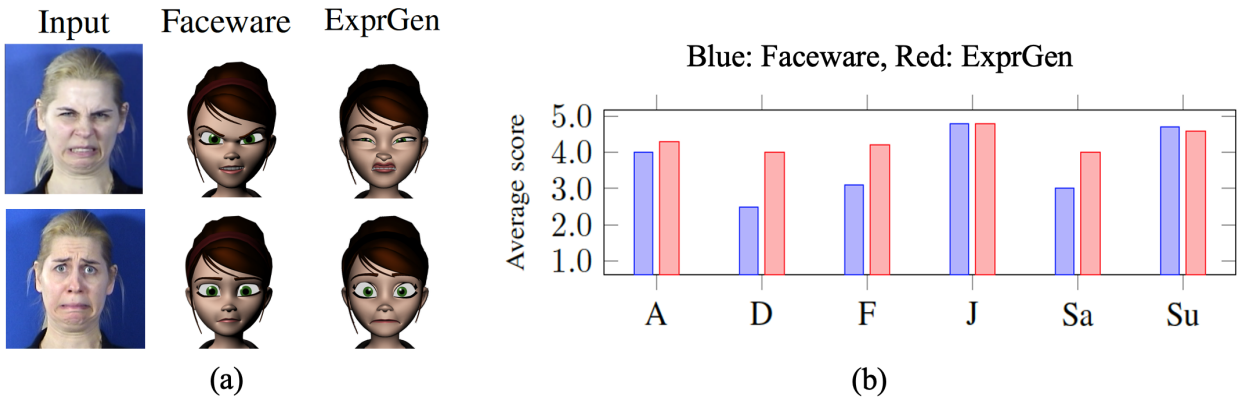


Figure 6.6: (a) Qualitative comparison of expression transfer results of Faceware and ExprGen (left to right: input human expression, Faceware output and ExprGen output), (b) Quantitative comparison of expression transfer results of Faceware (blue bars) and ExprGen (red bars).

6.3.3 Comparison with Faceware

ExprGen generates expressions with greater perceptual validity than popular commercially available software packages. We compared ExprGen with the award-winning Faceware technology [40], because it is the only feasible and comparable system that has the same input and output modality as ExprGen. Faceware includes *Analyzer* to convert human facial performance from a sequence of input images into motion capture data and *Retargeter* to map the captured data to the blendshapes of the 3D character face rig by manually creating an expression set for the character. Fig. 6.6b shows the comparison of average scores obtained for different expression classes when 30 MT test subjects were asked to rate the closeness of the expression generated on the character to the input human expression on a scale of 1-5, with 5 being the closest match. The average score over all classes for ExprGen is 4.31 versus an average score of 3.68 for Faceware. Fig. 6.6a shows the expression transfer results of Faceware and ExprGen for two cardinal expressions. These results show that blendshape-mapping-based approaches often produce incorrect expressions (row 1) or ambiguous expressions (row 2) owing to the limitations of correspondence mapping. We did not compare with the results of Faceshift Studio [39], since it requires a depth camera to capture human facial motion and uses a different approach compared to our 2D human image to 3D character rig

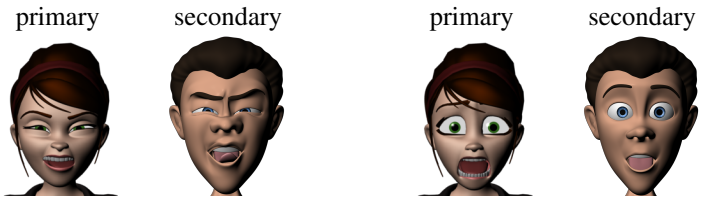


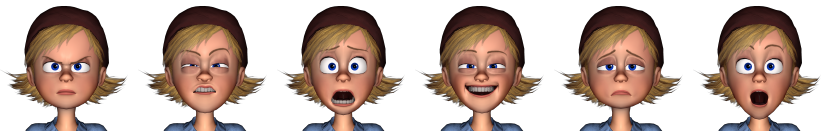
Figure 6.7: Error cases in obtaining training examples for new secondary characters. (a) Matching is perceptually valid (both expressions are disgust) but geometrically incorrect, (b) Matching is perceptually invalid (expression on left is fear and on right is surprise) but geometrically correct.

mapping.

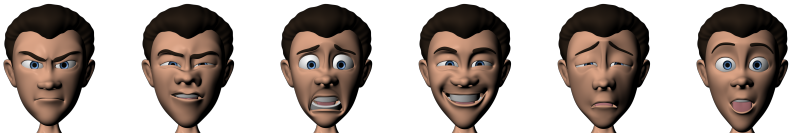
6.3.4 Character to Character Expression Transfer

In order to evaluate the performance of our character-to-character expression model, we selected 'Mery' as the primary character, 'Bonnie' as the existing secondary character (present in FERG-DB) and 'Tuna' and 'Cody' (non-human) as the new secondary characters (not present in FERG-DB). Fig. 6.8a shows six randomly chosen cardinal expressions on the primary character used as test cases, and Fig. 6.8 (b-d) show the facial expressions generated on the secondary characters at the output of the C-MLP. The results show that our network accurately learns the relationship between the 3D parameters of the characters, while maintaining the clarity of the expressions. Our network produces surprisingly good results for non-human characters as well, though the C-MLP is trained on only the key poses. However, the training examples for new secondary characters are critical to this approach, and there are two issues in selecting accurate training examples. First, when the new secondary character expression is perceptually valid but a similar expression does not exist for the primary character in the database (see Fig. 6.7a), our method retrieves the closest possible match which may be inaccurate. Second, when the new secondary character expression is perceptually ambiguous (see Fig. 6.7b), our method tries to find the closest match based on geometric features within the wrong expression classes, which may result in a wrong training

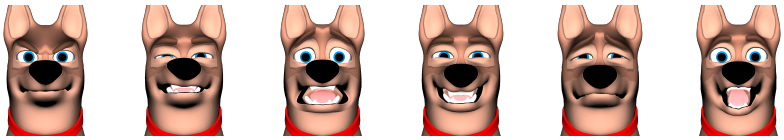
(a) Primary character ‘Mery’



(b) Existing secondary character ‘Bonnie’



(c) New secondary character ‘Tuna’



(d) New non-human secondary character ‘Cody’

Figure 6.8: Primary to Secondary character expression transfer results (left to right: anger, disgust, fear, joy, sadness and surprise). (a) ‘Mery’s’ expression classes, Expressions transferred to (b) ‘Bonnie’, (c) ‘Tuna’, and (d) ‘Cody’.

example. A possible future work can be to automate the process of generating large numbers of poses for each new character.

Chapter 7

LEVERAGING CYCLE-CONSISTENT GANS TO GENERATE 3D EXPRESSIVE CHARACTER RIGS

7.1 Introduction

The expression matching approach proposed in section 4.1 still has several limitations. Firstly, since the network is trained on only seven distinct expression categories, it does not generalize well to in-the-wild expressions belonging to categories which cannot be represented as a linear combination of the seven categories. Secondly, the approach controls only the 3D rig parameters, which can only manipulate the vertices in the local regions around their positions. Due to this lack of global constraint, the generated expressions may be erroneous at certain locations and need to be corrected through post-processing. Lastly, the multi-stage system makes the training process cumbersome, and overall accuracy of the approach relies on the accuracy of each individual component.

To overcome these limitations, we present a unified network that directly regresses the 3D vertex coordinates of the character rig given a human face image as input. Inspired by the unsupervised image-to-image translation approaches proposed in [175, 46], we use a cycle-consistent generative adversarial network (GAN) to accomplish our task and convert our 3D character rig into a 2D position map. Our network translates the input human expression image into a 2D position map of the character performing the same expression, which can then be converted into 3D vertex coordinates to generate the final 3D mesh. While perceptual validity is ensured by the features learned during image to image translation and by the 2D and 3D discriminators, we ensure geometric consistency by adding the 2D landmark loss during the forward cycle. Our main contributions are:

1. We propose a unified framework that directly regresses the 3D character vertex coordinates with the desired expression from an input image.

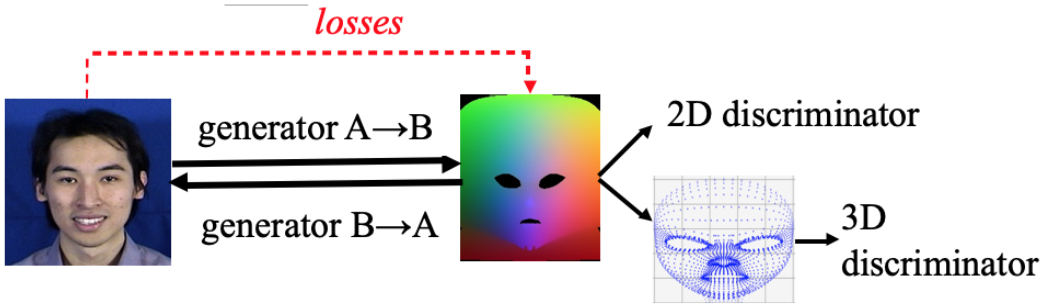


Figure 7.1: Schematic diagram of our proposed approach. We deploy a cycle-consistent GAN to take a human face image as input and predict the position map of the 3D character rig, which in turn generates the full 3D mesh of the character with the retargeted expression.

2. We combine the global representation of the input expressions via image-to-image translation with local representation via landmark constraints, and apply both 2D and 3D discriminators to ensure accurate 3D generated character expression.

7.2 Methodology

An overview of our approach is given in Fig. 7.1. Firstly, we represent the 3D face mesh of the target character in the form of a 2D UV position map. Each vertex of the 3D mesh corresponds to a point in the UV map when projected onto the 2D plane. The R, G and B values at a point in the UV map are then set to the values of the x, y and z coordinates of the corresponding 3D mesh vertex respectively to create the 3-channel position map. We define the domain A as the human face image domain, and domain B as the character position map. Generator A→B learns to translate an input human expression image to a 2D position map, and Generator B→A learns to convert the 2D position map back to the input human image. We train these networks using the following loss functions:

- Cycle-consistency losses (2D discriminator loss of both the domains and the reconstruction loss for each cycle).
- 3D discriminator loss. The generated 2D position map is converted back to 3D vertex coor-

dinates, but due to lack of depth information in the 2D domain, the generated 3D mesh may not be valid. We ensure the validity of the final generated mesh using the 3D discriminator.

- 2D landmark loss. The facial landmarks for the human image are obtained using [22], and the landmarks on the character are predefined by manually choosing the appropriate vertices of the neutral 3D mesh.

The generator and discriminator architectures are same as proposed in [175].

7.3 Experimental Setup

The human and character databases used in this approach are same as the HED and the CED mentioned in section 4.1.2.1. We implemented our networks using Pytorch [107] with Adam optimizer, learning rate 0.001 and batch size 1 (because of use of instance normalization). Note that our network is tailored to work with only the character 'Mery' at present, hence we only train on the 7324 images of 'Mery' expressions in our dataset.

7.4 Results

During testing, we only use the Generator A→B followed by UV map to 3D mesh conversion to retarget the expression of the input human image to the 3D character rig. Fig. 7.2 visually compares the retargeting results by our approaches proposed so far. It is interesting to note that while all the methods produce a result belonging to the correct expression category, the example-based approaches show more clarity in the perception of the expression. In fact, we observe that blendshape based approaches often require manual finetuning of the character blendshapes or blendshape weights in order to exactly replicate the input human expression, mainly because of the difference in geometry between humans and stylized characters. The results also demonstrate that our unified approach that combines global and local expression representations is able to capture more fine-grained details in the expressions (e.g. wrinkles on the forehead in row 1, mouth shape in row 4).

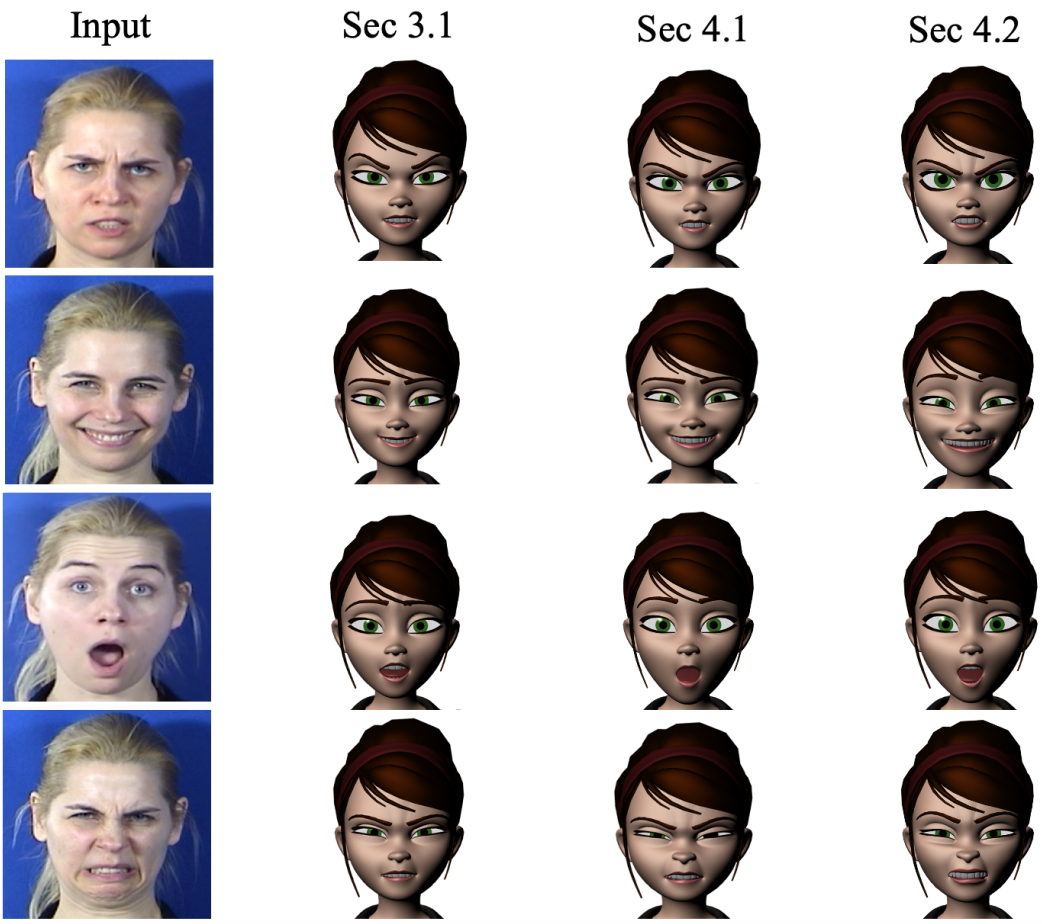


Figure 7.2: Qualitative comparison of facial expressions retargeted by our proposed approaches. We compare our latest approach with existing blendshape-based (proposed in Sec. 3.1) and example-based (proposed in Sec. 4.1) approaches.

Chapter 8

CONCLUSION

In this work, we propose blendshape-based approaches and example-based approaches specifically tailored for facial motion retargeting, as well as present a novel algorithm that can potentially further improve the efficiency of the proposed approaches. Facial motion retargeting has several applications including avatar animation for visual storytelling, live action video games, motion capture films, social AR/VR experience, virtual conversation using animated emoticons, human-robot interactions etc, as shown in Fig. 8.1. We believe that our approaches will have positive contributions in all these applications.

In our first approach, we propose a lightweight multitask learning network for joint face detection and facial motion retargeting on mobile devices in real time. The lack of 3DMM training data for multiple faces is tackled by generating weakly supervised ground truth from a network trained on images with single faces. We carefully design the network structure and regularization to enforce disentangled representation learning inspired by key observations. Extensive results have demonstrated the effectiveness of our design. In our second approach, we propose a novel deep learning based approach that learns a user-specific face model (expression blendshapes and dynamic albedo maps) and user-independent facial motion disentangled from each other by leveraging in-the-wild videos. Extensive evaluations have demonstrated that our personalized face modeling combined with our novel constraints effectively performs high-fidelity 3D face reconstruction, facial motion tracking, and retargeting of the tracked facial motion from one identity to another.

In our third approach, we propose a unified framework that combines 3D face information from both user expression video frames and user speech to create a more accurate personalized user face model. The information from audio and video are encoded together and used as a conditional input

video games motion capture films social VR experience

Figure 8.1: A few applications of facial motion retargeting.

to a dynamic neural radiance field for a complete 4D avatar generation at the output. The video embedding ensures accurate reconstruction of facial features whereas the audio embedding ensures accurate lip sync.

Our fourth approach demonstrates a novel multi-stage deep learning system to transfer human facial expressions to multiple 3D stylized characters that optimizes over expression clarity rather than over geometric markers. The resulting expressions, when validated by Mechanical Turk studies, show that our expression transfer clearly reproduces the input human expressions while retaining perceptual validity and geometric consistency. Our fifth approach shows a unified framework that combines global and local expression representations to directly regress 3D mesh vertices from an input human face image. Compared to our other proposed approaches, the fifth approach achieves the best clarity in the perception of retargeted expressions.

BIBLIOGRAPHY

[1] Character rigs for download. http://www.cgmeetup.net/forums/files/.

[2] Mass effect. https://www.masseffect.com.

[3] Real-time 3d face tracking based on active appearance model constrained by depth data. *Image and Vision Computing*, 32(11):860 – 869, 2014.

[4] Deepali Aneja, Bindita Chaudhuri, Alex Colburn, Gary Faigin, Barbara Mones, and Linda Shapiro. Learning to generate 3D stylized character expressions from humans. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[5] Deepali Aneja, Alex Colburn, Gary Faigin, Linda Shapiro, and Barbara Mones. Modeling stylized character expressions via deep learning. In *Asian Conference on Computer Vision (ACCV)*, 2016.

[6] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2), 2019.

[7] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[8] Chandraskehar Bhagavatula, Chenchen Zhu, Khoa Luu, and Marios Savvides. Faster than real-time facial alignment: A 3D spatial transformer network approach in unconstrained poses. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[9] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings SIGGRAPH*, pages 187–194, 1999.

[10] Sofien Bouaziz, Yangang Wang, and Mark Pauly. Online modeling for realtime facial animation. *ACM Transactions on Graphics*, 32(4), July 2013.

[11] Ian Buck, Adam Finkelstein, Charles Jacobs, Allison Klein, David H Salesin, Joshua Seims, Richard Szeliski, and Kentaro Toyama. Performance-driven hand-drawn animation. In *ACM International Symposium on Non-photorealistic Animation and Rendering*, pages 101–108, 2000.

[12] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks). In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[13] Peter Burkert, Felix Trier, Muhammad Zeshan Afzal, Andreas Dengel, and Marcus Liwicki. Dexpression: Deep convolutional neural network for expression recognition. *arXiv:1509.05371*, 2015.

[14] Chen Cao, Menglei Chai, Oliver Woodford, and Linjie Luo. Stabilized real-time face tracking via a learned dynamic rigidity prior. *ACM Transactions on Graphics*, 37(6), December 2018.

[15] Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics*, 33(4), Jul 2014.

[16] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 3D shape regression for real-time facial animation. *ACM Transactions on Graphics*, 32(4), July 2013.

[17] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, March 2014.

[18] Chen Cao, Hongzhi Wu, Yanlin Weng, Tianjia Shao, and Kun Zhou. Real-time facial animation with image-based dynamic avatars. *ACM Transactions on Graphics*, 35(4), July 2016.

[19] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] Miriam Cha, Youngjune Gwon, and H. T. Kung. Multimodal sparse representation learning and applications. *arXiv:1511.06238*, 2015.

[21] Bindita Chaudhuri, Noranart Vesdapunt, Linda Shapiro, and Baoyuan Wang. Personalized face modeling for improved face reconstruction and motion retargeting. In *European Conference on Computer Vision (ECCV)*, 2020.

[22] Bindita Chaudhuri, Noranart Vesdapunt, and Baoyuan Wang. Joint face detection and facial motion retargeting for multiple faces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[23] Lele Chen, Zhiheng Li, Ross K. Maddox, Zhiyao Duan, and Chenliang Xu. Lip movements generation at a glance. In *European Conference on Computer Vision (ECCV)*, 2018.

[24] Nikolai Chinaev, Alexander Chigorin, and Ivan Laptev. Mobileface: 3D face reconstruction with efficient cnn regression. In *European Conference on Computer Vision Workshops (ECCVW)*, 2018.

[25] François Chollet et al. Keras. https://keras.io, 2015.

[26] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[27] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018.

[28] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.

[29] Tim F. Cootes, Mircea C. Ionita, Claudia Lindner, and Patrick Sauer. Robust and accurate shape model fitting using random forest regression voting. In *European Conference on Computer Vision (ECCV)*, 2012.

[30] Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael Black. Capture, learning, and synthesis of 3D speaking styles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[31] Jiankang Deng, George Trigeorgis, Yuxiang Zhou, and Stefanos Zafeiriou. Joint multi-view face alignment in the wild. *arXiv:1708.06023*, 2017.

[32] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2019.

[33] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2011.

[34] Hui Ding, Shaohua Kevin Zhou, and Rama Chellappa. Facenet2expnet: Regularizing a deep face recognition net for expression recognition. *arXiv:1609.06591*, 2016.

[35] Dynamixyz. Performer suite. http://www.dynamixyz.com/.

[36] Pif Edwards, Chris Landreth, Eugene Fiume, and Karan Singh. Jali: An animator-centric viseme model for expressive lip synchronization. *ACM Transactions on Graphics*, 35(4), July 2016.

[37] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

[38] Irfan Essa, Sumit Basu, Trevor Darrell, and Alex Pentland. Modeling, tracking and interactive animation of faces and heads using input from video. In *Computer Animation Proceedings*, pages 68–79. IEEE, 1996.

[39] Faceshift. Faceshift. http://faceshift.com/studio/2015.2/.

[40] Faceware. Faceware live. http://facewaretech.com/.

[41] Bo Fan, Lijuan Wang, Frank K. Soong, and Lei Xie. Photo-real talking head with deep bidirectional lstm. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4884–4888, 2015.

[42] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3D face reconstruction and dense alignment with position map regression network. In *European Conference on Computer Vision (ECCV)*, 2018.

[43] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

[44] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[45] Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. Portrait neural radiance fields from a single image. *arXiv:2012.05903*, 2020.

[46] Lin Gao, Jie Yang, Yi-Ling Qiao, Yu-Kun Lai, Paul L. Rosin, Weiwei Xu, and Shihong Xia. Automatic unpaired shape deformation transfer. *ACM Transactions on Graphics*, 37(6), December 2018.

[47] Pablo Garrido, Levi Valgaerts, Chenglei Wu, and Christian Theobalt. Reconstructing detailed dynamic face geometry from monocular video. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 32(6), Nov 2013.

[48] Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. Reconstruction of personalized 3D face rigs from monocular video. *ACM Transactions on Graphics*, 35(3), May 2016.

[49] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. GANFIT: generative adversarial network fitting for high fidelity 3d face reconstruction. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[50] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman. Unsupervised training for 3d morphable model regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[51] Ross Girshick. Fast R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[52] Paulo Gotardo, Jérémy Riviere, Derek Bradley, Abhijeet Ghosh, and Thabo Beeler. Practical dynamic facial appearance modeling and acquisition. *ACM Transactions on Graphics*, 37(6), December 2018.

[53] Jianzhu Guo, Xiangyu Zhu, Chenxu Zhao, Dong Cao, Zhen Lei, and Stan Z Li. Learning meta face recognition in unseen domains. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[54] Yudong Guo, Juyong Zhang, Jianfei Cai, Boyi Jiang, and Jianmin Zheng. CNN-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[55] Yudong Guo, Juyong Zhang, Lin Cai, Jianfei Cai, and Jianmin Zheng. Self-supervised CNN for unconstrained 3D facial performance capture from a single RGB-D camera. *arXiv:1808.05323*, 2018.

[56] Awni Y. Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. Deep speech: Scaling up end-to-end speech recognition. *arXiv:1412.5567*, 2014.

[57] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R. Hershey, Tim K. Marks, and Kazuhiko Sumi. Attention-based multimodal fusion for video description. In *International Conference on Computer Vision (ICCV)*, pages 4203–4212, 2017.

[58] Pei-Lun Hsieh, Chongyang Ma, Jihun Yu, and Hao Li. Unconstrained realtime facial performance capture. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[59] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[60] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[61] Loc Huynh, Weikai Chen, Shunsuke Saito, Jun Xing, Koki Nagano, Andrew Jones, Paul Debevec, and Hao Li. Mesoscopic Facial Geometry Inference Using Deep Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[62] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. *arXiv:1602.07360*, 2016.

[63] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3D avatar creation from hand-held video input. *ACM Transactions on Graphics*, 34(4), July 2015.

[64] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167*, 2015.

[65] Aaron S. Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3D face reconstruction from a single image via direct volumetric cnn regression. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[66] Vidit Jain and Erik Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[67] Seyed Ali Jalalifar, Hosein Hasani, and Hamid Aghajan. Speech-driven facial reenactment using conditional generative adversarial networks. *arXiv:1803.07461*, 2018.

[68] Boyi Jiang, Juyong Zhang, Bailin Deng, Yudong Guo, and Ligang Liu. Deep Face Feature for Face Alignment and Reconstruction. *arXiv:1708.02721*, 2017.

[69] Luo Jiang, Juyong Zhang, Bailin Deng, Hao Li, and Ligang Liu. 3D Face Reconstruction With Geometry Details From a Single Image. *IEEE Transactions on Image Processing*, 27(10):4756–4770, Oct 2018.

[70] Amin Jourabloo and Xiaoming Liu. Large-pose face alignment via cnn-based dense 3d model fitting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[71] Heechul Jung, Sihaeng Lee, Junho Yim, Sunjeong Park, and Junmo Kim. Joint fine-tuning in deep neural networks for facial expression recognition. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[72] Samira Ebrahimi Kahou, Xavier Bouthillier, Pascal Lamblin, Caglar Gulcehre, Vincent Michalski, Kishore Konda, Sébastien Jean, Pierre Froumenty, Yann Dauphin, Nicolas Boulanger-Lewandowski, et al. Emonets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, 10(2):99–111, 2016.

[73] Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics*, 36(4), July 2017.

[74] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[75] Edward Kim, Darryl Hannan, and Garrett T. Kenyon. Deep Sparse Coding for Invariant Multimodal Halle Berry Neurons. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[76] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics*, 37(4):163:1–163:14, July 2018.

[77] Hyeongwoo Kim, Michael Zollöfer, Ayush Tewari, Justus Thies, Christian Richardt, and Christian Theobalt. Inversefacenet: Deep single-shot inverse face rendering from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[78] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. MultiPoseNet: Fast multi-person pose estimation using pose residual network. In *European Conference on Computer Vision (ECCV)*, 2018.

[79] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 25, pages 1097–1105, 2012.

[80] Rithesh Kumar, Jose Sotelo, Kundan Kumar, Alexandre de Brébisson, and Yoshua Bengio. Obamanet: Photo-realistic lip-sync from text. *arXiv:1801.01442*, 2018.

[81] Samuli Laine, Tero Karras, Timo Aila, Antti Herva, Shunsuke Saito, Ronald Yu, Hao Li, and Jaakko Lehtinen. Production-level facial performance capture using deep convolutional

neural networks. In *Proceedings of ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2017.

[82] Hao Li, Thibaut Weise, and Mark Pauly. Example-based facial rigging. *ACM Transactions on Graphics (Proceedings SIGGRAPH)*, 29(3), Jul 2010.

[83] Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics*, 32(4), July 2013.

[84] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics*, 36(6), November 2017.

[85] Jinpeng Lin, Hao Yang, Dong Chen, Ming Zeng, Fang Wen, and Lu Yuan. Face parsing with roi tanh-warping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[86] Feng Liu, Ronghang Zhu, Dan Zeng, Qijun Zhao, and Xiaoming Liu. Disentangling features in 3d face shapes for joint face reconstruction and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[87] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[88] Ping Liu, Shizhong Han, Zibo Meng, and Yan Tong. Facial expression recognition via a boosted deep belief network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[89] Yilong Liu, Feng Xu, Jinxiang Chai, Xin Tong, Lijuan Wang, and Qiang Huo. Video-audio driven real-time facial animation. *ACM Transactions on Graphics*, 34(6), October 2015.

[90] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[91] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Transactions on Graphics*, 37(4), July 2018.

[92] Patrick Lucey, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2010.

[93] D Lundqvist, A Flykt, and A Öhman. The karolinska directed emotional faces-kdef. cd-rom from department of clinical neuroscience, psychology section, karolinska institutet, stockholm, sweden. Technical report, ISBN 91-630-7164-9, 1998.

[94] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[95] Iain Matthews, Natasha Kholgade, and Yaser Sheikh. Content retargeting using facial layers, January 26 2016. US Patent 9,245,176.

[96] Iain Matthews, Jing Xiao, and Simon Baker. 2D vs. 3D deformable face models: Representational power, construction, and real-time fitting. *International Journal of Computer Vision*, 75(1):93–113, 2007.

[97] S. Mohammad Mavadati, Mohammad H. Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F. Cohn. DISFA: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, 4(2):151–160, 2013.

[98] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

[99] Gaurav Mittal and Baoyuan Wang. Animating face using disentangled audio representations. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[100] Mixamo. Face plus. https://www.mixamo.com/faceplus/.

[101] Ali Mollahosseini, David Chan, and Mohammad H Mahoor. Going deeper in facial expression recognition using deep neural networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[102] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. paGAN: Real-time avatars using dynamic textures. *ACM Transactions on Graphics*, 37(6), December 2018.

[103] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *International Conference on Machine Learning (ICML)*, 2011.

[104] Kyle Olszewski, Zimo Li, Chao Yang, Yi Zhou, Ronald Yu, Zeng Huang, Sitao Xiang, Shunsuke Saito, Pushmeet Kohli, and Hao Li. Realistic dynamic facial textures from a single image using gans. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[105] OptiTrack. Expression. http://optitrack.com/products/expression/.

[106] Maja Pantic, Michel Valstar, Ron Rademaker, and Ludo Maat. Web-based database for facial expression analysis. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2005.

[107] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[108] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2009.

[109] Frank E. Pollick. In search of the uncanny valley. In *User Centric Media*, pages 69–78, Berlin, Heidelberg, 2010. Springer.

[110] Yunxiao Qin, Chenxu Zhao, Xiangyu Zhu, Zezheng Wang, Z. Yu, Tianyu Fu, Feng Zhou, J. Shi, and Z. Lei. Learning meta model for zero- and few-shot face anti-spoofing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[111] Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[112] Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv:1603.01249*, 2016.

[113] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[114] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv:1804.02767*, 2018.

[115] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[116] Roger Blanco i Ribera, Eduard Zell, J. P. Lewis, Junyong Noh, and Mario Botsch. Facial retargeting with automatic range of motion alignment. *ACM Transactions on Graphics*, 36(4), 2017.

[117] Elad Richardson, Matan Sela, and Ron Kimmel. 3D face reconstruction by learning from synthetic data. In *International Conference on 3D Vision (3DV)*, 2016.

[118] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. Learning detailed face reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[119] Sami Romdhani and Thomas Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[120] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015.

[121] Joseph Roth, Yiying Tong, and Xiaoming Liu. Adaptive 3D face reconstruction from unconstrained photo collections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[122] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2013.

[123] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael Black. Learning to regress 3d face shape and expression from an image without 3d supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[124] Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. Real-time avatar animation from a single image. In *Face and Gesture*, 2011.

[125] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[126] Sumit Shekhar, Vishal M. Patel, Nasser M. Nasrabadi, and Rama Chellappa. Joint sparse representation for robust multimodal biometrics recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):113–126, 2014.

[127] Jie Shen, Stefanos Zafeiriou, Grigorios G. Chrysos, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2015.

[128] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[129] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[130] Yang Song, Jingwen Zhu, Dawei Li, Andy Wang, and Hairong Qi. Talking face generation by conditional recurrent adversarial network. In *International Joint Conference on Artificial Intelligence, (IJCAI)*, 2019.

[131] Robert W. Sumner and Jovan Popović. Deformation transfer for triangle meshes. In *ACM SIGGRAPH*, 2004.

[132] Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: Learning lip sync from audio. *ACM Transactions on Graphics*, 36(4), July 2017.

[133] Xiaoyang Tan and Bill Triggs. Fusing gabor and lbp feature sets for kernel-based face recognition. In *International Workshop on Analysis and Modeling of Faces and Gestures*, pages 235–249. Springer, 2007.

[134] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems (NIPS)*, 2020.

[135] Paul Tassi. "Mass Effect: Andromeda" Review (PS4): Every Man's Sky, March 2017.

[136] Sarah Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. A deep learning approach for generalized speech animation. *ACM Transaction on Graphics*, 36(4), July 2017.

[137] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[138] Ayush Tewari, Florian Bernard, Pablo Garrido, Gaurav Bharaj, Mohamed Elgharib, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. FML: face model learning from videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[139] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason M. Saragih, Matthias Nießner, Rohit Pandey, Sean Ryan Fanello, Gordon Wetzstein, Jun-Yan Zhu, Christian Theobalt, Maneesh Agrawala, Eli Shechtman, Dan B. Goldman, and Michael Zollhöfer. State of the Art on Neural Rendering. *Computer Graphics Forum (EG STAR 2020)*, 2020.

[140] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[141] Ayush Tewari, Michael Zollöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Theobalt Christian. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[142] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[143] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gerard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[144] Luan Tran, Feng Liu, and Xiaoming Liu. Towards high-fidelity nonlinear 3D face morphable model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[145] Luan Tran and Xiaoming Liu. Nonlinear 3D face morphable model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[146] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. In *ACM SIGGRAPH*, 2005.

[147] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. End-to-End Speech-Driven Facial Animation with Temporal GANs. *arXiv:1805.09313*, 2018.

[148] Guangting Wang, Chong Luo, Xiaoyan Sun, Zhiwei Xiong, and Wenjun Zeng. Tracking by instance detection: A meta-learning approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[149] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[150] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. In *ACM Transactions on Graphics (TOG)*, volume 30, page 77, 2011.

[151] Chenglei Wu, Takaaki Shiratori, and Yaser Sheikh. Deep incremental learning for efficient high-fidelity face tracking. *ACM Transactions on Graphics*, 37(6), December 2018.

[152] Mike Wu and Noah Goodman. Multimodal generative models for scalable weakly-supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[153] Shengtao Xiao, Shuicheng Yan, and Ashraf A. Kassim. Facial landmark detection via progressive initialization. In *IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015.

[154] Xuehan Xiong and Fernando De la Torre. Global supervised descent method. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[155] Xuehan Xiong and Fernando Torre. Supervised descent method and its applications to face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[156] Jing Yang, Jiankang Deng, Kaihua Zhang, and Qingshan Liu. Facial shape tracking via spatio-temporal cascade shape regression. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2015.

[157] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. From facial parts responses to face detection: A deep learning approach. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[158] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[159] Ugur Ulvi Yetiskin. Tuna rig. https://www.behance.net/gallery/31141085/Tuna-Rig-for-FREE.

[160] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. A 3D facial expression database for facial behavior research. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR)*, pages 211–216, 2006.

[161] Zi-Lu Ying, Zhe-Wei Wang, and Ming-Wei Huang. Facial expression recognition based on fusion of sparse representation. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, pages 457–464. 2010.

[162] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems (NIPS)*, 2014.

[163] Ronald Yu, Shunsuke Saito, Haoxiang Li, Duygu Ceylan, and Hao Li. Learning dense facial correspondences in unconstrained images. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[164] Xiang Yu, Jianchao Yang, Linjie Luo, Wilmot Li, Jonathan Brandt, and Dimitris Metaxas. Customized expression recognition for performance-driven cutout character animation. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[165] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[166] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor S. Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *International Conference on Computer Vision (ICCV)*, 2019.

[167] Thiago H. H. Zavaschi, Alceu S. Britto, Luiz E. S. Oliveira, and Alessandro L. Koerich. Fusion of feature sets and classifiers for facial expression recognition. *Expert Systems with Applications*, 40(2):646–655, 2013.

[168] Robert Zemeckis. The polar express. 2005.

[169] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016.

[170] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[171] Shun Zhang, Yihong Gong, Jia-Bin Huang, Jongwoo Lim, Jinjun Wang, Narendra Ahuja, and Ming-Hsuan Yang. Tracking persons-of-interest via adaptive discriminative features. In *European Conference on Computer Vision (ECCV)*, pages 415–433, 2016.

[172] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision (ECCV)*, 2014.

[173] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. Talking face generation by adversarially disentangled audio-visual representation. In *AAAI Conference on Artificial Intelligence*, 2019.

[174] Yang Zhou, Zhan Xu, Chris Landreth, Evangelos Kalogerakis, Subhransu Maji, and Karan Singh. Visemenet: Audio-driven animator-centric speech animation. *ACM Transactions on Graphics*, 37(4), July 2018.

[175] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[176] Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Face alignment by coarse-to-fine shape searching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[177] Wenbin Zhu, HsiangTao Wu, Zeyu Chen, Noranart Vesdapunt, and Baoyuan Wang. Reda:reinforced differentiable attribute for 3d face reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[178] Xiangyu Zhu, Zhen Lei, Stan Z Li, et al. Face alignment in full pose range: A 3d total solution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[179] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3D solution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[180] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3D face reconstruction, tracking, and applications. *Computer Graphics Forum*, 37:523–550, 2018.