LEARNING HUMAN FACIAL PERFORMANCE: ANALYSIS AND SYNTHESIS

by

HAI XUAN PHAM

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

VLADIMIR PAVLOVIC

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

January, 2019

ABSTRACT OF THE DISSERTATION

Learning Human Facial Performance: Analysis and Synthesis

by HAI XUAN PHAM

Dissertation Director:

VLADIMIR PAVLOVIC

Human faces convey a large range of semantic meaning through facial expressions, which reflect both actions, e.g. talking, and affective states such as happy, surprised, etc. More importantly, in the coming age of artificial intelligence and virtual persona, facial expressions can serve as a two-way communicative interface between human and machine. Thus, understanding human facial expressions has been a premier research area in the computer vision and machine learning community for decades, achieving significant advances in face tracking, reconstruction and synthesis. In this dissertation, we study two aspects of face modeling: the analysis and reconstruction of human facial expressions via interpretable 3D blendshape representation from different input modalities, and the reversed problem in which we train a model to hallucinate coherent facial expressions directly given any arbitrary portrait and facial action parameters.

First, we propose a real-time robust 3D face tracking framework from RGBD videos capable

of tracking head pose, facial actions and on-the-fly identity adaptation without precalibration or intervention from a user. In particular, we emphasize on improving the tracking performance in instances where the quality of input data is drastically reduced, e.g. when the face is at long distance or at large pose. This is accomplished by the combination of a flexible and extremely efficient 3D shape regressor and the joint 2D+3D optimization on shape parameters.

Second, we present recurrent neural network frameworks which automatically estimate facial action unit (AU) intensities of a speaker from just her speech, for real-time facial animation. Specifically, the time-varying contextual non-linear mapping between audio stream and visual facial transformations is realized by a recurrent neural network. Our models not only activate appropriate facial action units at inference to depict different utterance generating actions, in the form of lip movements, but also, without any assumption, automatically estimate emotional intensity of the speaker and reproduce her ever-changing affective states by adjusting strength of facial unit activations. We introduce a baseline model, which uses engineered acoustic features, and an end-to-end model that learn feature representation directly from audio spectrograms.

Finally, we propose a novel deep generative neural network that enables fully automatic, real-time facial expression synthesis of an arbitrary portrait with continuous action unit coefficients. In particular, our model directly manipulates image pixels to make the unseen subject in the still photo express various emotions controlled by values of facial AU coefficients, while maintaining her personal characteristics, such as facial geometry, skin color and hair style, as well as the original surrounding background. In contrast to prior work in facial expression editing, our proposed model is purely data-driven and it requires neither a statistical face model nor image processing tricks to enact facial deformations. Additionally, our model is trained from unpaired data, where the input image and the target frame are from different persons. Our work gives rise to template-and-target-free expression editing, where still faces can be effortlessly animated with arbitrary AU coefficients provided by the user, or driven by our aforementioned tracker and speech models.

Acknowledgements

I would like to express my deepest thanks to all friends and colleagues who have helped me complete my degree. The following acknowledgements are by no means exhaustive, for which I apologize.

I want to express my gratitude to my advisor, Professor Vladimir Pavlovic. His support and guidance made it possible for me to complete my doctoral study.

I also thank my thesis committee members, Professor Dimitris Metaxas, Professor Ahmed Elgammal and Professor Jiebo Luo for dedicating their time to better my work.

I would like to thank Professor William Steiger and Professor Matthew Stone for their generous supports throughout my graduate years.

I want to thank my fellow labmates, Saehoon, Jongpil, Sejong, Behnam, Cuong, Gang, Fangda, Yuting and Mihee for the insightful discussions and for making our group a joyful place. I also want to thank all of my friends and fellow graduate students for their companionships, which are unforgettable memories of my life.

I thank Professor Jian-fei Cai, Professor Tat-jen Cham, Chongyu Chen, Deng Teng and other students at Nanyang Technological University for their invaluable cooperation in the 3D face tracking and depth recovery projects.

I am grateful for my undergraduate thesis advisor, Ba Manh Luong. Without him, I would not have engaged in the adventure of computer vision and machine learning research to where I am today.

The generous fellowship of the Vietnam Education Foundation program kickstarted my Ph.D.

journey in the US. Without the program, I would not have considered to pursue graduate study at Rutgers. Although the program has ended this year, I would like to thank the VEF staff and hope that they can help many more students to pursue their graduate studies in the US.

But most of all, I want to thank my family - my parents, my sister and her family, my wife and our little daughter for their unconditional love, support and encouragement.

This dissertation contains materials from five following papers:

- [84] H. X. Pham, V. Pavlovic, J. Cai and T. Cham. Robust Real-time Performance-driven 3D Face Tracking. In ICPR, 2016.
- [83] H. X. Pham and V. Pavlovic. Robust Real-Time 3D Face Tracking from RGBD Videos under Extreme Pose, Depth, and Expression Variations. In 3DV, 2016.
- [81] H. X. Pham, S. Cheung and V. Pavlovic. Speech-driven 3D Facial Animation with Implicit Emotional Awareness: A Deep Learning Approach. In CVPRW, 2017.
- [85] H. X. Pham, Y. Wang and V. Pavlovic. End-to-end Learning for 3D Facial Animation from Speech. In ICMI, 2018.
- [87] H. X. Pham, Y. Wang and V. Pavlovic. Generative Adversarial Talking Head: Bringing Portraits to Life with a Weakly Supervised Neural Network. In arXiv, 2018.

Dedication

To my parents, my sister and her family: without their endless support and encouragement, I would not have started my study.

To my wife and our lovely daughter, who have taught me the happiness and joy of having a family and being a parent.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Facial expression[1] has long been a natural means of communicative interface between humans. Particularly, it was shown in the seminal work of Ekman et al. [39, 40, 41] that human facial expressions are universal across different cultures and languages. Nowadays when some aspects of life are intertwined with the virtual world, it is increasingly important for the computer to understand not only what the human wants but also how the human "feels", via their facial expressions, so that the machine (also called intelligent agent, or A.I. in this thesis) is able to produce a meaningful, pleasant feedback.

Conversely, it is also important that the intelligent agent is able to communicate its "feelings" back to the human as a part of its reaction, in order to make the mutual interaction between human and machine more natural and realistic. In other words, the agent can explicitly exhibit its (artificial) emotion to the user via facial expressions as a form of communication, blurring the line between human-machine and making the communicative interface be as transparent as between real humans. Let us look at a futuristic example of human-machine interaction in Fig. 1.1.

In this example, the physical interface between human and machine consists of visual and acoustic channels, and possibly other actions but here we focus on only voice and facial expressions carried in those information streams. The intelligent agent listens to the user's voice and

[1]The terms facial "expression", "action" and "performance" are used interchangeably in this dissertation, with the exception in Chapter 2, where facial performance also includes global head pose.

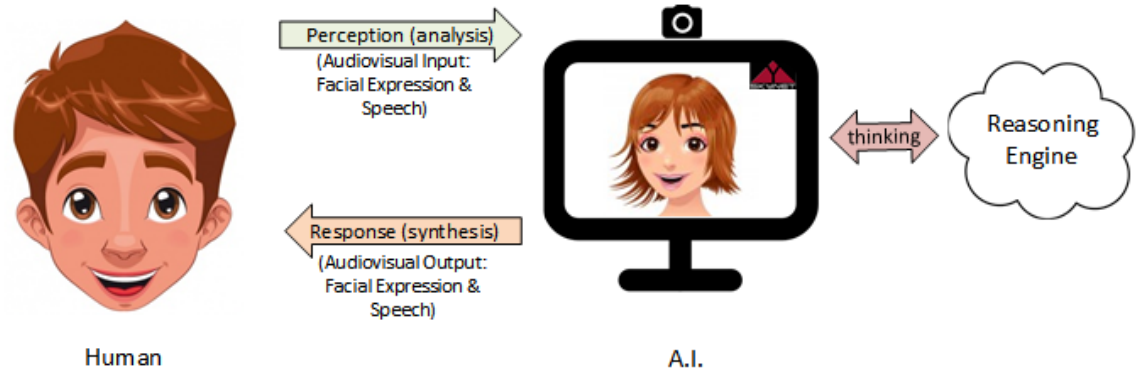


Figure 1.1: An imaginary conversation between human and artificial intelligent agent. The user says, and shows that, he is happy. The agent perceives his textural facial expression and speech (via a webcam) and after proper reasoning, it responds with adequate (artificial) facial expression and speech. The conversation becomes natural, similar to how humans interact with each other. *Source of images: Internet*.

captures his facial expressions with a webcam, which comprises her senses. This process is machine perception, in which she processes the semantic content of the user's speech, and recognizes his emotions from video and audio. Both speech content and emotion of the user are then propagated to a reasoning engine (equivalence of the neocortex of human brain) for further processing, or "thinking", to create a plausible, human-like answer to the human user. This response is manifested as synthesized speech, as well as artificial facial expressions. For instance, when the user says "Kids are really noisy today" with a happy face (and voice), the agent understands that he is amused by the kids and does not mind them being noisy at all. The agent just merrily replies him with a joke or something interesting to continue the conversation. But if he says that while being angry, the agent interprets that the user is being annoyed, it will ask if the user wants close the doors and windows. In this scenario, facial expressions are very important to explain the semantics of the user's true intent. Without knowing the context of the speech present in his emotion, the agent may just interpret that the user is irritated. Hence, being able to analyze both the user's words and emotions helps the intelligent agent understand correctly what the user really meant. In another

Figure 1.2: Virtual avatar animation. *Courtesy: CrazyTalk™ by Reallusion.*

example, the user thanks the agent for preparing his drink. The intelligent agent says that it is glad, with a synthesized smile, showing that it is happy to be of help to the user, and thus it creates a pleasant atmosphere for the user. In both scenarios, the machine can analyze the user input and emotion, process them properly to create a correct and "natural" response with adequate synthetic emotion that suits the situation, it is as if the user is really communicating with another human. Although it is beyond the scope of this thesis, we believe that in near future the human-machine interaction will be indistinguishable from real ones between humans.

Furthermore, there are various applications of facial expression/action modeling, such as computer games, animated movies, teleconferencing, etc. Frameworks such as Faceshift™ for 3D face tracking are integrated with graphics engines to enable in-game user-driven facial animation. CrazyTalk™ (Fig. 1.2) provides a framework that lets user animate a 3D virtual avatar with their speech. They can also be used to re-create and animate 3D face shape of a person for teleconferencing. In these applications, traditional facial capture approaches have gained tremendous successes, reconstructing high level of realism. Yet, active face capture rigs utilizing motion sensors/markers are expensive and time-consuming to use. Alternatively, passive techniques capturing facial transformations from cameras, although less accurate, have achieved very impressive performance and allowed easy deployment and can be used anywhere. We will show later that it is possible to predict facial expressions/actions from video or audio efficiently with reasonable to high accuracy. Facial capture is also heavily used in movies, where facial expressions are transferred from one actor to another target. This process involves many steps, including 3D face alignment, deformation transfer and texture mapping. We will show that using our learnt facial expression synthesis model,

this process is greatly simplified, in which the user can specify which facial actions[2] are desired through a set of interpretable parameters.

1.1 Research Statement

In this thesis, we study how to model facial expression in human-machine interaction, as a two-way communicative interface. Particularly, in machine perception, we let the agent learn to analyze human facial expression from audio and visual channels. Moreover, we present an approach that allows the agent to learn to generate any arbitrary facial expression, in order to convey its own emotions to the user.

In summary, we study three problems of facial expression modeling:

- **Learning** facial expression for real-time 3D facial capture using a commodity RGB-D camera: Combining learnt 3D face alignment and reconstruction for robust face tracking. An efficient 3D face regression is proposed, which is trained from standard image datasets.
- **Learning** facial expression from speech in real-time: Training deep neural networks to infer facial actions from just speech audio. Our models predict raw AU intensities, which can be used to determine a specific type of emotion, or other tasks such as expressive talking face animation.
- **Learning** to generate photo-real facial expression synthesis. Given any arbitrary portrait and action unit intensities, our learnt generative neural network can hallucinate expressive facial appearance without using any face template.

The first two topics make up the *analysis* part of this thesis, which enables the computer agent to capture facial expressions from video and audio via learning, as well as synthesize 3D facial

[2]Facial action units and their parameterization are discussed in details in section 1.3.

shape of the user. The third research topic is *synthesis* that enables the intelligent agent to learn to generate its own realistic facial expressions. In the next section, we discuss motivations that lead to the proposal of our models to solve three aforementioned tasks. It is followed by a brief introduction of the face and facial expression model which is used throughout this thesis.

1.2 Motivations

1.2.1 Learning Facial Performance for Visual Tracking from RGBD Videos

3D facial capture has various applications, including not only facial expression/action analysis - the main topic of this thesis, but also video games, animation, movies among others. Despite capturing highly accurate facial geometric changes to minute details, active capture rig is very complex and time-consuming to deploy. Passive (markerless) techniques, such as the solution provided by Faceware^{TM}Technologies Inc., utilize head-mounted cameras to track every facial movement. These techniques, although a little less accurate, have achieved very impressive performance. However, most commercial systems often require hi-resolution cameras, and do not work in real-time without special hardware. Recently, there are real-time 3D face tracking systems that perform very well using commodity capture devices with standard computer equipments. Faceshift^{TM}[118] and its derivatives [13, 66] use low-cost RGBD camera to track and reconstruct 3D face, while [19, 18, 17] use standard webcam for this task. All methods above reconstruct 3D face from personalized expression blendshapes.

Nonetheless, difficult problems remain due to variations in camera pose, video quality, head movement and illumination, added to the challenge of tracking different people with many unique facial expressions. Most systems assume that the input stream is clean and clear, i.e. the face is at frontal pose, well lit and has high enough resolution. RGBD-based tracking systems such as Faceshift^{TM}rely heavily on the quality of both input color and depth data. Tracking and reconstruction are formulated upon optical flow and Iterative Closest Point (ICP) optimizations. When the

face is at a relatively large distance from the camera, quality of depth data deteriorates exponentially, and the face texture resolution is low. These issues, which is common in consumer-grade cameras, greatly impact performance of RGBD-based tracking systems. These methods only work when the face is at close range, and the depth map retains fine geometric surface structure. One way to overcome sensor limitations is to 1) not rely on depth and 2) train a robust facial transformation predictor that can perform well in different imagery conditions (as was shown in [18]). But there exists other limitations with this approach. First, the face transformation predictor performs more poorly when the image resolution is reduced. Second, 3D face reconstruction from only color channels is an ill-posed problem, it lacks necessary information to resolve the depth ambiguity, e.g. a smaller face at close distance will appear similar to a larger face at a further distance. Instead, we expect to achieve more robust tracking results if we were able to incorporate depth data while intelligently handling its inaccuracies at greater distances.

This motivates us to propose a robust RGBD face tracker combining the advantages of 3D shape regression from standard RGB channels and 3D point cloud registration, which has the following advantages:

- Our tracker works in real-time, does not require any calibration or user intervention. User identity is adapted on-the-fly. It is driven by an *extremely* efficient 3D shape regressor, which we believe to be the fastest 3D face alignment technique running on CPU.

- Our tracker is robust to extreme conditions, where the face is at large distance, the data (color or depth) is noisy or has low quality, and the face is at large pose by intelligently exploiting information from both color and depth streams.

- Our tracker is performance-driven, i.e. it recovers facial parameters directly, which can be used to reconstruct a full 3D face model effortlessly, or they can be used to animate another blendshape model sharing the same configuration.

More details on 3D face shape regression and tracking can be found in Chapter 2.

1.2.2 Learning Facial Performance from Speech

In the previous section we mentioned applications of 3D face tracking and what motivates us to propose our face tracker, which can handle low-quality, unreliable input visual stream. Still, there raises a question: can we predict continuous facial expressions without visual input, and how? We addressed this issue by predicting facial expressions exclusively from speech audio.

Speech, as a natural form of communication among various modes of interactions, is increasingly used in human-machine interaction, evidenced by the ever-growing popularity of virtual voice assistants, such as Cortana^{TM}, Alexa^{TM}, etc. embedded in cellphones, computers or IoT devices. Speech recording carries not only the contextual information, but also emotional states of the speaker. Both source of information is necessary in order to interpret the true intention of the speaker, as in the example shown at the beginning of this chapter. Hence it is desirable that the intelligent is able to predict the affective state of the user from her speech. As we will show later in this work, it is possible to infer facial expressions from just speech without visual cues, so that it allows the intelligent agent to correctly understand how the user feels from what he say, and the agent can take the correct course of actions.

There are other applications of speech-driven facial expression estimation. For example, the Crazytalk^{TM}software lets the user to animate his own avatar with speech (example in Fig. 1.2. Such work can be used to replace Faceshift^{TM}for in-game animation. Speech-driven animation methods are even more convenient to deploy and use, because they do not require any camera device and the user does not have to stay close to the camera all the time. Moreover, in many situations, only audio recording of a person is available, which can be exploited to create her facial expressions.

In this work, we propose different recurrent neural network models to realize the highly non-linear mapping between speech and facial actions. These models can be used to create a talking 3D

virtual avatar that can naturally make micro facial movements to reflect the time-varying contextual information and emotional intensity carried in the input speech. Intuitively, this work is analogous to visual 3D face tracking, however, it is more challenging as we try to map acoustic signal to visual space, instead of conveniently relying on textural cues from input video. Our work has the following advantages:

- Our models are efficient, enabling real-time speech-driven 3D facial animation.
- Our models can implicitly infer various facial expressions represented by action unit parameters.
- Our end-to-end models are able to learn meaningful feature representation which not only improves performance but also reduces running time.

More details on this topic and various examples are presented in Chapter 3.

1.2.3 Learning Facial Performance Synthesis

Facial expression editing is the research of transferring the semantic expression from a target to a source face, which has achieved impressive results e.g. Face2Face [108]. Face retargeting has been used extensively in movies and animation. Putting manual editing aside, there is a common approach for automatic facial expression transfer: 1) aligning face shape models (2D or 3D templates) to both source and target; 2) transferring the transformation of the target face to the source; and 3) applying texture mapping on the source and applying other image processing tricks to make the transformed source face look realistic. However, this common approach is time-consuming to use, and it still does not guarantee that the transformed source face will look natural.

In recent years, with the widespread applicability of deep learning in general, there are efforts to let the computer learn how to perform expression transfer with a deep neural network, such as the works by Olszewski [78] or Yeh et al. [126]. This approach has a couple of advantages:

First, it performs facial expression transfer in one shot without many small steps involved. Second, it can generate more natural faces after observing many different faces. The learning serves as regularization itself, hence the generated face may not stray too far from the learnt natural face manifold. In contrast, there is hardly anything in the aforementioned model-based approach that enforces quality and realism of the transformed face.

The learning-based methods often assume that a pair of source face-target expressive face is available for transferring. However, there are situations in which the target face to drive facial deformation of the source does not exist, instead, facial expression can be inferred from other input modalities, such as speech (presented in Chapter 3). Facial expression can also be explicitly specified by user as vector of facial action unit (AU) intensities, in other words, the user is a director and vectors of facial action unit intensities are script that describes how the actor (the source face) should act.

In this thesis, we are interested in directly animating a human portrait given only AU coefficients. Particularly, our proposed deep model is able to modify a frontal face portrait of arbitrary identity and expression at pixel level, hallucinating a novel facial image whose expressiveness mimics that of a real face that has similar AU attributes. In other words, our model is formulated similar to a human-like thought process, it learns to extract identity features to preserve individual characteristic of the portrait, enact facial expression to animate the portrait according to values of AU coefficients, perform texture mapping, all in an end-to-end deep neural network.

Learning identity features requires a large number of training images from thousands of subjects, which are readily available in various public image datasets. On the other hand, the amount of publicly available expressive videos such as [71], from which we could collect a wide range of AU coefficients, is rather limited. A deep net trained on such a small number of subjects would not generalize well to unseen identity. To address this shortcoming, we propose to train the deep net with separate source and target sets, i.e. the animated facial image of subject A in the source set

does not have an exact matching target image, but there exists an image of subject B in the target set that has similar expression to the synthesized image of A, and their expressiveness similarity is measured in a facial expressiveness subspace. Our model learns to generate an expressive face of subject A by combining identity features extracted from the image of subject A, and expressiveness features from image of subject B. We will demonstrate later that, by learning to separate identity code from expression features, our model is able to not only stimulate facial expressions, but also suppress them, without any using any face template, Our work enables a whole new level of flexibility in facial editing, and we believe it is the first in introducing expression-mimicry learning from separate source and target subjects.

In summary, our facial performance synthesis models have the following advantages:

- Our model learns to generate expressive face by combining identity features of one subject with expressiveness features of another in an end-to-end learning framework. It does not require tremendous amount of training data in order to learn facial expression transformation effectively.

- Our model edits the source face expression by given AU coefficients, it does not require the presence of the target face.

- Our model allows both facial expression enactment as well as suppression. It accepts any arbitrary source image and the original expression in the image does not affect the generation of final expression.

Our proposed model enables a whole new level of flexibility for facial expression editing. More details can be found in Chapter 4.

1.3 Face Shape and Facial Expression Representation

The studies of Ekman et al. [39, 40, 41] have shown that human facial expressions are universal. This observation led to the design of Facial Action Coding System (FACS) [38], in which a particular expression is a weighted combination of facial action units (AUs), each unit represents micro movements of muscles in a particular face region. For example, if the weight of "Left brow raiser" is high, the eyebrow is raised and vice versa; or "lip raiser" and "jaw dropper" control the talking action. FACS defines common relative geometric changes of different parts on the human face to manifest expression, regardless of individual characteristics. A few parametric 3D face models have been designed based on FACS, e.g. Candide [3] and FaceWarehouse blendshape [20], shown in Fig. 1.3. Although these models do not fully conform to the system, they allow user to explicitly generate any (common) expression by adjusting intensities of different action units appropriately.

In this work we use exclusively the face model developed in the FaceWarehouse database, shown in Fig. 1.3b. As specified in [20], an arbitrary 3D face of a person including expression can be approximated by

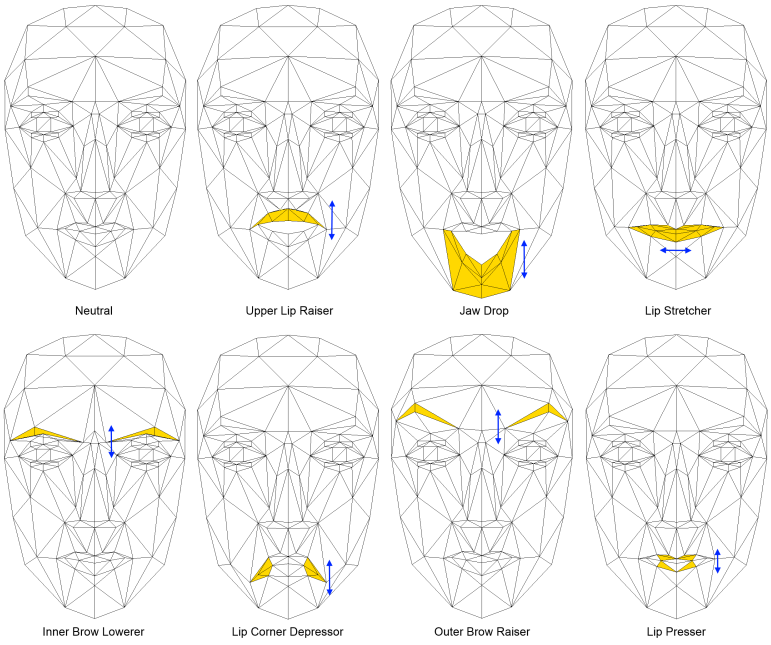
$$V = C_r \times_2 w_{id}^T \times_3 w_{exp}^T \tag{1.1}$$

where C_r is a 3D array (called reduced core tensor) of size (N_v, N_{id}, $N_e + 1$) (corresponding to number of vertices, number of identities and number of expressions, respectively), w_{id} is an N_{id}-dimension identity vector, and w_{exp} is an $N_e + 1$-dimension expression vector. N_e is the number of facial expressions and $N_e + 1$ includes the neutral pose. (1.1) basically describes tensor contraction at the 2nd mode by w_{id} and at the 3rd mode by w_{exp}.

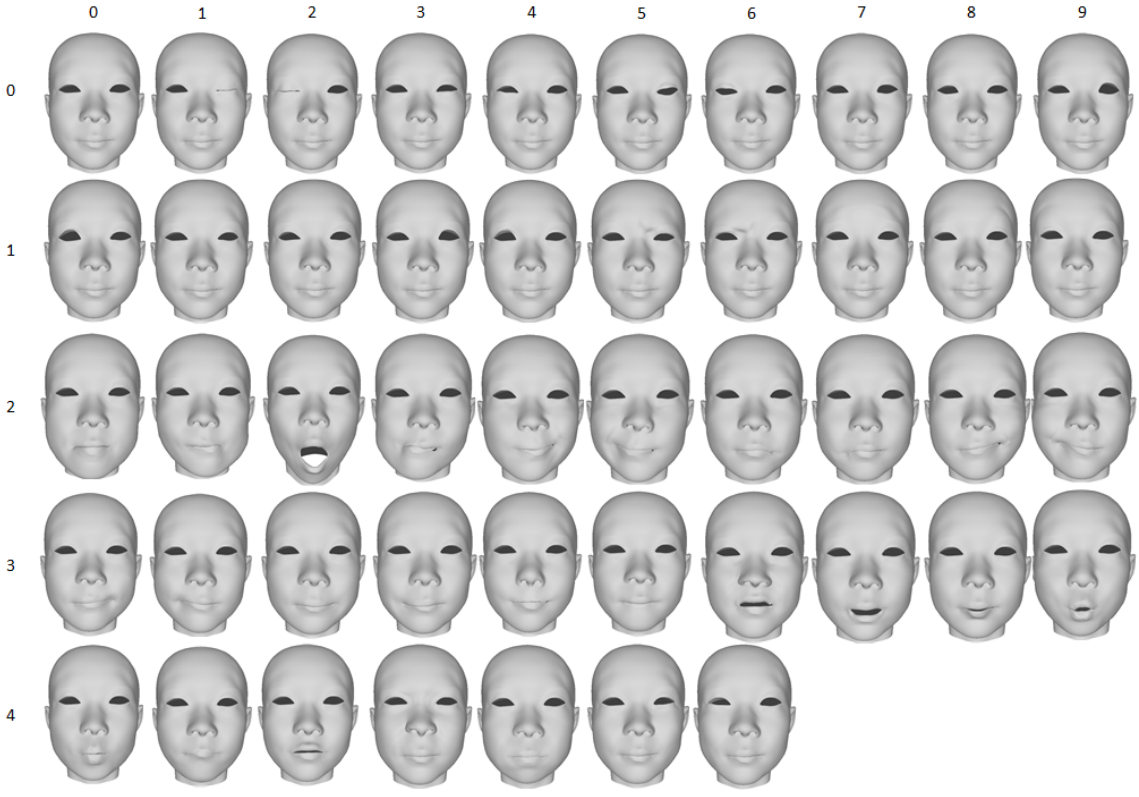
Similar to [19], for real-time face tracking of one person, given his identity vector w_{id}, it is more convenient to reconstruct the $N_e + 1$ expression blendshapes for the person of identity w_{id} as

$$B_i = C_r \times_2 w_{id}^T \times_3 u_{\exp_i}^T \tag{1.2}$$

where $u_{\exp_i}$ is the pre-computed weight vector for the i-th expression mode. B_i is also called the



(a) Candide-3



(b) FaceWarehouse blendshape

Figure 1.3: Some 3D face models based on FACS. In (a) we show some prominent action units of the Candide-3 wireframe model. In (b), there are 46 expression blendshapes and a neutral shape (at (0,0) coordinates).

i-th action unit and B_0 is the neutral shape. In this way, an arbitrary facial shape of the person can be represented as a linear sum of his expression blendshapes:

$$V = B_0 + \sum_{i=1}^{N_e} (B_i - B_0)e_i \tag{1.3}$$

where $e_i \in [0, 1]$ is a blending weight, $i = 1, \ldots, N_e$. Finally, a fully transformed 3D facial shape can be represented as

$$S = R \cdot V(B, e) + T \tag{1.4}$$

with parameters $\theta = (R, T, e)$, where R and T respectively represent global rotation and translation, and $e = \{e_i\}$ defined in (1.3) represent expression deformation parameters (or action unit intensities). In this work, we keep the 50 most significant identity knobs in the reduced core tensor C_r, hence (N_v, N_{id}, N_e) = (11510, 50, 46).

However, not all parameters in (R, T, e, w_{id}) are used throughout three tasks in this thesis. The list below describes the number of parameters modeled in each task.

- In face tracking, all parameters (R, T, e, w_{id}) are predicted. Specifically, the identity-agnostic shape regressor estimates (R, T, e), and all parameters including identity coefficients are refined using the 2D+3D optimization. More details in Chapter 2.

- In speech-driven facial expression analysis, e is inferred from speech, and used to animate a generic-identity facial blendshape in our experiments (i.e. w_{id} is fixed). More details in Chapter 3.

- In facial expression synthesis, given e and a portrait, our proposed model generates a new face image, portraying the given expression while preserving the identity of the source face, i.e. the identity description includes not only facial geometric surface details, but also color texture details such as hair, eyes and eyebrows, beard, ornaments and lastly, the image background. Since in this task we work on the image pixel space instead of the shape space, w_{id}

is replaced with more sophisticated identity code implicitly measured by hidden layers of a deep net to reconstruct the personalized expressive facial image. More details in Chapter 4.

1.4 Dissertation Outline

In this chapter we introduced motivations that lead to the proposal of our models for 3D face tracking, speech-driven facial expression analysis and facial expression synthesis, and their advantages compared to other works in literature. We also described the blendshape face model and the facial action unit system from the FaceWarehouse database that is exclusively used in this thesis.

Chapter 2 presents our real-time 3D face tracking framework. The tracker is driven by an efficient identity-agnostic shape regressor that recovers facial transformation parameters directly from RGB frame. These parameters and the identity code are refined using both raw estimates as well as the available depth data. Furthermore, we provide different modifications to the shape regressor to make it more robust to extreme head poses. Intensive experiments on synthetic and real data demonstrate the accuracy and robustness of our proposed face tracker(s).

Chapter 3 describes a family of recurrent deep neural networks to estimate facial action unit intensities directly from speech. Specifically, we first present a baseline model which utilizes engineered acoustic features as input. We show that, better feature representation can be learned directly from speech spectrograms, which brings significant performance gain in our experiments on various public audiovisual datasets.

Chapter 4 introduces GATH, a novel deep generative neural network that can create any novel facial expression image, given an arbitrary portrait and desired action unit parameters that control the expression. We demonstrate that GATH can directly manipulate image pixels to hallucinate a particular facial expression of the unseen subject, while maintaining her individual characteristics, regardless of the expression displayed in the portrait.

We conclude the thesis in Chapter 5, with potential future work directions.

Chapter 2

Learning Facial Performance for 3D Face Tracking from RGBD Videos

In this chapter, we introduce a novel robust real-time hybrid 3D face tracking framework from RGBD video streams, which is capable of tracking head pose and facial actions simultaneously without pre-calibration or intervention from a user. In particular, we emphasize on improving the tracking performance in instances where the tracked subject is at a large distance from the cameras, the quality of point cloud deteriorates severely, and the face may be occluded or at large pose. This is accomplished by the combination of a flexible 3D shape regressor and the joint 2D+3D optimization on shape parameters.

Our approach fits facial blendshapes to the point cloud of the human head, while being driven by an efficient and rapid 3D shape regressor trained on generic RGB datasets. Head pose-aware versions of the regressor are able to predict facial parameters even at profile pose, where half of the face is not visible. As an on-line tracking system, the identity of the unknown user is adapted on-the-fly resulting in improved 3D model reconstruction and consequently better tracking accuracy. The result is a robust on-line RGBD 3D face tracker that can model extreme head poses and facial expressions accurately in challenging scenes, which are demonstrated in our extensive experiments.

This chapter contains materials from our published papers [84, 83].

2.1 Introduction

Traditional facial capture approaches have gained tremendous successes, reconstructing high level of realism. Yet, active face capture rigs utilizing motion sensors/markers are expensive and time-consuming to use. Alternatively, passive vision-based techniques [7, 8, 45, 112] capturing facial transformations from cameras, although a little less accurate, have achieved very impressive performance. In particular, there are real-time systems that perform very well even when using commodity capture devices [19, 18, 17, 118, 13, 66]. However, difficult problems remain due to variations in camera pose, video quality, head movement and illumination, added to the challenge of tracking different people with many unique facial expressions.

Blendshape-based face models, such as the shape tensor used in the FaceWareHouse [20] which was introduced in Section 1.3, were developed for more sophisticated, accurate 3D face tracking. By deforming dense 3D blendshapes to fit facial appearances, facial motions can be estimated with high fidelity. Such techniques have gained attention recently due to the proliferation of consumer-grade range sensing devices, such as the Microsoft Kinect [76], which provide synchronized color (RGB) images and depth (D) maps in real time. By integrating blendshapes into dynamic expression models (DEM), several approaches [118, 13, 66] have demonstrated state-of-the-art tracking performance on RGBD input. It can be observed that all of these tracking frameworks rely heavily on the quality of input depth data. However, existing consumer-grade depth sensors tend to provide increasingly unreliable depth measurements when the objects are farther [76]. Therefore, these methods [118, 13, 66] only work well at close range, where the depth map retains fine structural details of the face. In many applications, such as room-sized teleconferencing, the individuals tracked may be located at considerable distances from the camera, leading to poor performance with existing methods.

One way of addressing depth sensor limitations is to use color as in [19, 18]. These RGB-based methods require extensive training to learn a 3D shape regressor. The learned regressor serves

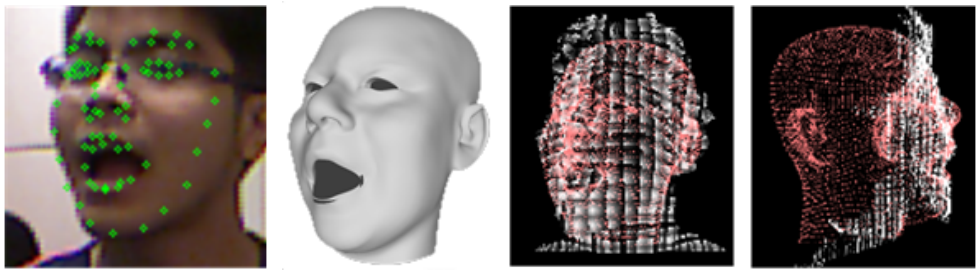
as a prior for DEM registration to RGB frames. Despite the high training cost, these methods have tracking results comparable to RGBD-based approaches. Although RGB-only methods are not affected by inaccurate depth measures, it is still challenging to track with high fidelity at large object-camera distances. This is in part due to reduced reliability of regression-based updates at lower image resolutions, when there is less data for overcoming depth ambiguity. Instead, we expect to achieve better tracking results if we were able to incorporate depth data while intelligently handling its inaccuracies at greater distances.

This motivates us to propose a robust RGBD face tracker combining the advantages of RGB regression and 3D point cloud registration. Our tracker is guided by a multi-stage 3D shape regressor based on random forests and linear regression, which maps 2D image features back to blendshape parameters for a 3D face model. This 3D shape regressor bypasses the problem of noisy depth data when obtaining a good initial estimate of the blendshape. The subsequent joint 2D+3D optimization matches the facial blendshape to both image and depth data robustly. This approach does not require an apriori blendshape model of the user, as shape parameters are updated on-the-fly.

We further improve our framework to address the difficulty in tracking extreme poses, e.g. profile-to-profile. To solve this problem, we propose a new unified RGBD face tracking framework. A critical aspect of this new framework for making the tracking robust across such ”extreme” conditions is (1) the ability to accurately and rapidly determine the visibility of landmarks and (2) make the visibility estimation an integral part of the regressor training and online inference.

Specifically, we propose two approaches to integrate pose and expression regression with visibility estimation. First, we leverage the local random forest framework to jointly predict landmark visibility and landmark displacement by a forest of decision trees that minimize a joint regression-classification entropy at no extra cost. While traditional visibility estimation approaches, based on 3D object models, leverage geometric and deterministic visibility of surface points, they do not readily take into account the uncertainty of the head pose, motion, and the surface structure

(expression, subject identity) that arise in tracking. As a consequence, landmarks predicted based on deterministic surface structure may not be sufficiently accurate. In contrast, our forest-based framework can model the correlation between displacement of landmark and the probability of it being visible. These probability scores are used as the feature vector for the global parameter regression. Furthermore, a single regression mapping may not fit all data samples of different poses. Therefore we propose the second extension by learning a visibility specific mixture of regression experts, where the choice of regressor is directly determined based on the visibility feature. These enhancements bring significant improvement in tracking large-posed faces.



(a) (b) (c) (d)

Figure 2.1: A tracking result of our proposed method. (a) The 3D landmarks projected to color frame. (b) The 3D blendshape. (c) The 3D frontal view, with the blendshape model in red and input point cloud in white. (d) The 3D side view.

2.2 Overview

Fig. 2.3 shows the pipeline of the proposed face tracking framework, which follows a coarse-to-fine multi-stage optimization design. In particular, our framework consists of two major stages: shape regression and shape refinement. The shape regressor performs the first optimization stage, which is learned from training data, to quickly estimate shape parameters $\theta = (R, T, e)$ from the RGB frame (cf. Section 2.3). Then, in the second stage, a carefully designed optimization is performed on both the 2D image and the available 3D point cloud data to refine the shape parameters,

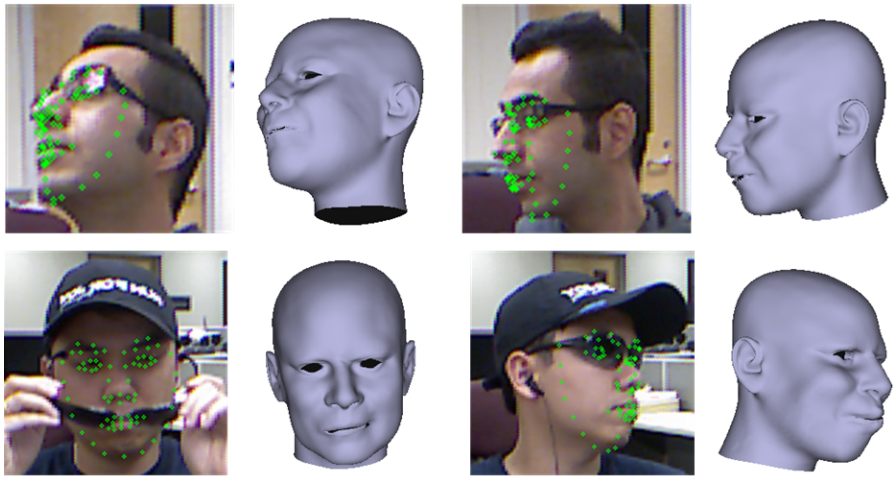


Figure 2.2: A few tracked frames, when the faces are at large pose.

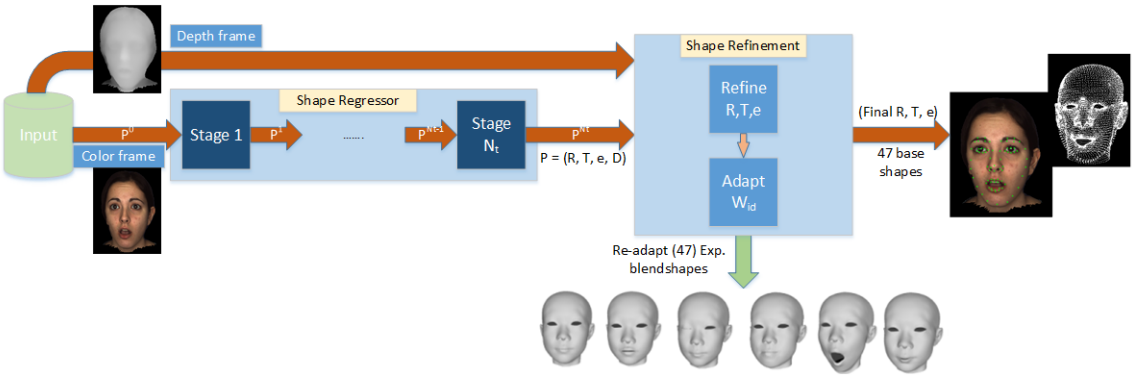


Figure 2.3: The pipeline of proposed face tracking framework.

and finally the identity parameter w_{id} is updated to improve shape fitting to the input RGBD data (cf. Section 2.4).

The 3D shape regressor is the key component to achieve our goal of 3D tracking at large distance, where quality of the depth map is often poor. Unlike the existing RGBD-based face tracking works, which either heavily rely on the accurate input point cloud (at close distances) to model shape transformation by ICP [118, 13] or use off-the-shelf 2D face tracker to guide the shape transformation [66], we predict 3D shape parameters directly from the RGB frame by the developed

3D regressor. This is motivated by the success of the 3D shape regression from RGB images used in [19, 18]. The approach is especially meaningful for our considered large distance scenarios, where the depth quality is poor. Thus, we do not make use of the depth information in the 3D shape regression to avoid profusion of inaccuracies from the depth map.

Initially, a color frame I is passed through the regressor to recover shape parameters θ. The projection of N_l ($N_l = 73$) landmarks vertices of the 3D shape to image plane typically does not accurately match the 2D landmarks annotated in the training data. We therefore include 2D displacements D in (2.3) into the parameter set and define a new global shape parameter set $P = (\theta, D) = (R, T, e, D)$. The advantages of including D in P are two-fold. First, it helps train the regressor to reproduce the landmarks in the test image similar to those in the training set. Second, it prepares the regressor to work with unseen identity which does not appear in the training set [18]. In such case the displacement error D may be large to compensate for the difference in identities. Hence we called the regressor "identity-agnostic". The regression process can be expressed as $P^{out} = f_r(I, P^{in})$, where f_r is the regression function, I is the current frame, P^{in} and P^{out} are the input (from the shape regression for the previous frame) and output shape parameter sets, respectively. The coarse estimates P^{out} are refined further in the next stage, using more precise energy optimization added with depth information. Specifically, $\theta = (R, T, e)$ are optimized w.r.t both the 2D prior constraints provided by 2D landmarks estimated by the shape regressor and the 3D point cloud. Lastly, the identity vector w_{id} is re-estimated given the current transformation.

2.3 3D Shape Regression

As mentioned in Section 2.2, the shape regressor regresses over the parameter vector $P = (R, T, e, D)$. To train the regressor, we must first recover these parameters from training samples, and form training data pairs to provide to the training algorithm. In this work, we use public face databases from [54, 106, 20] for training.

2.3.1 Shape Parameter Extraction

We follow the parameter estimation process in [19]. Denoting Π_p the camera projection function from 3D world coordinates to 2D image coordinates, (R, T, w_{id}, w_{exp}) are first extracted by minimizing the 2D errors in each sample:

$$\min_{R,T,w_{id},w_{\exp}} \sum_{i=1}^{N_l} \left\| \Pi_p \left(R\left(C_r \times_2 w_{id}^T \times_3 w_{\exp}^T\right)_i + T \right) - l_i \right\|^2 \tag{2.1}$$

where $\{l_i | i = 1, \ldots, N_l\}$ are the ground truth landmarks of the training data and $N_l = 73$. Note that w_{exp} *will be discarded* since we only need w_{id} to generate the individual expression blendshapes $\{B_j\}$ of the current subject as in (1.2) for later optimization over (R, T, e).

With the initially extracted parameters in (2.1), we refine w_{id} by alternatingly optimizing over w_{id} and (R, T, w_{exp}). Particularly, we first keep (R, T, w_{exp}) fixed for each sample, and optimize over w_{id} across all the samples of the same subject:

$$\min_{w_{id}} \sum_{k=1}^{N_s} \sum_{i=1}^{N_l} \left\| \Pi_p \left(R_k \left(C_r \times_2 w_{id}^T \times_3 w_{k_{\exp}}^T \right)_i + T_k \right) - l_{k_i} \right\|^2 \tag{2.2}$$

where N_s denotes the total number of training samples for the same subject. Then for each sample we keep w_{id} fixed and optimize over (R, T, w_{exp}) as in (2.1). This process is repeated until convergence. We empirically observe that running the above process for three iterations gives reasonably good results. We then can generate user-specific blendshapes $\{B_i\}$ as in (1.2).

Finally, we recover the expression weights e by minimizing the 2D error over (R, T, e) again:

$$\min_{R,T,e} \sum_{i=1}^{N_l} \|D_i\|^2 \tag{2.3}$$

where $D_i = \Pi_p(S_i) - l_i$ and S_i is a 3D landmark vertex of the blendshape corresponding to l_i. From (2.3), we also obtain the 2D displacement vector $D = \{D_i\}$ as a by-product. Eventually, following [18], for each training data sample, we generate a number of guess-truth pairs $\left\{I_i, P_i^0, P_i^g\right\}$, where the guessed vector P_i^0 is produced by randomly perturbing the ground truth parameters P_i^g extracted through the above optimization. In this way, we create N training pairs in total.

2.3.2 Shape Regression Training

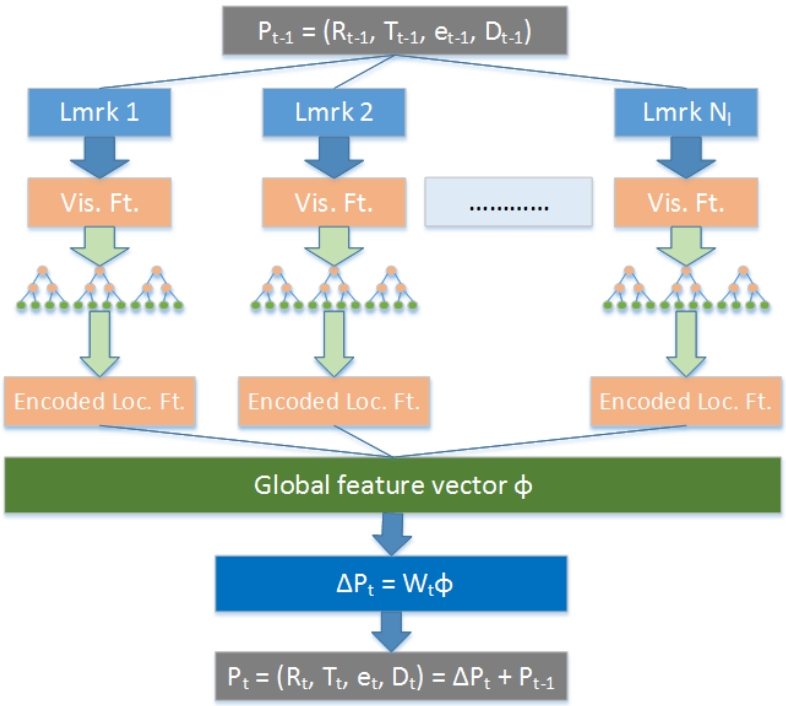


Figure 2.4: One regression stage, which updates global parameter vector P. It consists of two sub-stages: the local sub-stage encodes visual features around each individual landmark into local features, and the global sub-stage aggregates these local features to predict the parameter update ΔP_t.

Given the training pairs from the previous section, we follow the feature extraction and shape regression method in [92], which combines local binary features extracted using the trained random forests of all the landmarks. The local binary features are aggregated into a global feature vector which is then used to train a linear regression model to predict the shape parameters. In our work, we train the regressor to predict (R, T, e, D) simultaneously, directly from the input RGB frame in contrast to [92] where the regressor simply updates only the 2D displacements. The data flow diagram of one regression stage is illustrated in Fig. 2.4.

Algorithm 1 shows the detailed training procedure. In particular, we calculate the 2D landmark

positions from the shape parameters, and for each landmark l_i, we randomly sample pixel intensity-difference features [22] within a radius r_i. These pixel-difference features are then used to train a random forest $Forest_i$. For every training sample M_k, we pass it through the forest and recover a binary vector $F_{k,i}$ which has the length equal to the number of leaf nodes of the forest. Note that these local displacement updates $\{D_i\}$ are discarded, instead, the local forests encode visual features into local binary features. Each node that responds to the sample will be represented as 1 in $F_{k,i}$; otherwise it will be 0. The local binary vectors from N_l landmarks are concatenated to form a global binary vector Φ_k representing the training sample k. Then, the global binary feature vectors are used to learn a global linear regression matrix $\boldsymbol{W}$ which predicts the updating shape parameters ΔP from those binary global vectors. After that, the guessed shape parameters are updated and enter the next iteration.

Algorithm 1: The regressor training algorithm

Data: N training samples $M_k = \left\{I_k, P_k^0, P_k^g\right\}$
Result: The shape regressor

for $t \leftarrow 1$ **to** N_t **do**
  for $i \leftarrow 1$ **to** N_l **do**
    $Forest_i \longleftarrow$ TrainForest(l_i);
    for $k \leftarrow 1$ **to** N **do**
      $F_{k,i} \longleftarrow$ Pass(M_k, $Forest_i$);
    end
  end
  for $k \leftarrow 1$ **to** N **do**
    $\Phi^t\left(I_k, P_k^{t-1}\right) \leftarrow concat\left(F_{k,i}\right)$;
  end
  $\min \sum_{k=1}^{N} \left\|\Delta P_k^t - W^t \Phi^t\left(I_k, P_k^{t-1}\right)\right\|^2 + \lambda \left\|W^t\right\|^2$;
  for $k \leftarrow 1$ **to** N **do**
    $P_k^t \leftarrow P_k^{t-1} + W^t \Phi^t\left(I_k, P_k^{t-1}\right)$;
  end
end

Similar to [92], we let the regressor learn the best search radius r_i during training. The training

face samples have been normalized to the size of approximately 120x120 pixels, about the same size as the face captured by Kinect at 0.7m distance. Thus at runtime, we simply rescale the radius inversely proportional to the current z-translation T_z, making the regressor robust to distance.

The regressor shown in Fig. 2.4 and Algorithm 1 will be further improved to handle large poses by incorporating landmark visibility into prediction. The modifications are explained in the following section.

2.3.3 Pose-robust 3D Shape Regression

The approach described above assumes that all landmarks must be visible in the camera's field of view and does not take into account various out-of-plane head rotations, where a number of landmarks are self-occluded. Thus, the regressor cannot effectively predict shape parameters in such cases. One way to address this issue is to augment the regressor with landmark visibility to indirectly model large head poses. Specifically, the visibility of the i^{th} landmark is represented by a Bernoulli random variable $v_i \in \{0, 1\}$ together with its associated probability $p(v_i)$. Two different scenarios to integrate the landmark visibility information into the local regression framework and a novel global piecewise regression to improve the error rate is explained are discussed in details below.

3D Pose-deterministic Shape Regression

As a baseline model for visibility assessment, we used a traditional visibility test based on fixed, deterministic structure of the current face geometry S. Specifically, we change the way local binary features $F_{k,i}$ are aggregated. Given current parameters θ_k, we calculate the 3D shape S_k, and use z-buffer to determine which landmarks of S_k are visible ($p(v_{k,i} = 1) = 1$), then the local binary features is

$$\hat{F}_{k,i} = F_{k,i} \cdot p(v_{k,i} = 1) \tag{2.4}$$

This modification trains a global regression $\boldsymbol{W}$ capable of inferring shape parameters despite missing landmarks.

Joint Landmark Visibility-Displacement Prediction Local Forests

The extension described in the previous section is straightforward and works well in most cases. There are two drawbacks, however. First, calculating the 3D shape and z-buffer requires substantial computing power, especially with dense face models that we use, making the regressor less efficient. Second, if the transition between frames is not smooth enough, visibility of landmarks might not be predicted accurately. We therefore propose a data-driven approach, in which we train a joint classification-regression random forest [47] for each landmark i to predict both its 2D displacement D_i and the visibility v_i together with its associated probability $p(v_i)$. Random decision trees are trained using standard information gain maximization procedure. The information gain from a split at node $\mathbb{S}$ is defined as

$$I(\mathbb{S}) = H(\mathbb{S}) - \sum_{i \in \{Left, Right\}} \frac{|\mathbb{S}_i|}{|\mathbb{S}|} H(\mathbb{S}_i), \tag{2.5}$$

where $H(\mathbb{S})$ is the joint entropy of D and v[1]

$$H(\mathbb{S}) = -\sum_{v} \int_{D} p(D, v|x) \log p(D, v|x) dD. \tag{2.6}$$

$H(\mathbb{S})$ can be decomposed into two terms, $H_v(\mathbb{S})$ and $H_r(\mathbb{S})$, which are the Shannon and joint differential entropy, respectively:

$$H_v(\mathbb{S}) = -\sum_{v} p(v|x) \log p(v|x), \tag{2.7}$$

$$H_r(\mathbb{S}) = -\sum_{v} p(v|x) \int_{r} p(D|v, x) \log p(D|v, x) dD, \tag{2.8}$$

[1]We drop subscripts *i, k* for clarity.

where $p(D|v,x) \triangleq \mathcal{N}(D;\mu_{D|v},\Sigma_{D|v}|v,x)$, $\mu_{D|v}$ and $\Sigma_{D|v}$ are the mean and covariance of D w.r.t. v. Thus, the joint differential entropy H_r in (2.8) can be rewritten as

$$H_r = \sum_v p(v|x)\left(\frac{1}{2}\log\left[(2\pi e)^2\left|\Sigma_{D|v}\right|\right]\right). \tag{2.9}$$

$H_v(\mathbb{S})$ and $H_r(\mathbb{S})$ have different value ranges. In order to balance two tasks of classification and regression in training, we calculate a normalized joint entropy term $\mathcal{H}(\mathbb{S})$ over the entropies of root node $\mathbb{S}_0$:

$$\mathcal{H}(\mathbb{S}) = \frac{1}{2}\left(\frac{H_v(\mathbb{S})}{H_v(\mathbb{S}_0)} + \frac{H_r(\mathbb{S})}{H_r(\mathbb{S}_0)}\right). \tag{2.10}$$

Local features $F_{k,i}$ are extracted as before, but the 1 bit is replaced by the probability $p_\tau\left(v_i = 1|x_k\right)$,

$$F_{k,i} = p_\tau\left(v = 1|x_k\right) = \frac{\aleph_{\ell(x_k)}(v=1)}{\sum\limits_{v=0,1}\aleph_{\ell(x_k)}(v)}, \tag{2.11}$$

of a landmark being visible at the leaf node $\mathbb{S}_{\ell(x_k)}$ of tree τ where x reaches; the leaf indicated by index $\ell(x_k)$ contains histogram $\aleph$ of v. Global features Φ_k are then collected to learn the global regression matrix $\boldsymbol{W}$.

Piecewise Global Regression

A single global linear regression matrix $\boldsymbol{W}$ as described above may not be able to map the feature space to the parameter space correctly due to highly nonlinearly varying poses in training data. We exploit the collection of landmark visibility probability from the previous section to partition the training data into different subsets, to form a series of piecewise linear regressions to better model the data.

The averaged probability of landmark i of sample x_k being visible is measured as

$$\bar{p}_{k,i} = \bar{p}(v_i = 1|x_k) = \frac{1}{\mathbb{T}_i}\sum_\tau p_\tau(v_i = 1|x_k), \tag{2.12}$$

where $\mathbb{T}_i$ is the number of trees in the local forest of landmark i. Probabilities $\{\bar{p}_{k,i}\}$ from N_l landmarks are aggregated into a feature vector $\bar{p}_k$ to be used in clustering. The data $(\bar{p}_1,..,\bar{p}_N)$

is split into N_c subsets $\left\{\bar{P}_1, .., \bar{P}_{N_c}\right\}$ with centers $\{\mu_1, .., \mu_{N_c}\}$. Finally, N_c regression matrices $\{W_c\}_{c=1}^{N_c}$ are learned, one for each subset of data, similar to [122].

To determine cluster assignments at run-time, we use the cluster representatives μ_i in the visibility feature space. Specifically, the most likely cluster that the incoming sample $\hat{p}$ belongs to is determined using nearest neighbor search:

$$c^*(\hat{p}) = \arg\min_c \|\hat{p} - \mu_c\| . \tag{2.13}$$

Fig. 2.5 demonstrates the advantage of using piecewise regression, leading to lower error rate compared to using a single linear regression across all training samples, especially in the case of translation, which is particularly important in tracking fast moving faces.

2.4 3D Shape Refinement

2.4.1 Facial Shape Expressions and Global Transformation

We simultaneously refine (R, T, e) by optimizing the following energy:

$$R, T, e = \arg\min E_{2D} + \omega_{3D} E_{3D} + E_{reg} \tag{2.14}$$

where ω_{3D} is a tradeoff parameter, E_{2D} is the 2D error term measuring the 2D displacement errors, E_{3D} is the 3D ICP energy term measuring the geometry matching between the 3D face shape model and the input point cloud, and E_{reg} is the regularization term to ensure the shape parameter refinement is smooth across the time. Particularly, E_{2D}, E_{3D} and E_{reg} are defined as

$$E_{2D} = \frac{1}{N_l} \sum_{i=1}^{N_l} \|\Pi_p \left(S_i \left(R, T, e\right)\right) - l_i^*\|^2 \tag{2.15}$$

$$E_{3D} = \frac{1}{N_d} \sum_{k=1}^{N_d} \left(\left(S_k \left(R, T, e\right) - d_k\right) \cdot n_k\right)^2 \tag{2.16}$$

$$E_{reg} = \alpha \|\theta - \theta^*\|^2 + \beta \left\| \theta - 2\theta^{(t-1)} + \theta^{(t-2)} \right\|^2 . \tag{2.17}$$

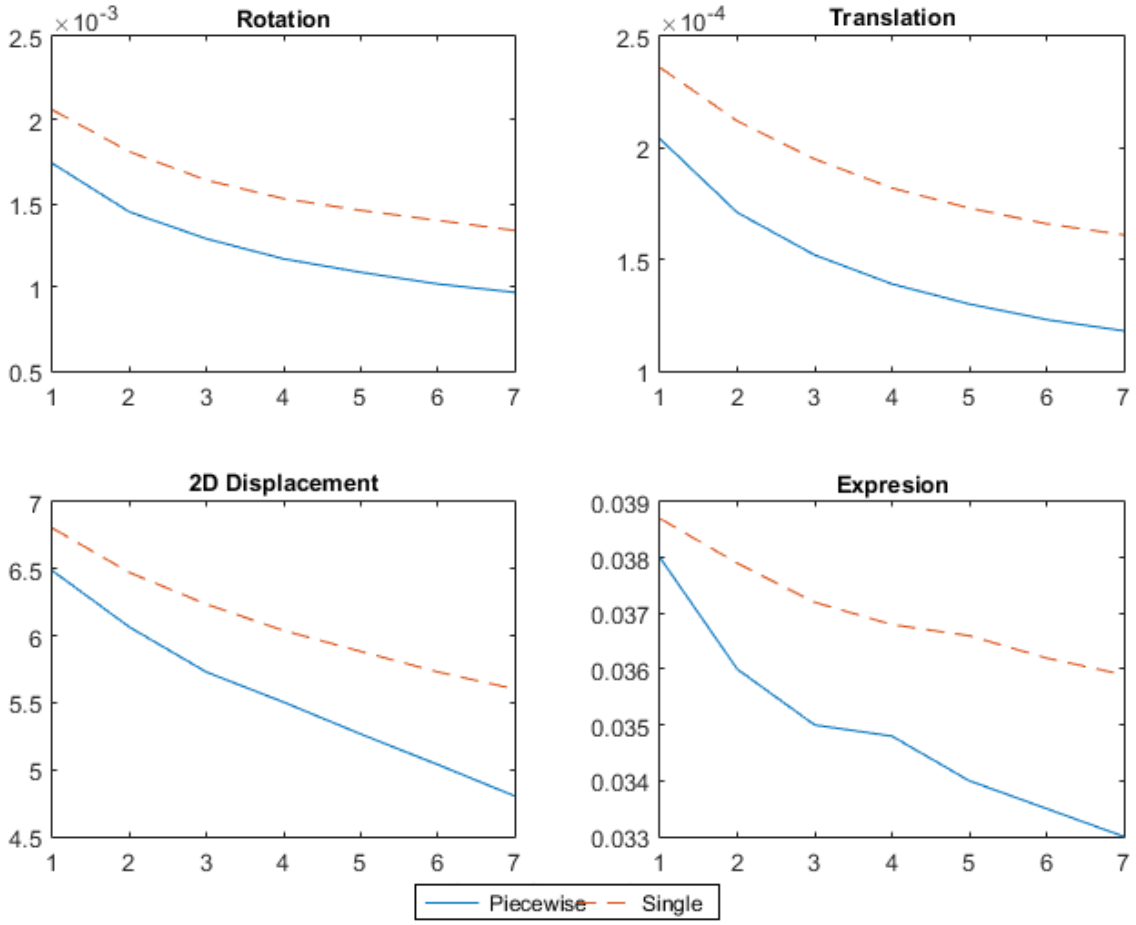


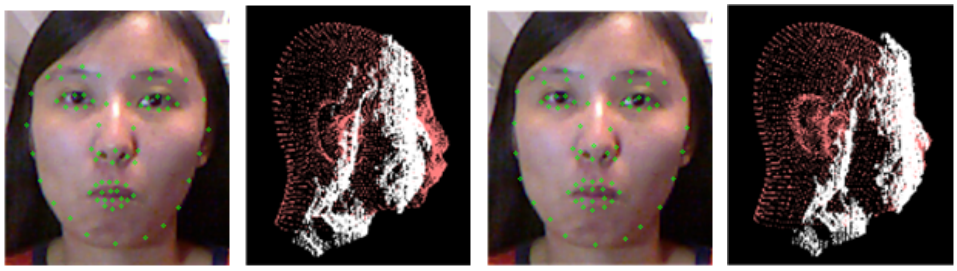
Figure 2.5: Comparing training error rates of single and piecewise regressions across seven iterations. The curves show *squared* errors w.r.t. different parameters. Rotation is represented by a unit quaternion, whereas translation is measured in meters.

In (2.15), the tracked 2D landmarks $\{l_i^*\}$ are computed from the raw shape parameters (R^*, T^*, e^*, D^*), which are usually quite reliable. In (2.16), N_d is the number of ICP corresponding pairs that we sample from the blendshape and the point cloud, and d_k and n_k denote point k in the point cloud and its normal, respectively. By minimizing E_{3D}, we essentially minimize the point-to-plane ICP distance between the blendshape and the point cloud [72]. This is to help slide the blendshape over the point cloud to avoid local minima and recover a more accurate pose. In (2.17), θ^* is the raw

output (R^*, T^*, e^*) from the shape regressor, $\theta^{(t-1)}$ and $\theta^{(t-2)}$ are the shape parameters from the previous two frames, and α and β are tradeoff parameters. The two terms in (2.17) represent a data fidelity term and a Laplacian smoothness term.

In our implementation, we iteratively optimize over the global transformation parameters (R, T) and the local deformation parameter e, which leads to faster convergence and lower computational cost. In the (R, T) optimization, ω_{3D} is set to 2; α, β are set to 100 and 10000 for R, 0.1 and 10 for T, respectively. For optimization over e, ω_{3D} is set to 0.5; α and β are both set to zero so as to maximize spontaneous local deformations. The non-linear energy function is minimized using the ALGLIB::BLEIC bounded solver[2] to keep e in the valid range of [0,1].

Fig. 2.6 gives an example to show the effect of the E_{3D} term. We can see that for the result without using E_{3D}, there is a large displacement between the point cloud and the model and there is also noticeable over-deformation of the mouth. This demonstrates that without using the 3D information, the 2D tracking may appear fine yet the actual 3D transformation is largely incorrect, because of the depth ambiguity problem.



(a) (b) (c) (d)

Figure 2.6: The effect of E_{3D} term. (a,b): The result without using E_{3D}. (c,d): The result using E_{3D}. Notice the displacement between the point cloud and the model, as well as the over-deformation of the mouth in (b).

[2] http://www.alglib.net/

2.4.2 Updating Shape Identity

In the last step, we refine the identity vector to better adapt the expression blendshapes to the input data. We solve for w_{id} by minimizing the following objective function:

$$w_{id} = \arg\min E'_{2D} + \omega_{3D} E'_{3D} \tag{2.18}$$

where

$$\begin{aligned} E'_{2D} &= \frac{1}{N_l} \sum_{i=1}^{N_l} \left\| \Pi_p \left(R\big(C_r \times_2 w_{id}^T \times_3 \gamma^T\big)_i + T \right) - l_i^* \right\|^2 \\ E'_{3D} &= \frac{1}{N_d} \sum_{k=1}^{N_d} \left\| R\big(C_r \times_2 w_{id}^T \times_3 \gamma^T\big)_k + T - d_k \right\|^2 \end{aligned} \tag{2.19}$$

with $\gamma = (1 - \sum_{j=1}^{N_e-1} e_j) u_{\exp_0} + \sum_{j=1}^{N_e-1} e_j u_{\exp_j}$.

Note that E'_{3D} is the point-to-point ICP energy and it behaves slightly differently from E_{3D} in (2.16). Minimizing E'_{3D} helps align the blendshape to the point cloud in a more direct way on the surface to recover detailed facial characteristics.

In our experiments, we empirically set ω_{3D} to 0.5, meaning that we give more weight to the 2D term to encourage the face model to fit closer to the tracked landmarks, especially the face countour. Gradient-based optimizations such as BFGS are ineffective toward this energy, and thus we run one iteration of coordinate descent at each frame to stay within the computational budget. We find that w_{id} usually converges in under 10 frames after tracking starts. To save computational time, we set a simple rule in which updating identity stops either after w_{id} converges or after 10 frames.

Fig. 2.7 shows some results on adapting the identity parameter over time. After a few iterations of updating w_{id}, the face model fits significantly better to each individual subject.

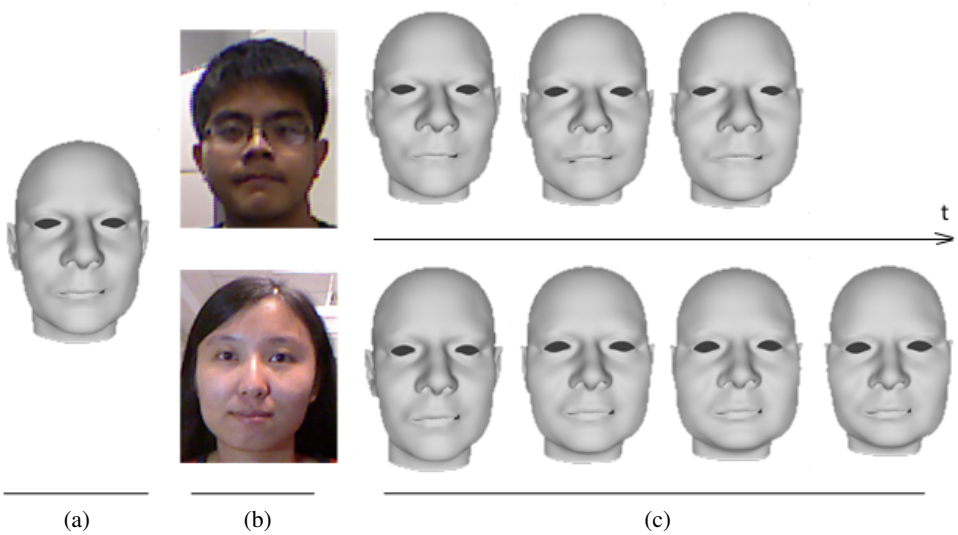


Figure 2.7: Adapting identity over time. (a) The common initial base shape. (b) Appearances of two testers. (c) For the male tester, the identity parameter w_{id} converges after three frames, compared to the female tester's four frame convergence.

2.5 Evaluation

We carried out extensive tracking experiments on synthetic RGBD sequences and real videos captured by a Kinect(v1) camera to test our trackers driven by four of our proposed regressors. The tracker using the original shape regressor in Section 2.3.2 is called **3DLGR**. The tracker with pose-deterministic regressor (Section 2.3.3) is called **3DLGR-Dt**. The tracker with joint classfication-regression forest single regression is henceforth denoted as **3DLGSR** while the one using piecewise regression is denoted as **3DLGMR**.

We separate our experiments into two categories: *near-frontal tracking* and *profile-to-profile tracking*. This is because **3DLGR** always outperforms the other three models on frontal face videos, as it was trained explicitly to track face with near-frontal pose. The other three models must handle a much larger face manifold while having the same capacity, hence their near-frontal tracking performance is not as good as **3DLGR**.

Test data includes synthetic and real RGBD sequences. Real sequences were captured with a Kinect v1 camera, while synthetic sequences were rendered from the BU4DFE dataset [127]. Details of synthetic sequence rendering can be found in the following experiments.

2.5.1 Near-frontal Face Tracking Experiments

We compared the tracking performance of **3DLGR** to that of RGB-based trackers DDER[18], CoR[129] and RLMS[101] in terms of average root mean square error (RMSE) in pixel positions of 2D landmarks. In the tracking context, we evaluated trackers' robustness by comparing the proportions of unsuccessfully tracked frames.

Evaluations on Synthetic Data

The BU4DFE dataset [127] contains sequences of high-resolution 3D dynamic facial expressions of human subjects. We rendered these sequences into RGBD to simulate the Kinect camera [76] at three distances: 1.5m, 1.75m and 2m with added rotation and translation. In total, we collected tracking results from 270 sequences. The dataset does not provide ground truth, so we used the RLMS tracker [101], which works well on BU4DFE sequences, to recover 2D landmarks on the images rendered at 0.6m, which were then reprojected to different distances and treated as ground truth.

The overall evaluation results are shown in Table 2.1. Our tracker performed comparably to the state-of-the-art CoR [129] and outperformed the blendshape-based DDER [18]. CoR did not produce results for sequences at 1.75m and 2m, with the faces too small for it to handle.

Experiments on Real Data

We compared the tracking performance of our approach to other methods on 11 real sequences at various distances, with different lighting conditions, complex head movements as well as facial

Table 2.1: Evaluation results of the proposed method and other face trackers on BU4DFE dataset. RMSE is measured in pixels.

Dataset	DDER [18]	CoR [129]	3DLGR (Ours)
BU4D (1.5 m)	2.20	**1.05**	1.27
BU4D (1.75 m)	1.94	n/a	**1.14**
BU4D (2.0 m)	1.76	n/a	**1.14**

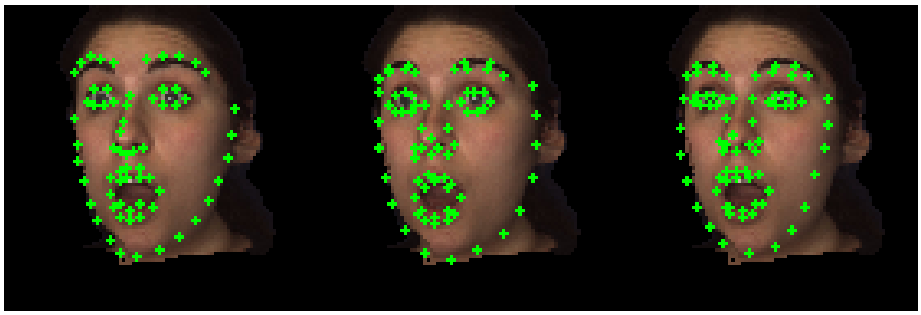


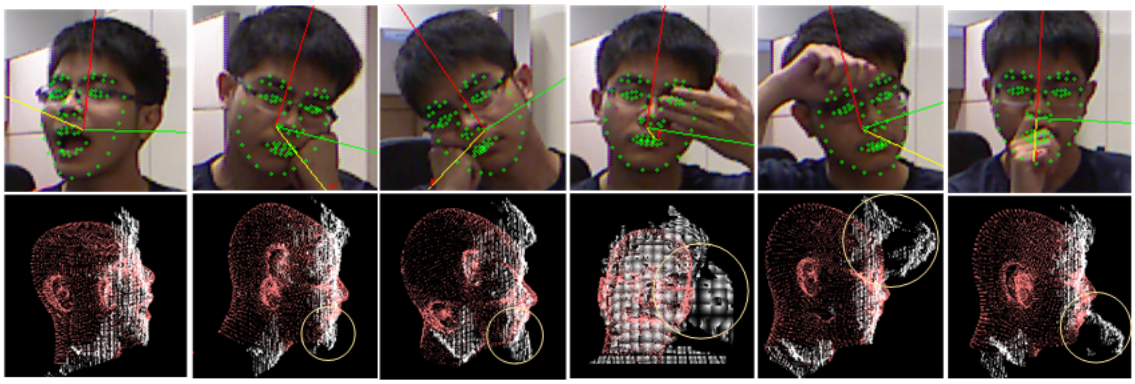
Figure 2.8: A sample from BU4DFE dataset, rendered at 1.5m. From left to right: results by CoR, DDER and our tracker, 3DLGR.

expressions. We used RLMS to recover the ground truth, and manually labeled the frames that were incorrectly tracked.

The results are shown in Table 2.2. For RLMS, we only considered the performance on frames that had been manually labeled, since its results were otherwise used as ground truth. Note that the inclusion of RLMS is mainly used as a reference and does not reflect its true performance, as only incorrectly tracked frames were measured. Once again, our method outperformed DDER and was very close to CoR. The consistent error values demonstrated that our tracker is stable, particularly under large rotations or when the face is partially covered, as illustrated in Fig. 2.9 and Fig. 2.10.

To better assess the robustness of each tracker, we compared the percentage of aggregated lost frames from all sequences in Table 2.3. The mistracked frames were decided either by empty output, or by large RMSE ($RMSE > \tau$, with $\tau = 10$). We also did not count sequences *luc03* for DDER, nor *luc03* and *luc04* for CoR, toward their overall percentages because the faces were not registered correctly from the beginning, which was perhaps largely due to the face detector failing

Figure 2.9: Each group of four shows results of four trackers on the same frame. From left to right: RLMS, CoR, DDER and our method. Our tracker and RLMS can handle occlusion by hair. In general, our tracker is robust to large rotation and it models realistic facial deformations.



(a) (b) (c) (d) (e) (f)

Figure 2.10: Example showing that the proposed tracker 3DLGR can handle partial occlusion of the face. The first row shows the resulting projected landmarks and the head orientation as 3 axes (red, green, yellow axes are yaw, pitch and roll, respectively). The second row shows the 3D view of the blendshape model (in red) and the input point cloud (in white) of each corresponding frame. Except (c) where the frontal view is shown, (a,b,c,e,f) show the side view. In each frame, the occlusion on the point cloud is circled in yellow. Tracking performance is not measured for this video and it is not included in Table 2.2, because we recorded this sequence after we had finished all the benchmarks.

to locate the face correctly. This showed that the 2D+3D optimization combination of our method provides robust tracking overall.

2.5.2 Profile-to-profile Face Tracking Experiments

Regressor Training Data

We use real RGB image samples from the FaceWarehouse [20], Labeled Face in the Wild [54] and GTAV [106] datasets. However, these datasets do not contain samples depicting large head poses. Thus, after fitting the 3D blendshape model to each sample to extract shape parameters, the sample image is artificial triangulated around the 3D head shape and rotated to create synthetic large-posed images from real images. We also render synthetic large-posed samples from the BU4DFE database [127] in order to increase the variation of face poses in the training set.

Table 2.2: Evaluation results of the proposed method and other face trackers on real videos. RMSE is measured in pixels.

Dataset	DDER [18]	CoR [129]	RLMS [101]	3DLGR (Ours)
dt01	9.65	4.15	6.04	4.51
ar00	3.41	66.72	7.41	2.36
dt00	3.57	1.65	4.63	2.29
my01	5.61	2.79	4.35	2.89
fw01	6.5	3.27	36.11	4.85
fw02	5.34	1.80	2.56	3.50
luc01	4.96	2.38	5.86	3.49
luc02	3.95	1.51	2.04	3.02
luc03 (2m)	37.17	n/a	1.67	1.77
luc04 (2m)	2.63	62.45	n/a	1.84
luc05	3.39	2.39	3.44	2.88

Table 2.3: The overall percentage of lost frames during tracking from all real videos.

DDER [18]	CoR [129]	RLMS [101]	3DLGR (Ours)
2.21%	7.22%	3.61%	**0.74%**

Experiment Settings

We compared the tracking performance of our two proposed models: **3DLGSR** and **3DLGMR**, against the baseline tracker using 3D geometry described in Section 2.3.3, **3DLGR-Dt**, as well as the near-frontal 3D face tracker **3DLGR**. We measure the 2D landmark error as RMSE (in pixels) w.r.t. 2D ground truth landmarks. Due to the lack of a publicly available profile-to-profile 3D face tracking software, we alternatively compare our methods to other 2D face alignment methods PMCDS [128] and TSPM [135], and a recent 3D dense face alignment method 3DDFA [134]. All three methods have been proven as capable of profile face registration. Particularly, PMCDS is configured in tracking mode, in which the facial landmarks of the previous frame are used to initialize the registration algorithm on the new frame. On the other hand, 3DDFA requires the exact bounding box of face region to be provided for each frame in order to work properly. We use the pre-trained models provided by the authors in all of our tests.

Evaluation on Synthetic Data

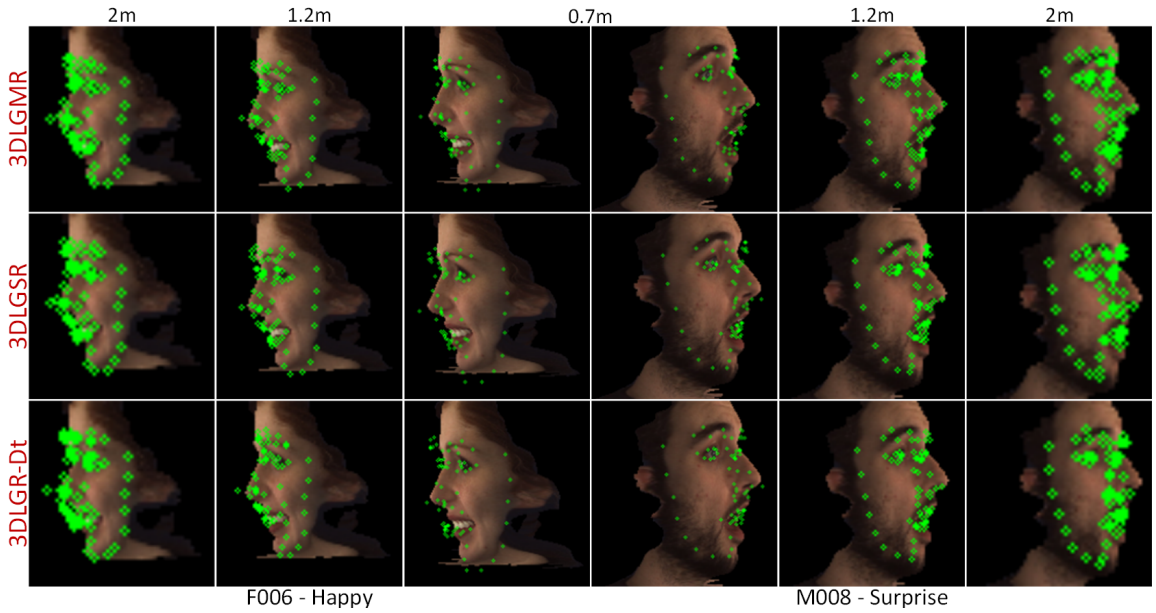


Figure 2.11: Tracking results on two sample frames from the BU4DFE dataset, rendered at different distances. From top to bottom, the figure shows results of 3DLGMR, 3DLGSR and 3DLGR-Dt, respectively. Sub-images were rescaled to the same size, therefore the landmark circles in the 2m samples appear larger than those in the 1.2m samples, which in turn are larger than those in the 70cm samples. Notice the differences in eyebrow and mouth areas in the 70cm frames, where 3DLGMR accurately registers both poses and mouth deformations.

We rendered BU4DFE sequences into RGBD to simulate the Kinect(v1) at six distances: 70cm, 1.2m, 1.5m, 1.75m, 2m and 2.5m with profile-to-profile head rotations, hence we can observe the same sequence of facial movements at different distances and in different resolutions as shown in Fig. 2.11. In total, we collected tracking results from 480 sequences. The dataset does not provide ground truth, hence we used the 3DLGR tracker to recover 3D landmarks on the frontal images of stationary head pose rendered at 0.7m, which were then rotated, readjusted using ICP, and reprojected to different distances and treated as ground truth.

As shown in Table 2.4, 3DLGMR achieves the lowest RMSE on average compared to other blendshape-based methods, including 3DLGSR. It also performs the best overall on sequences at

Table 2.4: Evaluation results on the profile-to-profile BU4DFE sequences. *RMSE* is measured in raw pixels, best results are marked in bold. The 2D error becomes smaller as the scene depth increases because of the camera perspective projection. TSPM did not work with low-resolution face images.

Dataset	PMCDS[128]	3DDFA[134]	TSPM[135]	3DLGR	**3DLGR-Dt**	**3DLGSR**	**3DLGMR**
BU4D (70 cm)	24.05 ± 9.68	3.86 ± 0.69	5.62 ± 0.82	5.85 ± 4.38	4.30 ± 1.01	4.59 ± 1.02	**3.81** ± 0.95
BU4D (1.2 m)	14.90 ± 7.60	2.36 ± 0.41	3.68 ± 0.48	2.88 ± 1.31	2.37 ± 0.59	2.51 ± 0.62	**2.28** ± 0.61
BU4D (1.5 m)	12.08 ± 6.54	1.96 ± 0.34	3.75 ± 0.46	2.32 ± 0.85	1.90 ± 0.42	1.99 ± 0.39	**1.72** ± 0.33
BU4D (1.75 m)	10.42 ± 5.85	1.74 ± 0.30	n/a	2.15 ± 1.50	1.76 ± 0.58	1.81 ± 0.44	**1.55** ± 0.31
BU4D (2m m)	9.31 ± 6.70	1.57 ± 0.27	n/a	2.65 ± 2.38	1.68 ± 0.74	1.83 ± 0.91	**1.38** ± 0.23
BU4D (2.5 m)	7.63 ± 8.35	**1.36** ± 0.23	n/a	2.64 ± 2.38	1.61 ± 0.38	1.74 ± 0.59	1.39 ± 0.28

Table 2.5: Evaluation results on real profile-to-profile sequences. *RMSE* was measured in raw pixel values. PMCDS performed poorly on *ro01* amd *sj01*, thus its results are not included. TSPM also did not work with *sj01*, since this sequence captures low-resolution face.

Dataset	PMCDS[128]	3DDFA[134]	TSPM[135]	3DLGR	**3DLGR-Dt**	**3DLGSR**	**3DLGMR**
ad01	4.23 ± 5.83	4.67 ± 2.76	**3.69** ± 1.57	3.93 ± 3.09	4.26 ± 2.28	3.96 ± 1.76	3.89 ± 1.82
bn01	6.74 ± 5.89	2.73 ± 2.06	3.86 ± 1.39	2.53 ± 1.12	2.68 ± 0.82	2.79 ± 0.93	**2.36** ± 0.62
bn02	3.71 ± 2.62	7.03 ± 4.25	3.36 ± 0.52	**2.74** ± 1.29	3.43 ± 1.79	3.69 ± 2.92	3.26 ± 1.73
jp01	3.08 ± 3.32	3.90 ± 2.45	**2.94** ± 0.68	3.13 ± 2.62	3.60 ± 3.34	6.09 ± 7.62	**2.95** ± 1.07
ro01	n/a	8.29 ± 6.68	**5.33** ± 6.26	7.94 ± 8.83	8.30 ± 11.02	12.54 ± 15.76	6.21 ± 2.02
sj01	n/a	5.04 ± 2.01	n/a	3.11 ± 2.09	**1.49** ± 0.53	1.90 ± 0.96	1.76 ± 0.76
sj02	3.49 ± 1.21	5.07 ± 2.54	3.70 ± 1.57	3.24 ± 1.12	**2.55** ± 0.62	2.99 ± 0.80	2.89 ± 0.72
sy01	7.17 ± 5.53	8.11 ± 4.94	4.06 ± 1.62	3.37 ± 1.19	3.71 ± 1.54	3.24 ± 1.04	**3.08** ± 0.81

distances ranging from 70cm to 2m, and only slightly worse than 3DDFA at 2.5m. In general, performance of 3DLGSR is not as good as 3DLGR-Dt and 3DLGMR as expected, because single regression model is not fully capable of modeling highly non-linear input features, while 3DLGR-Dt only takes binary features. Fig. 2.11 demonstrates a few sample frames from the test set, where 3DLGMR was able to achieve stable and highly accurate tracking results. These results reflect the advantage of combining local visibility prediction with piecewise linear regression models to predict shape parameters.

Evaluation on Real Data

Table 2.6: Tracking errors on large-posed frames from different datasets. 3DLGMR has substantial gain in majority of tests, except for *sj01*, *sj02* and *sy01*. In general, errors on large-posed frames are indeed larger than the average errors in Table 2.4 and 2.5.

Dataset	**3DLGR-Dt**	**3DLGSR**	**3DLGMR**
BU4D(70cm)	4.98 ± 1.07	5.14 ± 1.12	**3.97** ± 0.74
BU4D(1.2m)	2.69 ± 0.66	2.74 ± 0.62	**2.27** ± 0.62
BU4D(1.5m)	2.14 ± 0.49	2.16 ± 0.39	**1.77** ± 0.33
BU4D(1.75m)	1.97 ± 0.69	2.12 ± 1.10	**1.58** ± 0.33
BU4D(2m)	1.96 ± 0.86	1.99 ± 0.83	**1.59** ± 0.74
BU4D(2.5m)	1.95 ± 1.23	2.05 ± 1.30	**1.61** ± 0.83
ad01	5.50 ± 1.94	4.77 ± 1.45	**4.64** ± 1.48
bn01	2.79 ± 0.83	2.95 ± 0.89	**2.56** ± 0.57
bn02	3.50 ± 0.81	3.42 ± 0.69	**3.31** ± 1.23
jp01	4.17 ± 1.19	7.36 ± 7.97	**3.14** ± 0.68
ro01	9.11 ± 11.99	14.27 ± 15.56	**6.59** ± 2.05
sj01	**1.82** ± 0.54	2.55 ± 1.10	2.22 ± 0.93
sj02	**2.76** ± 0.60	3.35 ± 0.90	2.81 ± 0.60
sy01	4.30 ± 1.70	**3.38** ± 1.21	3.43 ± 0.65

Eight RGBD sequences capturing facial expressions, profile-to-profile head movements at different distances were recorded with a Kinect(v1) camera for testing, and ground truth landmarks were manually annotated on one among every 10 frames. The tracking results on these videos are

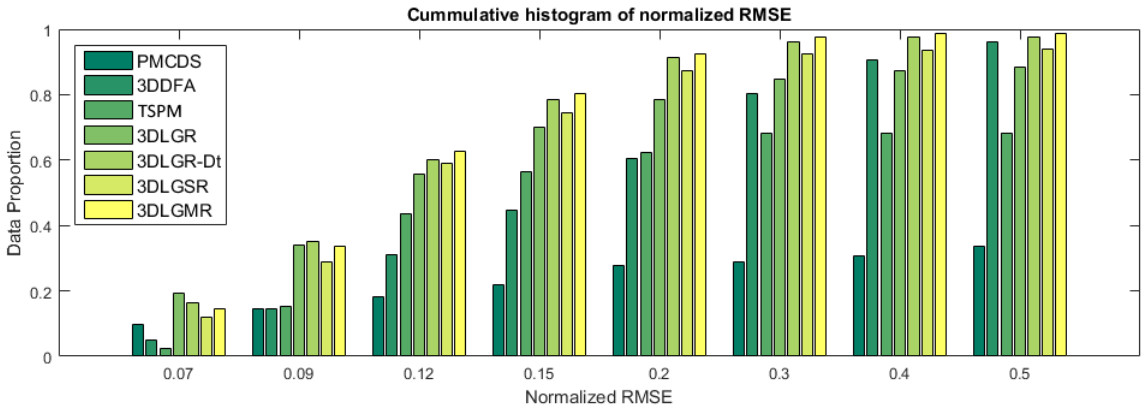


Figure 2.12: Cumulative Error Histograms of seven methods. Overall, 3DLGMR achieves the best tracking performance, followed by 3DLGR-Dt. In fact, 3DLGMR has the fastest CEH growth before reaching equilibrium of 98.6% at the 0.34 error bin, while 3DLGR-Dt reaches 97.7% at 0.36.

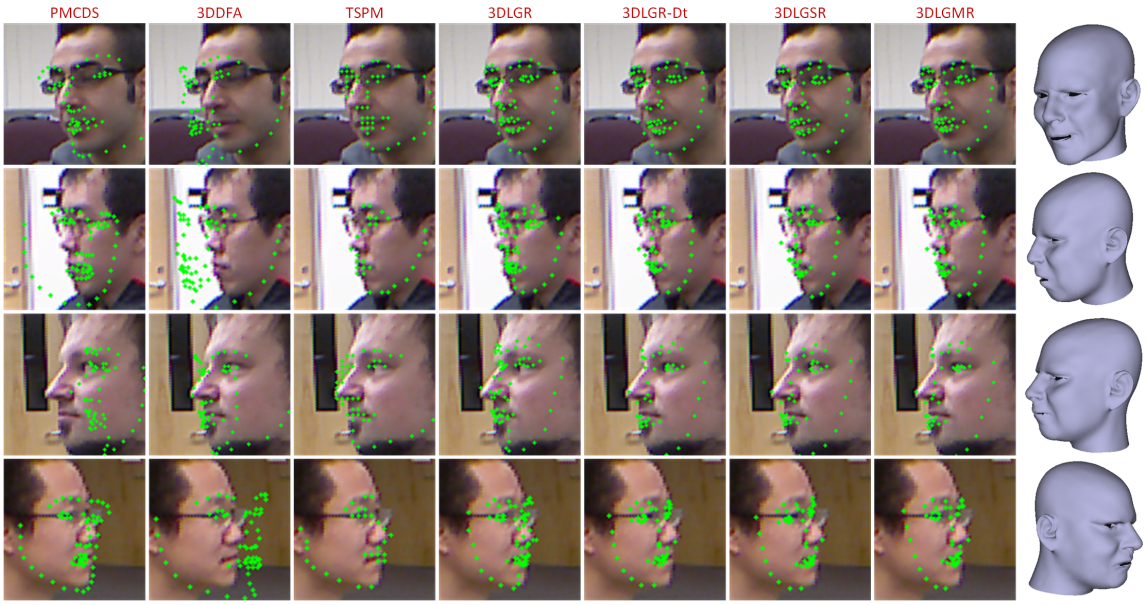


Figure 2.13: A few samples from real sequences. The 3D facial blendshapes produced by 3DLGMR are visualized in the last column.

shown in Table 2.5.

In these tests, 3DLGMR has the best results or only slightly below the best performers. Specifically, 3DLGR-Dt has slightly smaller errors on *sj01* and *sj02*. The fact that the tracked subjects in these two videos have darker skin tones, which affect the visibility prediction, may explain the lower performance of 3DLGMR as it maps the visibility features into shape parameters. Additionally, TSPM achieves lower average errors on *ad01*, *jp01* and *ro01* sequences, compared to 3DLGMR. This reflects the nature of the technique, because TSPM performs exhaustive search across all possible poses, thus it performs well when it is able to localize the face position, albeit at the cost of being considerably slower than 3DLGMR.

These errors were calculated from only successfully registered frames that do not account for tracking loss, which would very likely happen in unconstrained settings. To better understand the tracking performance, we measure additional Cumulative Error Histogram (CEH) metrics. CEH quantizes tracking error into κ bins, where the κ^{th} bin counts the number of frames with the error less than a threshold ϵ_k. The errors must be normalized over face size, which is specified as distance between outer-eye and mouth corners in our work. CEH is shown in Fig. 2.12.

In smaller error bins, both 3DLGR and 3DLGR-Dt have a slight advantage over 3DLGMR, but overall, 3DLGMR has the best histogram among all tested methods. This can be explained as these small error bins consist of near-frontal frames with which 3DLGR has been demonstrated to achieve really good performance. However, 3DLGR incurs larger errors in frames with large poses. 3DLGR-Dt is better at handling these poses, but with additional cost of using z-buffer. 3DLGMR performs better than 3DLGR-Dt in some cases, whilst retaining the computational advantage of 3DLGR. Table 2.6 shows that 3DLGMR outperforms 3DLGR-Dt in majority of frames with large head poses. All experiments on real RGBD videos demonstrate that 3DLGMR performs consistently well across different pose variations.

2.5.3 Implementation Details and Running Time

In the first frame when the tracker starts or restarts, the face is localized using the OpenCV face detector [113]. The tracker is written in native C++, parallelized with Intel Thread Building Blocks (TBB). We measure the running time of four tracker implementations driven by four versions of our shape regressor (each shape regressor has five sub-stages), excluding the identity adaptation process in (1.2) whose speed depends on the GPU (about 7ms on a Tesla K40c). Specifically, tested on an Intel Core i7 quad-core 3.4GHz CPU, 3DLGSR and 3DLGMR can process one frame in roughly 30ms, they are as fast as the original 3DLGR tracker. The baseline 3DLGR-Dt using z-buffer is slower, since each regression sub-stage takes additional 4ms to perform depth test in our implementation.

2.6 Related Work

Early work on articulated face tracking was based on classical Active Shape and Appearance Models (ASM, AAM) [29, 28, 74] and Constrained Local Models (CLM) [101] that fit a parametric facial template to the image. A common extension of these traditional methods to multi-view face tracking/alignment is to build multiple models for separate head poses, such as multi-view AAM [30], multi-view Direct Appearance Models [67]. TSPM [135] uses local detectors similar to CLM, but learns sparse tree structures of parts for different views. Yu et al. [128] use sparse TSPM to initialize CLM fitting. In practice, these parametric methods rely on the learned statistical shape model and may not generalize well to real world data.

In recent years, non-parametric shape models have achieved better results due to greater flexibility and efficiency. In one approach [33, 103, 132], individual landmarks are directly localized by creating response map for landmark locations, and landmarks are finally chosen by a mode seeking method. The second approach is shape regression, which maps visual features to landmark coordinates [121, 22, 92]. Although these shape regression methods have achieved state-of-the-art

performance, there is little effort in tackling multi-view face alignment. Recently, Xiong et al. [122] propose to learn multiple descent maps on different subsets of data to implicitly generalize shape models to different views.

Another popular approach is to use 3D deformable models, which are typically controlled by a set of deformation units, as priors. Past works [16, 34, 57, 82] employed simple, coarse 3D wireframe models, but they lack the presentation power to realistically model high-fidelity human face. More sophisticated models and registration techniques have been developed to obtain state-of-the-art 3D face reconstruction [10, 93, 94]. Zhu et al. [134] utilized these deformable models in a 3D dense face alignment framework across large poses using Convolutional Neural Networks.

More interestingly, recent approaches use 3D facial blendshapes [20] for real-time high-fidelity 3D face tracking. Such techniques have gained more attention lately due to the proliferation of affordable commodity depth sensing devices, such as the Kinect [76]. Several approaches [118, 13, 66, 109] based on blendshape DEM have demonstrated state-of-the-art tracking performance on RGBD input, or only depth input [59]. However, incoming data is assumed to be of high quality, thus they only work well in close range where fine details of the face are preserved. The RGB-based approaches by Cao et al. [19, 18] learn 3D shape regressor as priors for robust DEM registration. Although RGB-only methods are not affected by inaccurate depth measures, it is still challenging to track with high fidelity at large object-camera distances. This is in part due to reduced reliability of regression-based updates at lower image resolutions, when there is less data for overcoming depth ambiguity. These methods [19, 18] are robust by using the trained regressors, however, the shape regressors are unable to handle large angle poses, such as left/right profiles.

2.7 Summary

In this chapter we presented a novel RGBD face tracking framework, 3DLGR, that uses 3D facial blendshapes to simultaneously model head movements, as well as facial expressions in unconstrained environment, in forms of parameters including global rotation and translation, facial AU weights and identity weights. The tracker is driven by an efficient shape regressor, and in general it is robust to different conditions such as distance, lighting and low-quality input in general. We further proposed three variant of 3DLGR, namely 3DLGR-Dt, 3DLGSR and 3DLGMR, which can handle large-posed, profile-to-profile tracking.

Moreover, through extensive experiments on synthetic and real RGBD videos, our trackers performed consistently well in complex conditions and outperformed other contemporary state-of-the-arts especially in complex videos. Being real-time and fully automatic, our tracker can be readily deployed into various tasks in human machine interaction or virtual reality.

Chapter 3

Learning Facial Performance from Speech

Speech conveys not only the verbal communication, but also emotions, manifested as facial expressions of the speaker. In this chapter, we present deep learning frameworks that directly infer facial expressions from just speech signals. The objective is similar to face tracking in the previous chapter, but the task is more difficult, as the input is only speech audio lacking any visual features. Specifically, we propose recurrent neural networks to realize the time-varying contextual non-linear mapping between audio stream and micro facial movements to drive a 3D blendshape face model in real-time.

Our models not only activate appropriate facial action units (AUs) at inference to depict different utterance generating actions, in the form of lip movements, but also, without any assumption, automatically estimate emotional intensity of the speaker and reproduces her ever-changing affective states by adjusting strength of related facial unit activations.

In the baseline model, conventional handcrafted acoustic features are utilized to predict facial actions. Furthermore, we show that it is more advantageous to learn meaningful acoustic feature representation from speech spectrograms with convolutional nets, which subsequently improves the accuracy of facial action synthesis.

Experiments on diverse and challenging audiovisual corpora of different actors across a wide range of facial actions and emotional states show promising results of our approaches. Being independent of speaker and language, our generalized models are readily applicable to various tasks in human-machine interaction and animation.

This chapter contains materials from our published papers [81, 85].

3.1 Introduction

Human-machine interaction has been an active research area for decades, with the ultimate goal to make interaction between human-machine transparent. Speech, as a natural form of communication among various modes of interactions, is becoming more immersive, evidenced by the increasing popularity of virtual voice assistants in our daily lives. Furthermore, the audio recording carries not only the contextual sound units (phonemes), but also emotions of the speaker reflected in speed or intensity of her speech. Hence, it is beneficial for the computer to comprehend emotional states of the speaker, for instance, to make a joke when it perceives the user is happy, in an imaginary mutual human-machine conversation.

Moreover, there are various applications of speech-driven facial expression synthesis such as computer games, animated movies, teleconferencing, talking agents, among others. Traditional facial capture approaches have gained tremendous successes, reconstructing high level of realism. Yet, active face capture rigs utilizing motion sensors/markers are expensive and time-consuming to use. Alternatively, passive techniques capturing facial transformations from cameras, although less accurate, have achieved very impressive performance. There lies one problem with vision-based facial capture approaches, however, where part of the face is occluded, e.g. when a person is wearing a mixed reality visor, or in the extreme situation where the entire visual appearance is non-existent. In such cases, other input modalities, such as audio, may be exploited to infer facial actions. Indeed, research on speech-driven face synthesis has regained attention of the community in recent time. Latest works [58, 105, 107] employ deep neural networks in order to model the highly non-linear mapping from speech domain, either as audio or phonemes, to visual facial features. Particularly, in the approach by Karras et al. [58], the reconstruction of facial emotion is also taken into account to generate fully transformed 3D facial shapes. Their method explicitly

specifies the emotional state as an additional input beside raw audio data. However, the task of directly inferring facial actions and expressions from speech has not been addressed properly.

In this work, we aim to recreate a 3D virtual talking avatar that can make micro facial movements to reflect the time-varying contextual information and emotional intensities of the speaker carried in the input speech. Intuitively, this work is analogous to visual 3D face tracking [84, 83], however, it is more challenging as we try to map acoustic information to visual space, instead of conveniently relying on textural cues from input images. Moreover, speech-emanated facial movements involve different activations of correlated regions on the geometric surface, thus it is difficult to achieve realistic looking, emotion-aware facial deformation from speech sequence.

Thus, we propose a regression framework based on recurrent neural network (RNN) to estimate facial action unit parameters of a 3D blendshape face model from audio sequence, for real-time life-like facial animation. To tackle the difficulty of avatar generation, we utilize the blendshape model in the FaceWarehouse database (introduced in Section 1.3), which is purposefully designed with enough constraints on facial action units to ensure that, the facial shape would always look realistic given a specific set of blending coefficients. In addition, it can represent various emotional states, e.g. sadness, happiness, etc., without explicitly specifying them. We propose a baseline neural net model, in which engineered acoustic features are utilized as input to an RNN. Furthermore, in order to overcome the limitation of using handcrafted features, which are inherently lossy and may cause the loss of important information, we propose to learn meaningful feature representation from raw spectrograms with convolutional neural net (CNN), which indeed leads to significant performance gain. It also simplifies the data processing pipeline, and speeds up facial expression inference. Experiments on different challenging audiovisual corpora demonstrate promising results of our approach in real-time speech-driven 3D facial animation.

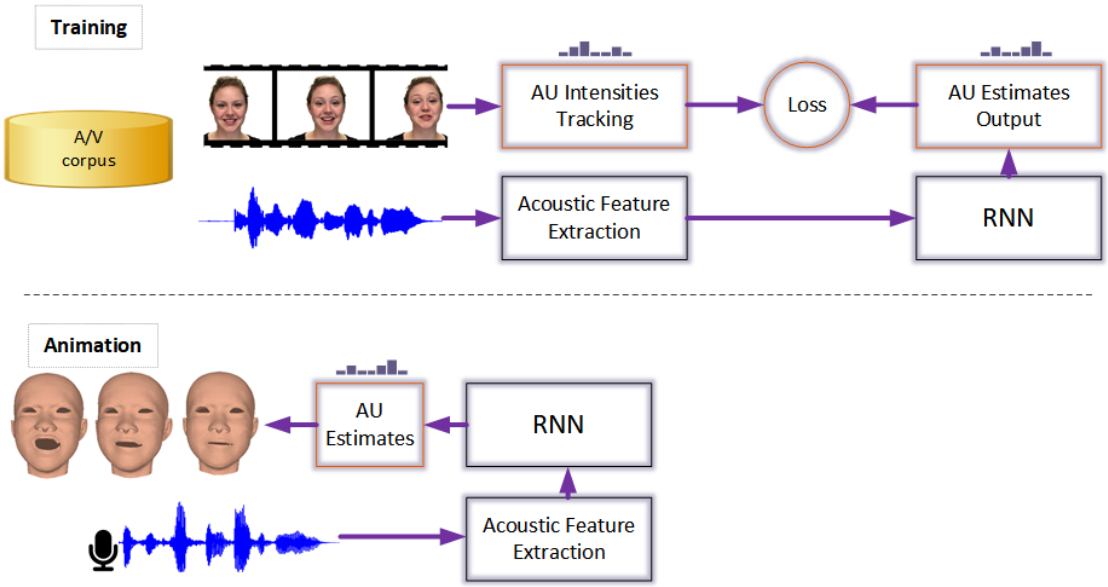


Figure 3.1: The general learning framework for speech-driven facial expression synthesis. At test time, the model only accepts audio input. In the baseline model, a set of handcrafted features are extracted. Whereas in the end-to-end model, feature representation is learned directly from spectrogram.

3.2 Overview

Fig. 3.1 illustrates the general framework of our proposed speech-driven 3D facial expression synthesis approach, which includes a training phase and an animation stage. In the training phase, the speech-to-facial parameters mapping is learned by an RNN from a set of audiovisual corpora, formulated as a regression problem. The input to our system can be any arbitrary speech of any length $X = \{x^t | t = 1..T\}$ be an audio sequence, where T is the number of feature frames (each frame is an acoustic feature vector in the baseline, or a spectrogram in the end-to-end model). Denote $E = \{e^t | t = 1..T\}$ as the corresponding output sequence of facial action unit intensities, e is a 46-D vector of AU parameters, defined in (1.3) in Section 1.3. The mapping function $\mathcal{F} : X \rightarrow E$, is realized as an RNN. In this work, we emphasize on real-time applications, hence we only use unidirectional recurrent network. Moreover, in order to get a comprehensive understanding

whether knowing future data is more advantageous, we also compare these unidirectional models to their bidirectional counterparts. More details are discussed in Section 3.6.

In the animation phase, the trained recurrent model converts the input sequence of audio signal to facial action parameters to drive a 3D blendshape face model. Note that our models only estimate facial actions, independent of speaker identity. In our experiments, we reconstruct facial shapes of a single generic identity to maintain the consistency of facial expression synthesis, and to facilitate fair comparison between different models.

As we only use low-level acoustic features (or not at all, in the end-to-end model) to learn the universal facial expression subspace from speech, our models are not tied to any particular language, and it can be easily extended given more training samples. In the baseline model, acoustic features are extracted using a combination of conventional pre-defined audio processing functions, e.g. MFCC (cf. 3.3). The discriminative recurrent neural network simply maps these features to facial action unit intensities. In practice, these engineered features have been proven to be able to convey the speech content (i.e. talking - lip movements), and they have been used successfully to produce photo-real lip-syncing syntheses [43, 105]. However, most of these features are originally designed to diminish emotional states of the speaker carried in her speech, thus the ability to accurately predict emotional facial actions of the deep model is more or less reduced. To circumvent this limitation, we propose to learn more meaningful feature presentation pertaining to facial expressions directly with CNN (cf. 3.4). This approach indeed leads to significant performance gain according to the experimental results.

3.3 Facial Expression Estimation from Handcrafted Features

3.3.1 Feature Extraction

In the basline model, we extract Mel-scaled spectrogram, Mel frequency cepstral coefficients (MFCCs) and chromagram from the audio sequence. Mel-scaled spetrogram and MFCCs are standard acoustic features proven to be very effective in presenting the contextual information, whereas chromagram is necessary to determine the pitch in the speech, which reflects the affective states of the speaker throughout the whole sequence.

We assume that every input audio sequence is synchronized to the corresponding video at 30 FPS and the audio sampling rate is at 44.1 kHz. Thus, for every video frame, there are 1,470 corresponding audio samples. We include additional samples from the previous video frame, such that for each video frame there is enough audio data to extract three windows of 25ms each, with hop length of 512 samples. In every audio window, values of 128 Mel bands, 13 Mel frequency cepstral coefficients and their delta and delta-delta coeffcients, and 12 chroma bins, are extracted. In summary, the input feature vector for every video frame has 537 dimensions, and each variable is normalized to zero mean - unit variance. Fig. 3.2 illustrates different feature sequences extracted from videos of the same actor speaking the same sentence in different emotional states.

3.3.2 Model Learning

The recurrent neural network in the baseline model consists of two hidden layers constructed with LSTM cells, illustrated in Fig. 3.3 ,to model the highly non-linear mapping between acoustic features and facial expression coefficients. In our experiments on standard audiovisual corpora, we found that using more or less hidden layers leads to higher errors, thus only two-layered LSTM-RNN is thoroughly evaluated in this paper. Its architecture and training details are presented in the Appendix.

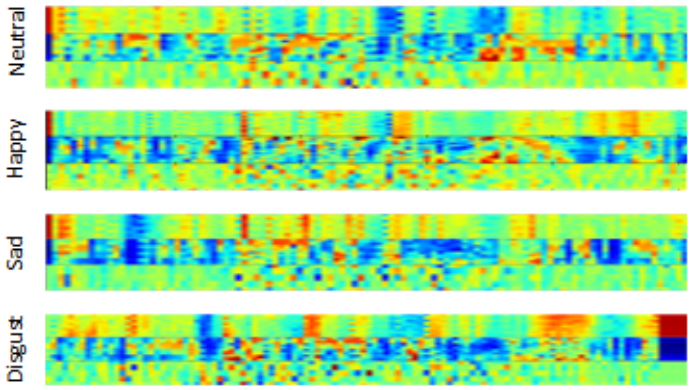


Figure 3.2: Feature sequences extracted from videos of the same actor in RAVDESS speaking the same sentence "*Kids are talking by the door*" under different emotional states. From top row: Neutral, Happy, Sad and Disgust, respectively.

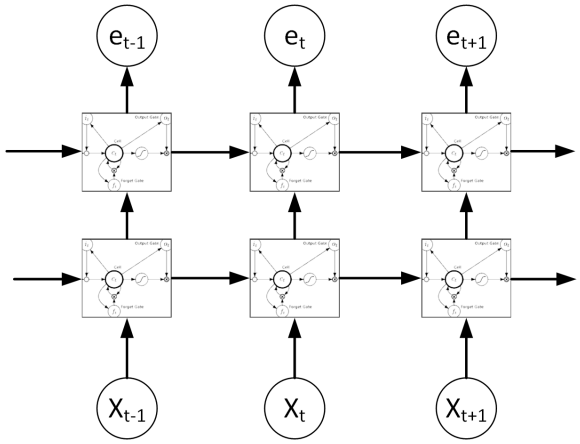


Figure 3.3: The baseline model architecture consisting of two recurrent hidden layers.

As noted in Sec. 3.2, the speech-to-facial expression mapping is formulated as a standard non-linear regression problem. Thus, parameters of the mapping function $\mathcal{F}$ can be learn by minimizing the following least-squares loss:

$$\min \frac{1}{T}\frac{1}{N}\sum_{t=1}^{T}\left\|e^t - \hat{e}^t\right\|_2^2, \tag{3.1}$$

in which $\hat{e}^t$ is the vector of expected facial action parameters at time t extracted from training video. We empirically found that applying this standard $\mathcal{L}2$ loss on the baseline model is sufficient

to make the optimization converge to a reasonably good solution. However, this is not the case with the end-to-end model, which will be discussed shortly.

3.4 End-to-end Learning for Facial Expression Synthesis

3.4.1 Audio Processing

For each video frame t in the corpus, we extract a 96ms audio frame sampled at 44.1kHz, including data of the current video frame and previous frames. Similar to the baseline, we do not consider any delay to gather future data, as they are unknown in a live streaming scenario. Instead, temporal transition will be modeled by the recurrent layer. We apply FFT with window size of 256 and hop length of 128, to recover a power spectrogram of 128 frequency bins across 32 time frames.

3.4.2 Model Architecture

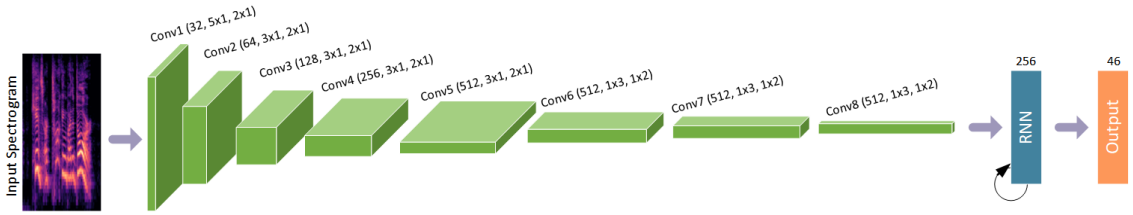


Figure 3.4: The end-to-end neural network. 1-D convolutions are applied throughout. The first five conv. layers perform convolutions along the frequency axis, and the last three conv. layers carry out convolutions along the time axis.

Our end-to-end deep neural net is illustrated in Fig. 3.4. The input to our model is raw time-frequency spectrogram of audio signal. Specifically, each spectrogram is constructed as a 2D (frequency-time) array suitable for CNN. We apply convolutions on frequency and time separately, similar to [58, 97], as this practice has been empirically shown to reduce overfitting, furthermore,

using smaller filters requires less computation, which consequently speeds up training and inference. In particular, the input spectrogram is first convolved on the frequency axis with down-sampling factor of two. Then, two-strided convolution is applied on the time axis. In this work, we report model performance where the recurrent layer is formulated as either LSTM or gated recurrent unit (GRU) [26] cells.

3.4.3 Model Learning

The end-to-end model learns not only the temporal non-linear mapping between speech audio and facial actions, but also the feature presentation specifically pertaining to expressions. We observe that, when trained using the standard $\mathcal{L}2$ loss in (3.1), the model eventually learns to focus mostly on lip-related actions and discard other important information describing emotions, because talking is the most prominent action in a speech video, all videos depict actors speaking but they do not always show a particular expression. As a result, the learned features are relevant to lip movements, but they may not be meaningful to describe other actions to depict some particular emotions. For instance, given the same speech content, the speaker may also raise their eyebrows when they feel happy, or frown to show that they are sad.

To overcome the aforementioned problems and learn correct features, we train the model by minimizing the following objective function:

$$\min \frac{1}{T}\frac{1}{N}\sum_{t=1}^{T}\left(\sum_{i=1}^{N}\left(\frac{e_i^t - \hat{e}_i^t}{\sigma_i}\right)^2 + \lambda\left\|e^t\right\|_1\right), \tag{3.2}$$

where $\hat{e}^t$ is the expected output, σ_i is the standard deviation of the i^{th} AU component extracted from training data, and λ is the trade-off weight. Essentially, the first term is the $\mathcal{L}2$ loss normalized by variance to reduce the bias towards mouth-related action units. The second term encourages parameter sparsity, as not all action units are activated simultaneously. We empirically choose $\lambda = 0.1$. Note that when trained with this loss, the baseline model actually has worse performance

than being trained with the standard $\mathcal{L}2$ loss. Hence, we only report the baseline performance when it is trained with (3.1).

3.5 Implementation Details

3.5.1 End-to-end Model Architecture

Configurations of eight convolutional layers are listed in Table 3.1. All convolutions are 1-D with stride two, which helps reduce the number of weights overall. The first five layers perform convolution along the frequency axis, and the other three layers convolve along the time axis. Each convolutional layer is followed by Batch Normalization [55] and Leaky ReLU [73] nonlinearity. The recurrent layer has 256 of either LSTM or GRU cells. In the static model, it is replaced by a fully connected layer of 1,024 units. This layer is followed by a drop out layer of probability 0.2. In bidirectional models, demonstrated in Fig. 3.5, there is one forward passing recurrent layer and one backward passing layer. To keep the number of parameters roughly the same as unidirectional models, in bidirectional models each recurrent layer consists of 128 cells. Finally, the output layer has 46 units with *sigmoid* activation.

Table 3.1: List of convolutional layers in our proposed models.

Layers	**No. Filters**	**Filter Size**	**Stride**	**Layer Size**
Conv1	32	5x1	2x1	32x64x32
Conv2	64	3x1	2x1	64x32x32
Conv3	128	3x1	2x1	128x16x32
Conv4	256	3x1	2x1	256x8x32
Conv5	512	3x1	2x1	512x4x32
Conv6	512	1x3	1x2	512x4x16
Conv7	512	1x3	1x2	512x4x8
Conv8	512	1x3	1x2	512x4x4

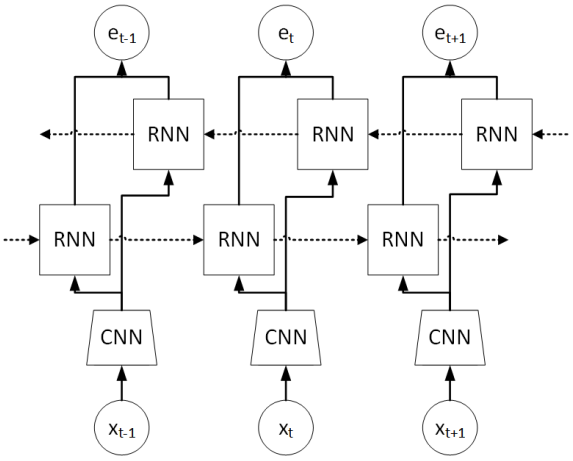


Figure 3.5: The bi-directional recurrent neural network architecture used in our experiments (*C-BiLSTM*, *C-BiGRU*). There is one forward recurrent layer, and a backward passing recurrent layer. This model passes the entire sequence at once, hence it is not suitable for real-time application.

3.5.2 Baseline Model Architecture

In our experiments, the baseline model is implemented as a two-layered LSTM-RNN, the first layer has 600 cells while there are 200 cells in the second layer. Each layer is followed by a drop out layer of probability 0.2. We found this network architecture achieves the highest performance for the baseline using engineered features.

3.5.3 Model Training

Facial action unit intensities are recovered using our 3DLGR face tracker presented in Chapter 2, running in RGB mode. We implemented our neural network models using the CNTK deep learning toolkit. Training hyperparameters are chosen as follows: minibatch size is 300, epoch size is 50,000, momentum per batch is 0.9 and weight decay is 1e-4. Learning rates are tuned differently for different models as follows.

- **End-to-end recurrent model**. Learning rate per sample is gradually decreased from 5e-3 in the first epoch to 2.5e-3 in the next two, 1e-3 in the next four, 5e-4 in the next eight, 2.5e-4

in the next 16, 1e-4 and 5e-5 each for 1,000 epochs respectively, and 2.5e-5 for the rest.

- **End-to-end static model**. Learning rate per sample is gradually decreased from 5e-3 in the first epoch to 2.5e-3 in the next two, 1.25e-3 in the next five, and 1e-4 in remaining epochs.
- **Baseline model**. Learning rate is decreased from 0.003 for the first two epochs, to 0.0015 in the next 12 epochs and 0.0003 for the rest.

Model parameters are learned by the ADAM optimizer [60] in 2,000 epochs.

3.6 Evaluation

3.6.1 Datasets

We use RAVDESS, VidTIMIT, SAVEE, GRID and GEMEP audiovisual corpora for training and evaluation. There is no overlapping between the training set and test set, as indicated in Table 3.2.

Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [71]. The database consists of 24 professional actors (12 male and 12 female, respectively) speaking and singing with various emotions. The speech set consists of eight general emotional expressions: *neutral, calm, happy, sad, angry, fearful, surprised, and disgusted*, where each video sequence is associated with one among eight affective states. Similarly, the song set, in which the actors sing short sentences, consists of six general emotional expressions: *neutral, calm, happy, sad, angry, and fearful*. We use sequences of the first 20 actors for training, data of four remaining actors for evaluation.

VidTIMIT [100]. The dataset is comprised of video and corresponding audio recordings of 43 people, reciting 10 short sentences, while keeping mostly neutral emotion throughout. Sequences of the first 40 actors are used for training, and data of the other three is used in evaluation.

Surrey Audio-Visual Expressed Emotion (SAVEE) [49]. The database consists of recordings from four male actors in six emotions and neutral. Sequences of three actors are included in

training, while data of the last actor is used for evaluation.

GRID [27]. The GRID audiovisual sentence corpus consists of recordings of 1,000 sentences (more precisely, each sentence is a sequence of unrelated words) spoken by each of 34 talkers. This database is similar to VidTIMIT, in which all actors keep neutral emotion while talking, hence we do not include GRID for training since it does not improve the facial expression modeling capability of our models. We use data of 100 randomly sampled sequences of the first 10 talkers (s1-s10) for testing.

GEneva Multimodal Emotion Portrayals (GEMEP) [12]. GEMEP is a collection of recordings featuring 10 actors speaking in French. We show that even though our models were trained on only English corpora, they can generalize well to different language, which is French in this study, thanks to the use or learning of low-level acoustic features.

Table 3.2: Data distribution for training and testing.

	Training set		Test set	
	# actors	# sequences	# actors	# sequences
RAVDESS	20	2,048	4	432
VidTIMIT	40	400	3	30
SAVEE	3	360	1	120
GRID	n/a	n/a	10	1,000
GEMEP (Fr.)	n/a	n/a	10	140

3.6.2 Experimental Protocols

The baseline model that uses engineered features is denoted as *mLSTM*. Our proposed end-to-end neural network is trained in two configurations: *C-LSTM* and *C-GRU*, in which the recurrent layer uses LSTM and GRU cells, respectively. As a baseline, we replace the recurrent layer in our proposed model with a fully connected layer and denote it as just *CNN*. This static model cannot handle smooth temporal transition, it estimates facial parameters in a frame-by-frame basis. We

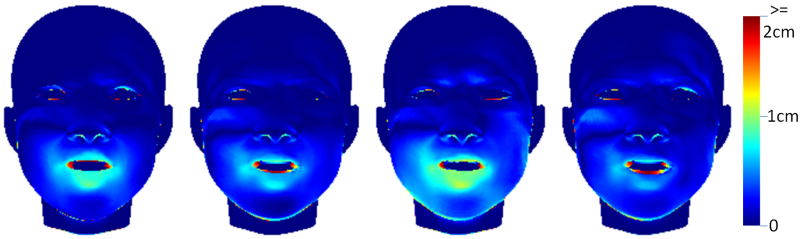


Figure 3.6: Surface errors on a sample frame, with scale indicator.

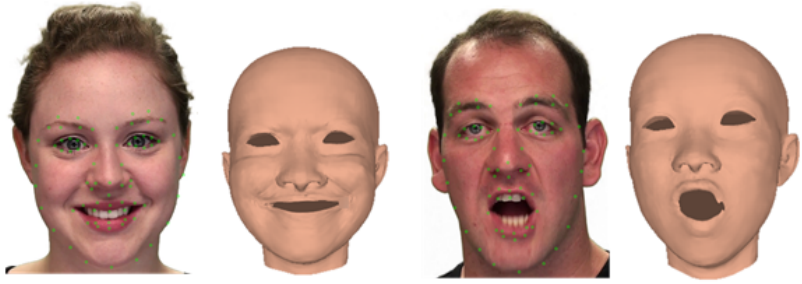


Figure 3.7: A few samples from the RAVDESS corpus, in which green dots mark 3D landmarks of the model projected to image plane. The blendshape rendered here is, however, a generic model animated given parameters estimated by the tracker. We use inner landmarks extracted from this generic model to calculate landmark RMSE.

also modify *C-LSTM* and *C-GRU* to create two bidirectional variants: *C-BiLSTM* and *C-BiGRU*, in which the recurrent layers are split into two halves, one models forward pass and one model backward pass, the total number of cells remains the same. Note that two bidirectional models do not work in real-time, they pass the entire sequence at once. More details about bidirectional model variants can be found in the Appendix.

We measure performance of these models on *three* metrics: in addition to RMSE of 3D landmarks (lRMSE) and MSE of facial action parameters (pMSE) with respect to ground truths recovered by the visual tracker [84], we also report *temporal smoothness*:

$$\frac{1}{N} \left\| \left(e^{t+1} - e^{t}\right) - \left(\hat{e}^{t+1} - \hat{e}^{t}\right) \right\|_2^2, \tag{3.3}$$

which shows how smooth frame transition in output sequences is, compared to ground truths. Landmark errors are calculated as real-world distances in millimeter from inner landmarks (shown in

Fig. 3.7) on the reconstructed 3D face shape, to those of the ground truth 3D shape (of a generic identity, to avoid inaccuracy in identity recovery). Additionally, we also visualize surface reconstruction error as the absolute difference between the ground truth surface depth map and one created from output AU parameters. Sample error heat maps and scale indicator are shown in Fig. 3.6. Nevertheless, these error metrics do not truly reflect performance of our deep models on 3D face reconstruction quality, because facial expression may not always relate to speech. For example, the speaker may open her mouth or raise eyebrows but does not utter a sound.

3.6.3 Evaluation Results

Table 3.3: Error metrics of all models on the test set. Best results are marked in bold.

	mLSTM	**CNN**	**C-LSTM**	**C-GRU**	**C-BiLSTM**	**C-BiGRU**
MSE on AU coefficients ($\times$ 1e-2)						
RAVDESS	7.179	6.53	7.225	6.46	6.55	**6.382**
VidTIMIT	8.271	7.646	8.059	**7.135**	7.426	**7.135**
SAVEE	8.316	7.817	9.591	**7.717**	8.155	8.02
GRID	8.143	8.247	8.216	7.028	7.342	**6.825**
GEMEP	10.369	10.047	10.315	**9.481**	10.247	9.636
RMSE (unit: mm) on 3D landmarks						
RAVDESS	1.067	1.038	1.049	1.022	1.024	**1.013**
VidTIMIT	0.974	0.993	0.983	**0.948**	0.969	0.963
SAVEE	**1.2**	1.228	1.263	**1.2**	1.222	1.24
GRID	1.045	1.076	1.08	1.015	1.033	**1.011**
GEMEP	1.265	1.219	1.225	**1.201**	1.231	1.21
Temporal smoothness ($\times$ 1e-2)						
RAVDESS	**0.284**	1.94	0.333	0.328	0.325	0.323
VidTIMIT	**0.575**	2.888	0.631	0.63	0.637	0.64
SAVEE	**0.759**	2.223	0.788	0.787	0.81	0.804
GRID	**0.552**	2.243	0.592	0.58	0.592	0.597
GEMEP	**0.606**	2.549	0.658	0.642	0.661	0.67

We separate evaluation on RAVDESS, VidTIMIT, SAVEE from that of GRID and GEMEP in this section. While RAVDESS-VidTIMIT-SAVEE test sequences were captured under the same

conditions with those in the training set, in which all actors speak a few sentences in different ways, GRID sequences were captured from another group of actors under a different setting, including a distinct set of sentences that are not used in other corpora. More challengingly, the language recorded in GEMEP is totally different (French vs. English). This will help us ascertain whether our proposed models are able to generalize beyond those limited observations they were originally trained with.

RAVDESS-VidTIMIT-SAVEE Test Set

Table 3.3 shows the aforementioned error metrics of all models on the test set. Based on parameter MSE and landmark RMSE, both *C-GRU* and *C-BiGRU* outperform their LSTM-based counterparts, as well as the static model and the baseline. Furthermore, *C-BiGRU*, which holds the advantage of knowing all past and future events, only performs slightly better than the unidirectional model *C-GRU*.

Specifically, on RAVDESS, *C-BiGRU* is slightly better than *C-GRU* with 1.2% lower pMSE and 0.9% lower lRMSE. In terms of pMSE, *C-GRU* outperforms *C-BiLSTM* by 1.3%, *C-LSTM* by 10.6%, *CNN* by 1.1%, and *mLSTM* by 10%. *C-GRU* also achieves lower landmark errors compared to other models, except *C-BiGRU*. However, it is observed that errors of *CNN* are comparable or slightly higher than those of *C-GRU* and *C-BiGRU* across different emotion categories in RAVDESS as demonstrated in Table 3.4 and 3.5. This can be explained by the inherent characteristics of RAVDESS speech sequences. RAVDESS actors manifest spontaneous and varying facial expressions while speaking, thus, the static model may have a slight edge in estimating those sudden expression changes, especially in the cases of Actor 23 and Actor 24. *CNN* scores marginally lower errors for these two actors, while GRU-based models have better estimates for Actor 21 and 22.

On the other hand, *C-GRU* has similar performance to *C-BiGRU*, and outperforms *CNN* by

Table 3.4: MSE ($\times$ 1e-2) of expression blending weights, organized by categories on RAVDESS (average errors grouped by emotion or actor). Best results are marked in bold.

	Neutral	Calm	Happy	Sad	Angry	Fear.	Disgu.	Surpr.	Act.21	Act.22	Act.23	Act.24
mLSTM	7.001	7.059	7.378	7.339	6.738	6.928	7.965	7.583	5.049	6.93	7.105	9.576
CNN	6.505	6.356	7.026	6.607	6.169	6.281	**6.734**	6.87	4.457	6.191	**6.3**	**9.109**
C-LSTM	7.045	7.16	7.876	7.386	6.702	6.793	7.367	7.777	4.755	6.611	7.399	10.079
C-GRU	6.506	6.427	7.255	6.357	5.746	6.242	6.79	6.76	**3.907**	6.171	6.438	9.262
C-BiLSTM	6.842	6.536	7.293	6.85	**5.631**	6.108	6.823	**6.666**	4.289	5.646	6.643	9.558
C-BiGRU	**6.482**	**6.317**	**6.909**	**6.334**	5.83	**6.029**	6.876	6.927	3.909	**5.409**	6.662	9.492

Table 3.5: RMSE of 3D landmarks in millimeter on RAVDESS, organized by categories. Best results are marked in bold.

	Neutral	Calm	Happy	Sad	Angry	Fear.	Disgu.	Surpr.	Act.21	Act.22	Act.23	Act.24
mLSTM	1.072	1.052	1.111	1.047	1.073	1.042	1.053	1.1	1.055	0.997	1.055	1.153
CNN	1.048	1.022	1.092	1.014	1.046	1.015	1.028	1.046	1.029	0.991	0.99	1.134
C-LSTM	1.054	1.04	1.082	1.033	1.055	1.019	1.048	1.078	1.048	0.942	1.043	1.149
C-GRU	**1.029**	1.017	1.078	0.989	1.016	1.005	1.019	1.033	**0.98**	0.973	**0.986**	1.136
C-BiLSTM	1.039	1.021	1.076	1.001	**1.014**	1.007	1.015	**1.014**	1.01	0.952	1.011	1.113
C-BiGRU	1.035	**1.006**	**1.062**	**0.98**	**1.014**	**0.985**	**1.013**	1.028	0.993	**0.928**	1.007	**1.112**

6.7% on VidTIMIT. This is because in VIDTIMIT, each actor maintains almost uniform expression throughout the sequence, mostly only the mouth region is deformed, i.e. speaking. The facial deformation dynamics is smooth and stable, hence, recurrent models are able to estimate temporal changes of facial action intensities effectively. It is also worth noticing that knowing future events in *C-BiLSTM* improves from the performance of *C-LSTM* by 9%. Overall, our end-to-end models, except *C-LSTM*, outperform the baseline that uses engineered features, signifying the advantage of learning acoustic features directly within the mapping function $\mathcal{F}$.

C-GRU outperforms all other models on SAVEE: lMSE is 3.8% lower than *C-BiGRU*, 1.3% lower than *CNN* and 7.2% lower than *mLSTM*. However, in general test errors on SAVEE are higher than those on RAVDESS and VidTIMIT. This may be caused by the slightly unnatural acting of the test speaker, "KL", unlike speakers in RAVDESS and VidTIMIT. Thus AU parameters estimated from audio do not match ground truths recovered from video.

However, the static model *CNN* does not handle temporal smoothness, thus sequential transitions in its generated animation are often jarring, unlike recurrent models. Indeed, this is demonstrated through the smoothness metric in Table 3.3. Temporal smoothness measures of recurrent models are roughly similar, and are an order of magnitude better than that of *CNN*. It means that all recurrent models generate satiny animated sequences. Particularly, *mLSTM* has the best temporal smoothness measures across all datasets, it means that this model estimates frame-by-frame facial expression transition closest to ground truth, but it often under- or over-estimates frame-wise facial actions, thus its pMSE is higher. These results show that our proposed *C-GRU* model achieves both accuracy and temporal smoothness of predicted facial action sequences, despite being oblivious to future events in both training and testing. It also significantly outperforms the baseline model using engineered features, *mLSTM*.

Fig. 3.8 provides further insight on how each model estimates the most prominent facial action parameters from speech. GRU-based models and *CNN* demonstrate superior performance across

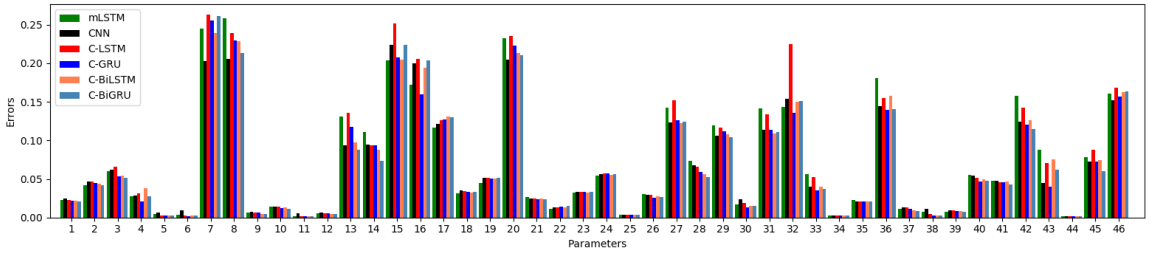


Figure 3.8: Plotting RAVDESS-VidTIMIT-SAVEE average MSE of individual action unit parameters. In general *C-GRU*, *C-BiGRU* and *CNN* have similar performance across different AUs in terms of mean square error, each model achieves the smallest errors for some AUs. LSTM-based models in general have higher errors than the GRU-based counterparts. End-to-end models outperform *mLSTM* in most cases, especially on major AUs: 8, 10, 13-18, 36 and 42.

different action units. Nonetheless, these error metrics are calculated from compressed information (landmarks and parameters), they do not fully indicate the quality of shape reconstruction. Fig. 3.9, 3.10 show errors when comparing reconstructed surfaces by all models to the shapes estimated by the visual tracker. It is observed that *C-GRU* and *C-BiGRU* often have better 3D face reconstruction. In Fig. 3.9(a-f,h,i), and all speech-driven models reproduce reasonably precise facial expressions, indicated by low surface errors. In fact, in Fig. 3.9(e,f), speech-driven models, especially *C-GRU*, can actually estimate better lip deformations than the visual tracker.

In Fig. 3.9g, the visual tracker predicts that AU20, "chin sticking out", is activated. However, speech-based models are unable to predict this facial action, thus errors in the chin region are a little higher. Fig. 3.9(j-l) demonstrate error estimates on three frames of Actor 24. In her video, this actor moves her head fast often with large poses, thus the visual tracker was unable to predict her identity and facial expressions accurately and it recovered noisy estimates, demonstrated in her ground truth 3D shapes in the top row. Interestingly, speech-based models can estimate her facial actions rather well, especially *C-GRU*, *C-BiLSTM* and *C-BiGRU*, e.g. shown in Fig. 3.9(j,l). Hence, even though their errors compared to the visual tracker are higher and skewed the average error on the test set (shown quantitatively in the column of Actor 24 in Table 3.4 and 3.5), the speech-driven models actually predict more accurate facial expressions in this case. It suggests the

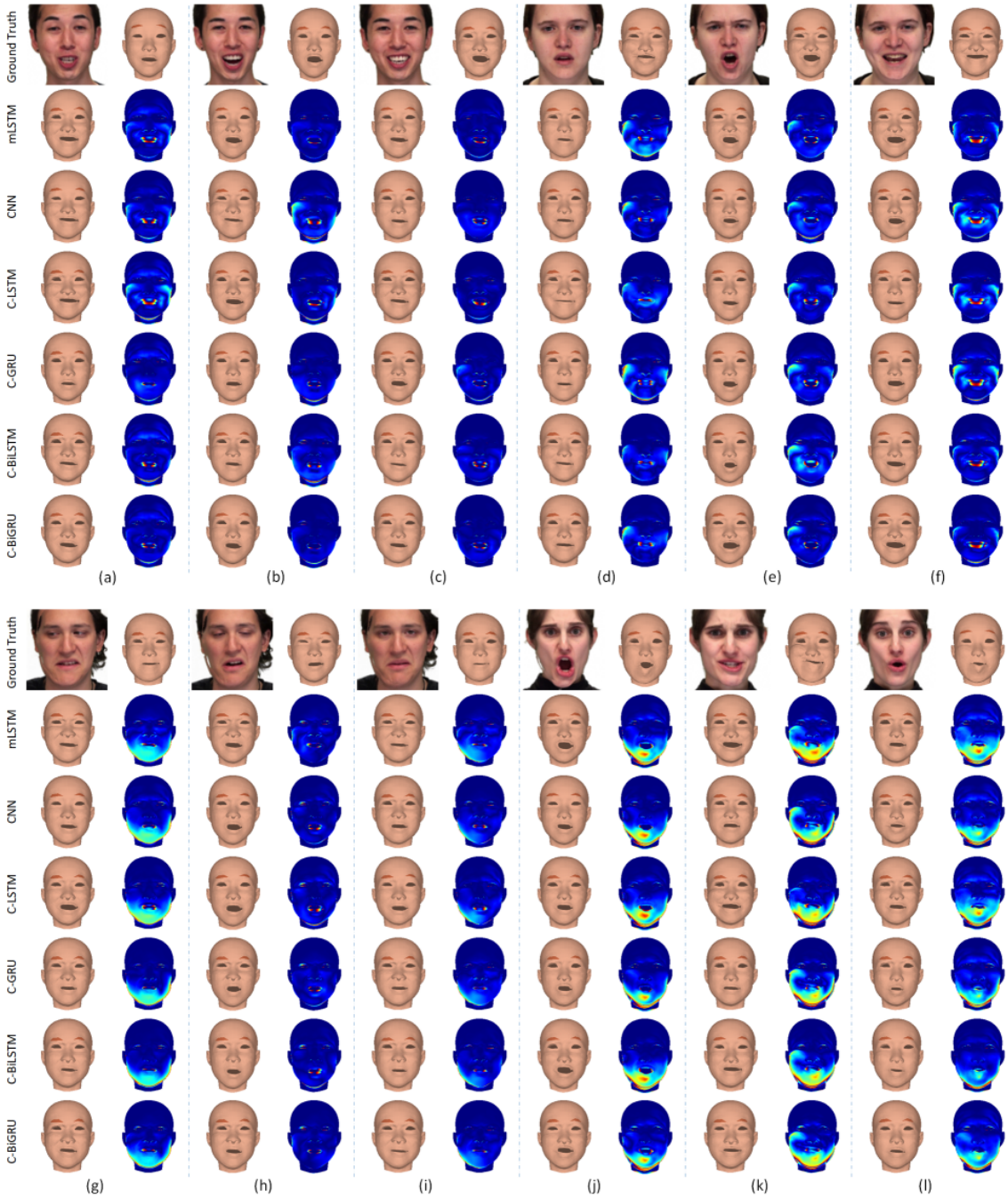


Figure 3.9: Reconstruction results from the RAVDESS corpus. Each sub-figure (a-l) demonstrates the original facial texture (top-left), the shape reconstructed with AU parameters recovered by the visual tracker (top-right), four shapes generated with parameters estimated by four models (bottom-left) and their corresponding surface error maps (bottom-right). The identities of actors in sub-figures are as follows: (a-c): Actor 21 "Happy"; (d-f): Actor 22 "Angry"; (g-i): Actor 23 "Disgusted"; (j-l): Actor 24 "Surprised". Facial textures are frontalized, and frames with large pose may contain artifacts in the frontalized patches (e.g: Actor 23). Notice that regions on the heat maps marked with "red" pixels typically indicate a hole, e.g. a shape has mouth opening while the other one does not. In (j-l), the "ground truth" estimates of Actor 24 are noisy, hence errors when comparing output parameters to the ground truth are high.

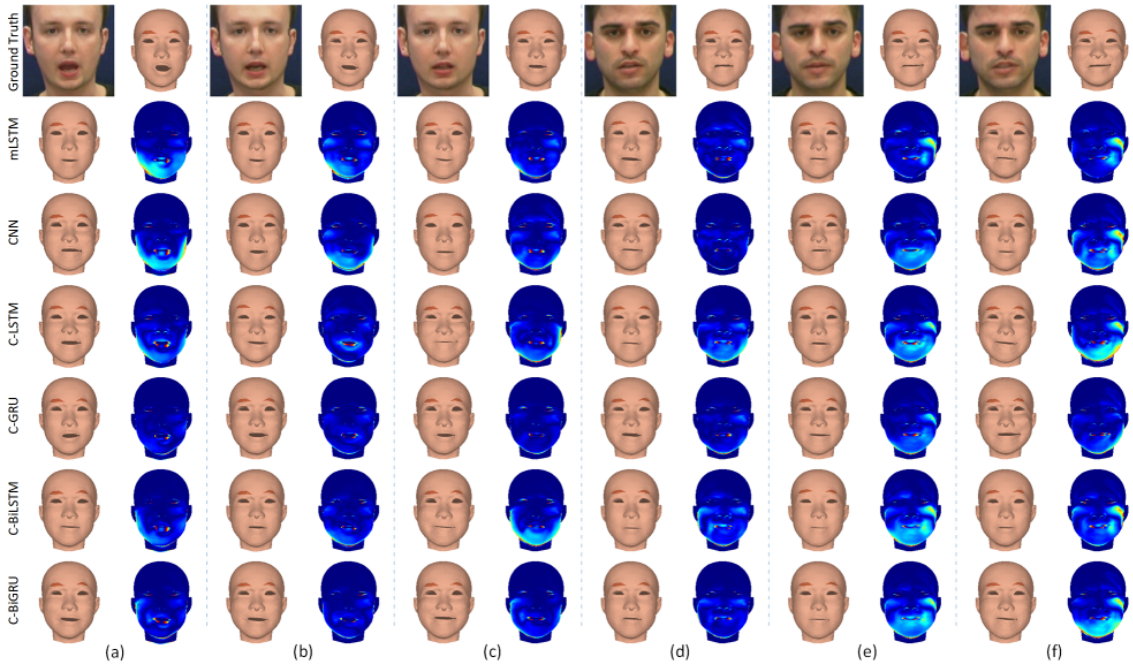


Figure 3.10: Reconstruction results for two sequences of two actors in the VidTIMIT test set. (a-c): actor "mwbt0"; (d-f): actor "mtmr0". It is observable that *C-GRU* has the best 3D shape reconstruction among all models indicated by lower surface errors, supporting the lower error numbers shown in Table 3.3.

potential of combining visual and acoustic information to improve face tracking in general, as the audio signal can provide complementary cue when visual data is missing (i.e. face is occluded) or noisy. Fig. 3.10 illustrates that *C-GRU* performs the most accurately on VidTIMIT, as it is reflected by the testing errors in Table 3.3.

GRID Corpus

As shown in Table 3.3, pMSEs of *C-GRU*, *C-BiLSTM* and *C-BiGRU* on GRID are lower than values of the same errors on VidTIMIT. These results are important, they show that our models can generalize well to unseen speeches. Particularly, *C-BiGRU* achieves 2.9% lower error than *C-GRU* as expected. *C-GRU* outperforms *CNN* and *mLSTM* by 14.6% and 13.7%, respectively. Table 3.6

Table 3.6: MSE ($\times$ 1e-2) of AU parameters on 10 talkers (s1-s10) in GRID.

	Talker 1	Talker 2	Talker 3	Talker 4	Talker 5	Talker 6	Talker 7	Talker 8	Talker 9	Talker 10
mLSTM	5.623	6.062	5.484	16.246	4.75	6.999	7.892	9.242	12.286	7.538
CNN	5.293	5.894	5.249	16.239	4.596	6.74	8.826	9.297	12.641	7.665
C-LSTM	5.039	5.569	5.397	15.088	4.261	7.092	8.112	10.798	13.122	7.74
C-GRU	3.816	**3.904**	4.132	13.188	**4.012**	5.556	**6.324**	9.338	13.761	**6.317**
C-BiLSTM	**3.764**	5.59	4.323	**12.991**	4.879	5.84	7.794	8.941	12.192	7.094
C-BiGRU	3.846	4.524	**3.384**	13.182	4.597	**5.268**	7.414	**8.265**	**11.372**	6.378

Table 3.7: 3D landmark RMSE on 10 talkers in GRID.

	Talker 1	Talker 2	Talker 3	Talker 4	Talker 5	Talker 6	Talker 7	Talker 8	Talker 9	Talker 10
mLSTM	**0.919**	1.036	0.95	1.364	0.9	1.006	1.073	1.029	1.074	1.055
CNN	1.025	0.999	0.977	1.396	0.925	1.009	1.149	1.005	**1.067**	1.13
C-LSTM	1.033	1.047	0.991	1.385	0.913	1.032	1.106	1.01	**1.067**	1.141
C-GRU	0.937	**0.917**	0.898	**1.275**	0.864	0.971	**1.002**	1.015	1.193	**1**
C-BiLSTM	0.962	0.965	0.92	1.29	0.85	0.976	1.091	0.969	1.124	1.103
C-BiGRU	0.959	0.942	**0.862**	1.3	**0.848**	**0.962**	1.062	**0.951**	1.092	1.052

Figure 3.11: Synthetic facial expression dynamics of two sequences of two talkers, "s1" and "s7" in GRID. Notice the dynamics of lip movements generated by all models: *mLSTM* and *CNN* always activate mouth opening AUs, that is undesirable. *C-LSTM* tends to predict lip corner raiser. *C-GRU* produces the most realistic facial action dynamics among six models.

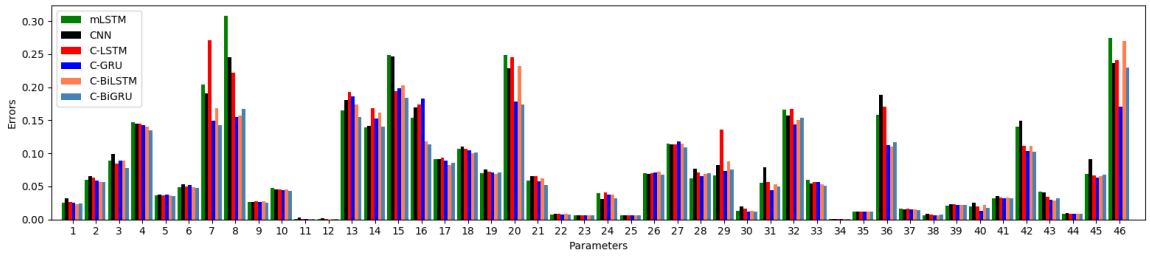


Figure 3.12: Plotting GRID average MSE of individual action unit parameters. *C-GRU* and *C-BiGRU* have superior performance on GRID, especially for AUs: 7, 8, 15, 20, 36 and 42. (AU36 controls talking action mostly).

presents pMSEs organized by talker. Errors on Talker 4, 8 and 9 are specifically higher than the rest, while errors on other speakers are roughly similar to errors on RAVDESS in Table 3.4. Higher

errors on three speakers may be caused by the mismatch between the visual expression and the actual speech, in other words, unnatural acting. This table also shows that *C-GRU* and *C-BiGRU* outperform other models in most cases, especially *C-GRU* achieves the best performance on Talker 2, 7 and 10. Table 3.7 also shares similar observations in landmark RMSE.

Fig. 3.12 illustrates individual AU errors, similar to what is presented in Fig. 3.8. *C-GRU* and *C-BiGRU* have similar error numbers across different AUs. However, unlike the experiment on RAVDESS-VidTIMIT-SAVEE, *CNN* performs much worse than those two models. *mLSTM* and *C-LSTM* also have much higher errors than two GRU-based models. This may imply that models using Gated Recurrent Unit learn more meaningful and relevant feature representation, thus they can generalize better to unseen data. However, more thorough study is required in order to support this hypothesis.

As stated earlier, these error metrics may not fully reflect the face synthesis quality of our models, hence we also visualize and inspect the animated sequences of subjects in GRID. Two sequence of "s1" and "s7" are demonstrated in Fig. 3.11. In these samples, *CNN* and *mLSTM* often generate incorrect lip deformations, e.g. the fourth and sixth columns of Fig. 3.12a, or the second column in Fig. 3.11b. This may explain why error of these models on AU36 are high. Interestingly, in the last column of Fig. 3.11b, speech-driven models predict correct mouth-opening action while the face tracker failed at this frame, thus it further shows the usefulness of modeling facial actions from speech audio, which can complement visual tracking. Overall, it seems that *C-GRU* generate the most accurate lip motions among six models on GRID.

GEMEP Corpus

Testing on the GEMEP database really pushes our models beyond their limit, since all actors speak French which has different speech patterns compared to English that all models were trained upon. As expected, Table 3.3 shows that both pMSE and lRMSE of all models on GEMEP are

Table 3.8: MSE ($\times$ 1e-2) of AU parameters on all 10 actors of GEMEP.

	Actor 1	Actor 2	Actor 3	Actor 4	Actor 5	Actor 6	Actor 7	Actor 8	Actor 9	Actor 10
mLSTM	8.753	7.367	8.721	7.123	**8.852**	12.076	12.564	8.693	14.752	13.228
CNN	9.164	6.49	8.405	7.259	9.602	11.594	11.342	7.872	13.535	13.64
C-LSTM	9.353	6.289	9.021	7.345	9.719	12.378	12.019	8.271	13.785	13.209
C-GRU	9.138	**5.596**	**7.987**	**6.214**	9.201	**10.976**	11.259	7.889	**12.645**	**12.329**
C-BiLSTM	9.504	6.784	9.002	6.721	9.334	12.696	11.463	8.307	14.068	12.872
C-BiGRU	**8.628**	6.494	8.213	6.864	9.492	11.005	**10.87**	**7.805**	12.935	12.525

Table 3.9: 3D landmark RMSE on 10 actors in GEMEP.

	Actor 1	Actor 2	Actor 3	Actor 4	Actor 5	Actor 6	Actor 7	Actor 8	Actor 9	Actor 10
mLSTM	1.156	1.286	1.135	1.182	1.202	1.335	1.231	1.135	1.505	1.384
CNN	1.122	1.183	1.1	1.175	1.199	1.28	1.191	1.082	1.405	1.374
C-LSTM	1.118	1.189	1.13	1.158	**1.184**	1.267	1.212	1.128	1.415	1.364
C-GRU	1.13	**1.129**	**1.078**	**1.122**	1.19	1.262	1.169	1.108	**1.376**	1.367
C-BiLSTM	1.141	1.189	1.115	1.175	1.188	1.298	1.209	1.109	1.432	1.372
C-BiGRU	**1.116**	1.194	1.09	1.174	1.185	**1.255**	**1.167**	**1.076**	1.407	**1.359**

Figure 3.13: Synthetic facial expression dynamics of two sequences of two actors, “Actor 1” and “Actor 7” in GEMEP.

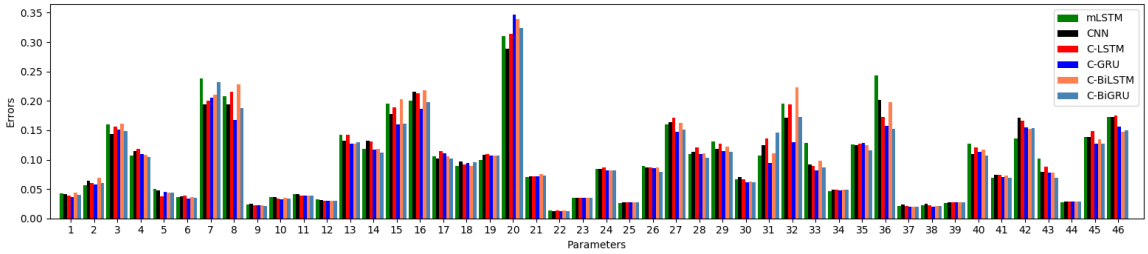


Figure 3.14: Plotting GEMEP average MSE of individual action unit parameters. Overall, all individual AU errors are uniformly higher than corresponding errors on other datasets.

clearly higher than those errors on other English speech datasets. Another possible cause of high error is that GEMEP actors tend to over-act, which makes it difficult to predict correct facial expressions from speech alone.

We also organize errors by actor in Table 3.8 and 3.9 in the same way as previous experiments. Table 3.8 shows that pMSEs of Actor 6, 7, 9, 10 are significantly higher than errors on other actors. In their videos, they display very strong expressions, which are not reflected in their speech recordings. These individual errors cause the average GEMEP error values in Table 3.3 higher than average errors of other datasets. This is further illustrated in Fig. 3.14, all individual AU errors on GEMEP are higher than corresponding errors on other datasets. GRU-based models often outperform other models in most cases, and in general *C-GRU* achieves the lowest pMSE as well as the best landmark reconstruction.

Fig. 3.13 demonstrates two sequences of "Actor 1" and "Actor 7" in GEMEP. *C-GRU* reconstructs the best facial expression dynamics, similar to ground truth 3D face shapes. *mLSTM* generates inaccurate lip motions, which probably indicate the limitation of handcrafted acoustic features in generalizing to different languages and speech patterns. The static model also does not perform as well as recurrent models, e.g. second column in Fig. 3.13a, third and fifth columns in Fig. 3.13b. Perhaps learning feature in combination with modeling temporal speech context holds the advantage in learning better feature representation that generalize well to different, unseen speeches.

Inference Speed

Testing on a laptop with a Quadro K1000M GPU, and measuring running time after data preprocessing, we record that all models implemented using CNTK take approximately 5ms to process one frame, They would run even faster on modern hardware. For the baseline model, data preprocessing includes feature extraction which requires non-trivial amount of computation, whereas in the end-to-end models, feature extraction is already a part of inference. Thus our proposed end-to-end models only need minimal audio processing, and the data processing pipeline is favorably simplified. Overall, they will perform faster, and better, than the baseline *mLSTM*.

3.7 Related Work

3.7.1 Talking Head Synthesis

"Talking head", is a research topic where an avatar is animated to imitate human talking. Various approaches have been developed to synthesize a face model driven by either speech audio [43, 120, 98] or transcripts [115, 31]. Essentially, every talking head animation technique develops a mapping from an input speech to visual features, and can be formulated as a classification or regression task. Classification approaches usually identify phonetic unit (phonemes) from speech and map to visual units (visemes) based on specific rules, and animation is generated by morphing these key images. On the other hand, regression approaches can directly generate visual parameters and their trajectories from input features. Early research on talking head used Hidden Markov Models (HMMs) with some successes [116, 117], despite certain limitations of HMM framework such as oversmoothing trajectory.

In recent years, deep neural networks have been successfully applied to speech synthesis [89, 130] and facial animation [37, 131, 43] with superior performance. This is because deep neural networks (DNN) are able to learn the correlation of high-dimensional input data, and, in case of recurrent neural network (RNN), long-term relation, as well as the highly non-linear mapping between input and output features. Taylor et al. [107] propose a system using DNN to estimate active appearance model (AAM) coefficients from input phonemes, which can be generalized well to different speeches and languages, and face shapes can be retargeted to drive 3D face models. Suwajanakorn et al. [105] use long short-term memory (LSTM) RNN [52] to predict 2D lip landmarks from input acoustic features, which are used to synthesize lip movements. Fan et al. [43] use both acoustic and text features to estimate active appearance model AAM coefficients of the mouth area, which then be grafted onto an actual image to produce a photo-realistic talking head. Karras et al. [58] propose a deep convolutional neural network (CNN) that jointly takes audio autocorrelation coefficients and emotional state to output an entire 3D face shape.

In terms of the underlying face model, these approaches can be categorized into image-based [14, 31, 42, 116, 120, 43] and model-based [11, 9, 99, 119, 37, 23] approaches. Image-based approaches compose photo-realistic output by concatenating short clips, or stitch different regions from a sample database together. However, their performance and quality are limited by the amount of samples in the database, thus it is difficult to generalize to a large corpus of speeches, which would require a tremendous amount of image samples to cover all possible facial appearances. In contrast, although lacking in photo-realism, model-based approaches enjoy the flexibility of a deformable model, which is controlled by only a set of parameters, and more straightforward modeling. In our earlier work. [81], we proposed a mapping from acoustic features to blending weights of a blendshape model [20], in addition to head pose. This face model allows emotional representation that can be inferred from speech, without explicitly defining the emotion as input, or artificially adding emotion to the face model in postprocessing. Our approach also enjoys the flexibility of blendshape model in 3D face reconstruction from speech. This work reuses the model in [81], modified to target only facial expressions.

3.7.2 CNN-based Speech Modeling

Convolutional neural networks [63] have achieved great successes in many vision tasks e.g. image classification or segmentation. Their efficient filter design allows deeper network, enables learning features from data directly while being robust to noise and small shift, thus usually having better performance than prior modeling techniques. In recent years, CNNs have been also employed in speech recognition tasks, which directly model the raw waveforms by taking advantage of the locality and translation invariance in time [111, 79, 53] and frequency domain [35, 2, 1, 95, 96, 97]. In this work, we also employ convolutions in the time-frequency domain, and formulate an end-to-end deep neural network that directly maps input spectrogram to blendshape weights.

3.8 Summary

In this chapter we introduced a deep learning framework for real-time speech-driven 3D facial animation from audio recording, which is realized by different recurrent neural network models. Our proposed deep neural networks learn a mapping from audio signal to the temporally varying context of the speech, as well as emotional states of the speaker represented implicitly via blending weights of a 3D face model. Our baseline model utilizes handcrafted acoustic features to infer the facial action parameters. However, these inherently lossy features may exclude important information pertaining to facial expressions carried within the speech. Hence, we propose to learn more meaningful and relevant feature representation directly with convolutional net, which leads to significant improvement in predicting facial actions.

Experiments on diverse and challenging datasets demonstrate that our models could estimate lip movements together with emotional state intensities of the speaker reasonably well from just her speech. Furthermore, our experiments show that our models can generalize to unseen speech patterns and language in GRID and GEMEP audiovisual corpora, thanks to the learning of low-level acoustic feature representation. Moreover, we observe that using Gate Recurrent Unit leads to better performance and generalization for this task. Our end-to-end unidirectional model, *C-GRU*, outperforms other models, and is comparable to the bidirectional variant, despite not knowing any future events.

Chapter 4

Learning Facial Performance Synthesis

In this chapter we present *Generative Adversarial Talking Head* (*GATH*), a novel deep generative neural network that enables fully automatic facial expression synthesis of an arbitrary portrait with continuous action unit (AU) coefficients. Specifically, our model directly manipulates image pixels to make the unseen subject in the still photo express various emotions controlled by values of facial AU coefficients, while maintaining her personal characteristics, such as facial geometry, skin color and hair style, as well as the original surrounding background.

In contrast to prior work, *GATH* is purely data-driven and it requires neither a statistical face model nor image processing tricks to enact facial deformations. Additionally, our model is trained from unpaired data, where the input image, with its auxiliary identity label taken from abundance of still photos in the wild, and the target frame are from different persons. In order to effectively learn such model, we propose a novel weakly supervised adversarial learning framework that consists of a generator, a discriminator, a classifier and an action unit estimator.

Our work gives rise to *template-and-target-free expression editing*, where still faces can be effortlessly animated with arbitrary AU coefficients provided by the user.

This chapter contains materials from our paper [87].

4.1 Introduction

Human faces convey a large range of semantic meaning through facial expressions, which reflect both actions e.g. talking, eye-blinking, and emotional states such as happy (smiling), sad (frowning) or surprised (raising eyebrows). Over the years, much research has been dedicated to the task of facial expression editing, in order to transfer the semantic expression from a target to a source face, with impressive results [5, 108, 32, 44, 78]. In general, these state-of-the-art techniques assume that a pair of source-target images is available, and there exists a pair of matching 2D or 3D facial meshes in both images for texture warping and rendering. Additionally, recent work by Thies et al. [108] and Cao et al. [21] require a set of source images in order to learn a statistical representation, that can be used to create a source instance at runtime. The above requirement limits the application of these techniques to certain settings, where source data is abundant. In [5, 78, 126], the authors propose to directly transfer expressions from the target image to the source face, forgoing the need of prior statistics of the source subject. However, there are situations in which the target face to drive facial deformation of the source does not exist, instead, facial expression can be inferred from other input modalities, such as speech [81, 86, 43], or explicitly specified by user as vector of facial action unit (AU) intensities [38].

In this work, we are interested in mid-level facial expression manipulation by directly animating a human portrait given only AU coefficients, thereby enabling a whole new level of flexibility to the facial expression editing task. Particularly, our proposed GATH model is able to modify a frontal face portrait of arbitrary identity and expression at pixel level, hallucinating a novel facial image whose expressiveness mimics that of a real face that has similar AU attributes. In other words, our model learns to extract identity features to preserve individual characteristic of the portrait, facial enactment to animate the portrait according to values of AU coefficients and texture mapping, all in an end-to-end deep neural network.

(a) source and target are of the same subject

(b) source and target are from different persons

Figure 4.1: Some samples generated by our proposed GATH model Each triplet consists of the source, the target and the synthesis. Note that our model only knows the source image and a vector of action unit coefficients that resemble the target. Could the reader tell apart which image is the source, target and synthesis? Hint: start with (b) first.

Learning identity features requires a large number of training images from thousands of subjects, which are readily available in various public datasets. On the other hand, the amount of

publicly available emotional videos such as [71], from which we could collect a wide range of AU coefficients, is rather limited. A deep net trained on such a small number of subjects would not generalize well to unseen identity. GANimation [88] was proposed recently to learn facial expression transformation in an unsupervised framework. However, because the target expression is randomly sampled in their work, it may turn out to be unnatural. To address these shortcomings, we propose to train the deep net with separate source and target sets, i.e. the animated facial image of subject A in the source set does not have an exact matching target image, but there exists an image of subject B in the target set that has similar expression to the synthesized image of A, and their expressiveness similarity is measured by an auxiliary function. Inspired by recent advances in image synthesis with adversarial learning [48, 90], we jointly train the deep face generator with a discriminator in a minimax game, in which the generator gradually improves the quality of its synthesis to try to fool the discriminator in believing that its output is from the real facial image distribution. Furthermore, taking advantage of the availability of subject class labels in the source set, we jointly train a classifier to recognize the subject label of the generated output, therefore encouraging the generator to correctly learn identity features, and producing better synthesis of the input subject.

Our main contributions are as follows:

- Generative Adversarial Talking Head, a deep model that can generate realistic expressive facial animation from arbitrary portraits and AU coefficients. The model is effectively trained in an adversarial learning framework including a generator, a discriminator and a classifier, where the discriminator and the classifier supervise the quality of synthesized images, while the generator learns facial deformations from separate source and target image sets, and is able to disentangle latent identity and expression code from the source image.

- An action unit estimator (AUE) network, whose hidden features are used as an expressiveness similarity measure between the synthetic output and its unpaired target facial image in order

to guide the generator to synthesize images with correct expression.

- Extensive evaluations and applications to demonstrate the effectiveness and flexibility of our proposed model in animating various human portraits from video-driven and user-defined AU coefficients.

4.2 Overview

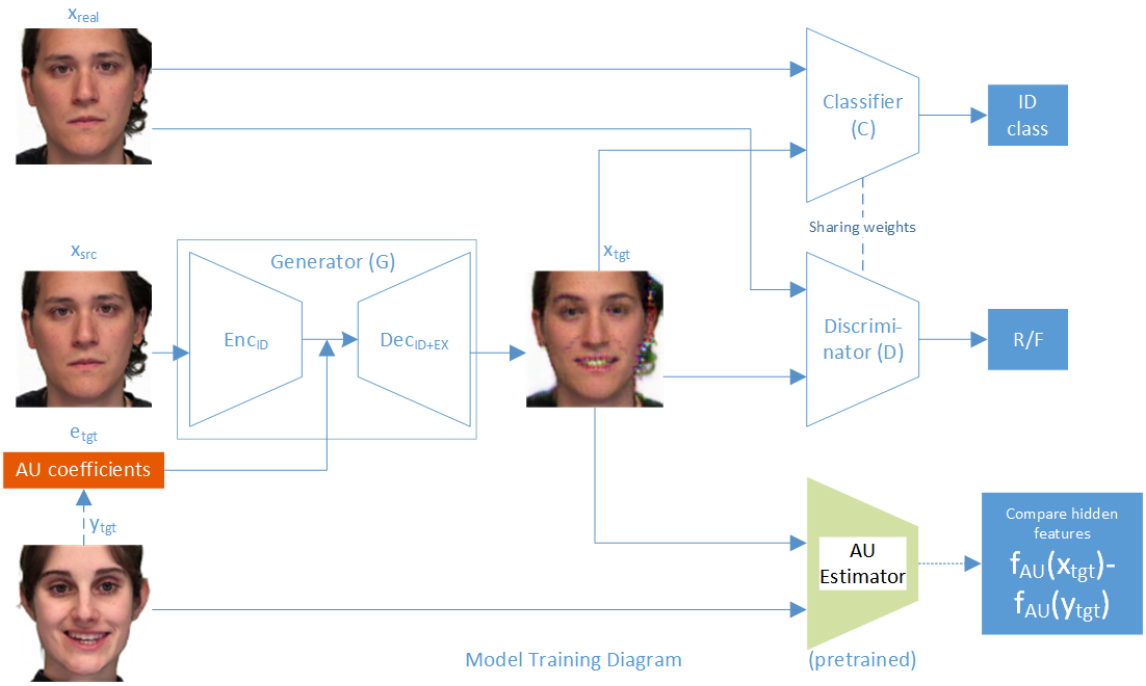


Figure 4.2: The proposed GATH learning framework. Except for the AU Estimator that is pre-trained, other networks including the generator, the discriminator and the classifier are jointly trained. The discriminator and the classifier share hidden layer weights. The generator only knows AU coefficients e_{tgt} extracted from target frame y_{tgt}. The generator G learns to produce output x_{tgt} from input image x_{src}, such that the synthesized face has similar facial expression as the target frame y_{tgt}. x and y are different subjects.

We first denote the following notations that will be used throughout the paper: G = generator; f_{G_en} = encoder subnetwork of G; f_{G_de}: decoder subnetwork of G; D = discriminator; $\mathcal{C}$ =

classifier; $\mathcal{E}$ = AU Estimator; f_{au} = a function that maps image to a latent facial expression space; x_{src} = source portrait to be transformed; x_{tgt} = image synthesized by the generator; x_{re} = real image used to train D and $\mathcal{C}$; c = the class label associated with x_{re}; y_{tgt} = target image; e_{tgt} = continuous AU coefficient vector corresponding to y_{tgt}. x_{src} and x_{re} are sampled from the same training source set, and are not necessarily the same. y_{tgt} is sampled from the training target set. e_{tgt} is a 46-D vector in which each component varies freely in [0,1], following the convention of the FaceWarehouse database [20], defined in (1.3) in Section 1.3.

Our general GATH framework is illustrated in Fig. 4.2. The generator G synthesizes x_{tgt} from the input x_{src} given AU coefficients e_{tgt}: $x_{tgt} = G(x_{src}, e_{tgt}) = f_{G\_de}(f_{G\_en}(x_{src}), e_{tgt})$. Since x_{src} may contain arbitrary expression, the generator specifically disentangles the latent identity code from expression features in the source image with the encoder, effectively making the transformation of the source face independent of the expression manifested in the input image.

Unlike previous work [78, 126], our source and target sets are disjoint[1]. In other words, an exact correspondence y^0_{tgt} of x_{tgt} does not exist, hence we are unable to use the conventional pixel-wise reconstruction loss to learn facial deformation. However, there exists a frame y_{tgt} that shares similar values of AU intensities to x_{tgt}. One might naively minimize the difference $\|x_{tgt} - y_{tgt}\|$, but using this loss has major drawbacks: Firstly, there is not necessarily pixel-wise correspondence between x_{tgt} and y_{tgt}, hence a local facial deformation at a specific coordinate in the target does not mean that the same visual change would also happen at the exact same coordinate in the source. Secondly, directly minimizing the difference between the source and the target frame would make the model learn to hallucinate the identity of the target into the source, which violates the identity preserving aspect of our model. Furthermore, what we want to compare is the expressiveness similarity of the source and target, not their entire appearances. Inspired by recent work in artistic style transfer [46], we wish to compare the source and target in a latent expressiveness space with

[1] In training, we actually mix a small amount of image samples of subjects in the target set into the source set, which accounts for 2.7% of its size, to increase its diversity.

a projecting function f_{au}. Thus, we propose to train a deep Action Unit Estimator network, and measure the similarity of source and target in the hidden feature space of AUE.

One core objective of our work is to learn a generator G that can generate realistic looking face synthesis indistinguishable from a real image, especially in our case where the exact corresponding target does not exist. To this end, we integrate the adversarial loss proposed by Goodfellow et al. [48] into our framework, by jointly training a discriminator D that can tell the difference between real and fake images, that eventually guides G to generate "fine enough" samples via a minimax game.

A straightforward approach to learn identity disentanglement is to minimize the intra-subject reconstruction loss $\|x_{tgt} - x_{src}\|$, as they are largely similar except sparse local deformation parts. However, we can utilize the available auxiliary class labels of the source set to provide additional feedback to make the generator learn the disentanglement more effectively. We propose to jointly train a classifier $\mathcal{C}$ that share all hidden layer weights with the discriminator. The advantages of this approach are two-fold. First, jointly learning the classifier $\mathcal{C}$ and discriminator D helps discover relevant facial hidden features better, and D can tell apart the real image from the fake more easily. In return, these players provide stronger feedback to the generator, encouraging G to generate finer synthesis and better preserve the identity of the source.

4.3 Action Unit Estimator

We propose a CNN model $\mathcal{E}$ based on VGG-9 architecture [102] to predict AU coefficients from a facial image: $\mathcal{E} : x \rightarrow e$. The network architecture is shown in Fig. 4.3c. $\mathcal{E}$ is learned by minimizing the squared loss:

$$\min_{\mathcal{E}} \|e_{gt} - \mathcal{E}(x)\|_2^2, \tag{4.1}$$

where e_{gt} is a ground truth AU vector. In essence, AUE learns hidden features tailored to facial expression and independent of identity, as well as invariant to position and scale of the face in the image. We take the last convolutional layer of AUE as the latent space mapping function f_{au} to measure the expressiveness similarity of images. First, this layer retains high-level features with rich details to represent the facial expression. Furthermore, the convolutional layer still preserves the spatial 2D structure layout, hence it can pinpoint where in the source image that the local deformation should happen.

4.4 GATH Learning

The models in our framework are jointly trained by optimizing the following composite loss:

$$\mathcal{L}_{au} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{tv}\mathcal{L}_{tv}. \tag{4.2}$$

The first term is the AU loss, to make G learn to expressively transform the source image to manifest the target emotion. It enforces that expressiveness features of the synthetic image are similar to those of the target example frame.

$$\mathcal{L}_{au} = \min_{G} \left\| f_{au}\left(y_{tgt}\right) - f_{au}\left(G\left(x_{src}, e_{tgt}\right)\right)\right\|_2^2. \tag{4.3}$$

The intra-subject reconstruction loss, $\mathcal{L}_{rec}$, minimizes the pixel-wise difference between the source and the synthetic image, because except some small parts on the face are deformed at one time to manifest spontaneous expression, such as eye blinking or mouth opening, the rest of the face should remain the same. In other words, this term is to preserve the subject identity as well as the background.

$$\mathcal{L}_{rec} = \min_{G} \left\| x_{src} - G\left(x_{src}, e_{tgt}\right)\right\|_1. \tag{4.4}$$

The third term is the adversarial loss. In this work, we replace the vanilla Jensen-Shannon divergence GAN loss [48] with the least square loss proposed in CycleGAN [133], as we found that this objective makes optimization more stable.

$$\mathcal{L}_{adv} = \max_{G} \min_{D} \left(1 - D\left(x_{re}\right)\right)^2 + \left(D\left(G\left(x_{src}, e_{tgt}\right)\right)\right)^2. \tag{4.5}$$

The classifier loss, $\mathcal{L}_{cls}$, consists of two cross-entropy loss terms. Minimizing the first term updates the weights of the classifier, while the second term updates the weights of G. Intuitively, the classifier updates its parameters from real image samples x_{re}, and provides feedback to the generator, such that G learns to generate better samples to lower the classification loss, and consequently preserve the source identity better.

$$\mathcal{L}_{cls} = \min_{\mathcal{C}} -\sum_{i} c_i \log \mathcal{C}_i\left(x_{re}\right) + \min_{G} -\sum_{i} c_i \log \mathcal{C}_i\left(G\left(x_{src}, e_{tgt}\right)\right). \tag{4.6}$$

The last term is total variation loss to maintain spatial smoothness of the synthetic image:

$$\mathcal{L}_{tv} = \frac{1}{HW} \sum_{i,j} \left(x_{tgt}^{i,j+1} - x_{tgt}^{i,j}\right)^2 + \left(x_{tgt}^{i+1,j} - x_{tgt}^{i,j}\right)^2. \tag{4.7}$$

In the adversarial learning framework, the generator and discriminator are alternatively and iteratively updated. In GATH, there are also two players, the generator and the joint discriminator-classifier network. Essentially, the joint discriminator-classifier network is updated by minimizing this loss:

$$\begin{aligned} \min_{D,\mathcal{C}} \lambda_{adv} \left[\left(1 - D\left(x_{re}\right)\right)^2 + \left(D\left(G\left(x_{src}, e_{tgt}\right)\right)\right)^2\right] \\ -\lambda_{cls} \sum_{i} c_i \log \mathcal{C}_i\left(x_{re}\right), \end{aligned} \tag{4.8}$$

whereas minimizing the following composite loss updates the generator:

$$\min_{G} \mathcal{L}_{au} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{tv}\mathcal{L}_{tv} - \lambda_{adv}(D\left(G\left(x_{src}, e_{tgt}\right)\right))^2 \\ -\lambda_{cls}\sum_{i} c_i \log \mathcal{C}_i\left(G\left(x_{src}, e_{tgt}\right)\right). \tag{4.9}$$

As we can see, minimizing the loss in (4.8) encourages the discriminator and classifier to recognize images from the real distribution correctly, and penalizes the discriminator if it wrongly predicts that the synthetic sample is real. On the other hand, the loss in (4.9) makes the generator learn to produce realistic looking output to fool the discriminator while having the desired facial expression and preserve personal characteristics of the portrait.

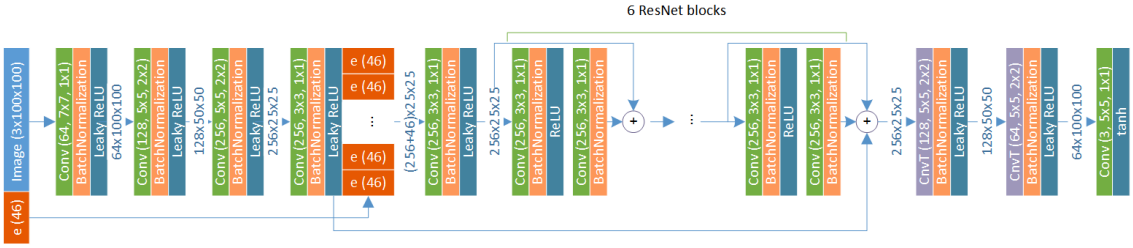
4.5 Implementation Details

4.5.1 Network Architecture

Architectures of three networks in our GATH framework are illustrated in Fig. 4.3. These networks are designed such that all of them can reside in the GPU memory of a Tesla K40 at the same time.

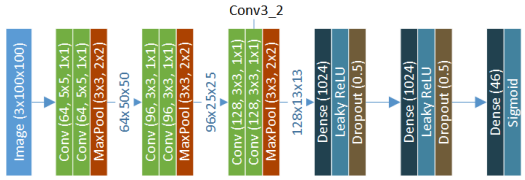
The input to each network is a standard 100x100px RGB image, with pixel values normalized to [-1,1]. The generator G includes two subnetworks: encoder and decoder. The encoder consists of four blocks, each one starts with a convolutional layer, followed by a batch normalization layer and Leaky ReLU activation with slope factor of 0.1. The AU vector is concatenated to the output of the forth block by spatially 2D broadcasting. The decoder starts with a convolutional block, followed by six ResNet [51] blocks, two convolutional transpose blocks and a final convolutional output layer. Particularly, there is a residual connection from the output of the encoder to the end of the bottle neck to help G learn the identity code more effectively.

Fig. 4.3b shows the weight sharing discriminator-classifier network. D and $\mathcal{C}$ share all hidden layer parameters, but have separate output layers.



(a) The fully convolutional generator network

(b) The discriminator-classifier network



(c) The AU Estimator network

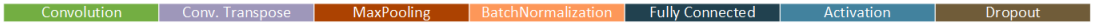


Figure 4.3: *(best viewed in color)* Architectures of deep neural networks in our GATH framework. The last convolutional layer of the AUE, 'conv3_2', is used to extract expressive hidden features to calculate $\mathcal{L}_{AU}$.

4.5.2 Training and Post-processing

Training. We organize the source set as a combination of the following still photo datasets: Cross-age Celebrity dataset (CACD) [24], FaceWareHouse [20], GTAV [106], consisting of 2,168 identities. We also mix in a small set of frames from 20 actors in RAVDESS [71] and 40 actors in VIDTIMIT [100] dataset, making a total of 2,228 identities in the source set.

The target set consists of a large number of frames extracted from RAVDESS and VIDTIMIT,

two popular audiovisual datasets, in which actors display a wide range of facial actions and emotions. We extract AU coefficients from these two datasets using our proposed face tracker in Chapter 2 [84, 83], running in RGB-tracking mode. The AUE is trained on the target set.

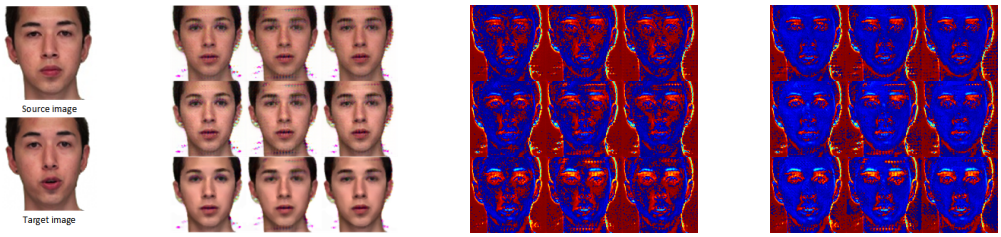
Furthermore, since these datasets include many face images at large pose, we use the face frontalization technique proposed by Hassner et al. [50] to roughly convert original images into portraits in both source and target sets and crop them to 100x100px. The alignment is not perfect, however, our AUE is invariant to translation and scale, hence our generator can learn facial deformation reliably.

The models in our GATH framework, G, D and $\mathcal{C}$, are trained end-to-end with the ADAM minibatch optimizer [60]. We set minibatch size = 64, initial learning rate = $1e-4$ and momentum = 0.9. Other hyperparameters in (4.2) are empirically chosen as follows: $\lambda_{rec} = \lambda_{tv} = 1.0$, $\lambda_{adv} = \lambda_{cls} = 0.05$. The whole framework is implemented in Python based on the deep learning toolkit CNTK.

Post-processing. A side effect of using the composite loss in (4.2) to train our weakly supervised model and pixel scaling to and from the [-1,1] range is that the synthesis loses the dynamic range of the original input. In order to partially restore the original contrast, we apply the adaptive histogram equalization algorithm CLAHE [136] to synthesized images. Parts of evaluation where CLAHE is applied will be clearly indicated. Furthermore, for visualization purpose, we clear the noise in the output with non-local means denoising [15], followed by unsharp masking. Fig. 4.4b demonstrates the visual effects of these two enhancements on the syntheses.

4.6 Evaluation

Due to the lack of a publicly available implementation of the face editing methods (mentioned in Section 4.7), we simplify our GATH framework to create two baselines. **GATH-DC** (GATH *minus* DC): GATH without the joint discriminator-classier network; **GATH-C**: GATH without the



(a) (b) syntheses (c) original outputs (d) after CLAHE

Figure 4.4: (a) Source and target images of the same tester. (b) From left to right: syntheses created by two baselines and GATH; top to bottom: raw outputs, histogram-equalized (CLAHE) outputs and sharpened outputs, respectively. (c,d) The pixel-wise error heat maps of one sample in two cases: the raw output and CLAHE-applied output. In each figure, from left to right: output error of GATH-DC, GATH-C and GATH, respectively; from top to bottom: error heat maps on three channels B, G and R, respectively.

classifier.

For quantitative evaluation, we record facial expression synthesizing performance on a hold out test set of four actors from RAVDESS and three actors in VIDTIMIT, which are not included in training. Specifically, we choose two source frames from each actor. In one frame, the actor displays neutral (or close to neutral) expression. In the other one, the actor shows at least one expression (mouth opening). Each source image is paired with all video sequences (94 in total), resulting in 188 intra-class (same subject) pairs and 1032 inter-class pairs, where the source actor is different from the target actor.

We quantify the synthesis performance of our model with a set of metrics: for intra-class experiments, we measure the pixel-wise Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), as well as the AU error (RMSE of intensity values), with respect to ground truth frame. We use the OpenFace toolkit [6] to extract intensities of 17 AUs (which are different from our set of 46 AUs), each value varies within the range of [0,5] (note that our input AU coefficients vary in [0,1]). For inter-class experiments, we only measure the AU error.

We also provide qualitative experiments on a random set of actors from the two popular face

datasets: CelebA [68] and Labeled Face in the Wild (LFW) [54].

Lastly, we present an application of *template-and-target-free expression editing*, where user-defined AU coefficients are used to transform arbitrary sources. Mainly within the scope of this paper, we perform expression suppression (i.e. neutralization). However, our model is flexible enough to transform a source facial image with any arbitrary AU values.

4.6.1 Intra-class Synthesis

Table 4.1 shows pixel-wise MAE and RMSE when comparing the synthesized output with the ground truth image of the same subject (i.e. x_{tgt} and y_{tgt} are of the same person, manifesting the same expression and they should look the very similar). The errors are organized by dataset, and gathered from four different settings: whether taking the background error into account or not (by using a mask to localize the face region), and whether using CLAHE to increase the contrast of the outputs. It is observed that training only the generator, without feedback from the discriminator and classifier, actually makes the model produce better pixel-wise color reconstruction without using histogram equalization, although the difference in error is rather small. After apply CLAHE, the generated outputs of GATH have smallest errors. It is also observed from Table 4.1 that the reconstructed background pixels actually incur higher error than the face region. In addition, the error heat map in Fig. 4.4 indicates that errors on the face region are almost uniform, indicating a constant shift in the color space.

However, color reproduction is only one criterion to measure the quality of expression synthesis. Our main objective in this paper is to synthesize animation driven by AU coefficients. The AU estimation errors with respect to the ground truth frame are shown in Table 4.2. It proves that the output of GATH has higher fidelity than the two baselines, as better images will help OpenFace estimate AU intensities more accurately. It is also unsurprising that GATH-C, trained jointly with a discriminator, performs better than GATH-DC. These results prove the benefit of our proposed

Table 4.1: Pixel-wise MAE and RMSE of intra-class synthesis.

	MAE			RMSE		
	RAVDESS	VIDTIMIT	All	RAVDESS	VIDTIMIT	All
full image, without CLAHE						
GATH-DC	**141.53**	**112.78**	**127.16**	**182.31**	**162.47**	**172.68**
GATH-C	144.8	119.13	131.97	184.66	167.34	176.21
GATH	146.46	118.53	132.5	185.8	167.02	176.66
full image, with CLAHE						
GATH-DC	94.49	**58.66**	76.58	138.62	**94.26**	118.53
GATH-C	92.31	59.35	75.83	136.86	95.23	117.89
GATH	**91.65**	59.66	**75.66**	**136.21**	95.89	**117.79**
mask, without CLAHE						
GATH-DC	**128.48**	**122.36**	**125.42**	**174**	**170.94**	**172.47**
GATH-C	134.78	128.53	131.66	178.44	175.48	176.97
GATH	136.14	130.16	133.15	179.48	176.78	178.13
mask, with CLAHE						
GATH-DC	56.47	55.42	55.94	93.42	85.49	89.54
GATH-C	57.37	55.28	56.32	94.84	85.26	90.18
GATH	**55.88**	**54.97**	**55.43**	**92.63**	**84.47**	**88.64**

Table 4.2: RMSE of Action Unit Itensity in intra-class synthesis.

	RAVDESS	VIDTIMIT	All
GATH-DC	0.592	0.35	0.486
GATH-C	0.591	0.336	0.481
GATH	**0.585**	**0.334**	**0.477**

GATH model in synthesizing facial expressions.

Fig.s 4.5, 4.6 demonstrate the synthetic results of GATH. The generated texture has been enhanced visually with the aforementioned post-processing procedure. Notice that in Fig. 4.5b,c,d, it is shown that our model is able to hallucinate eye-blinking motions.

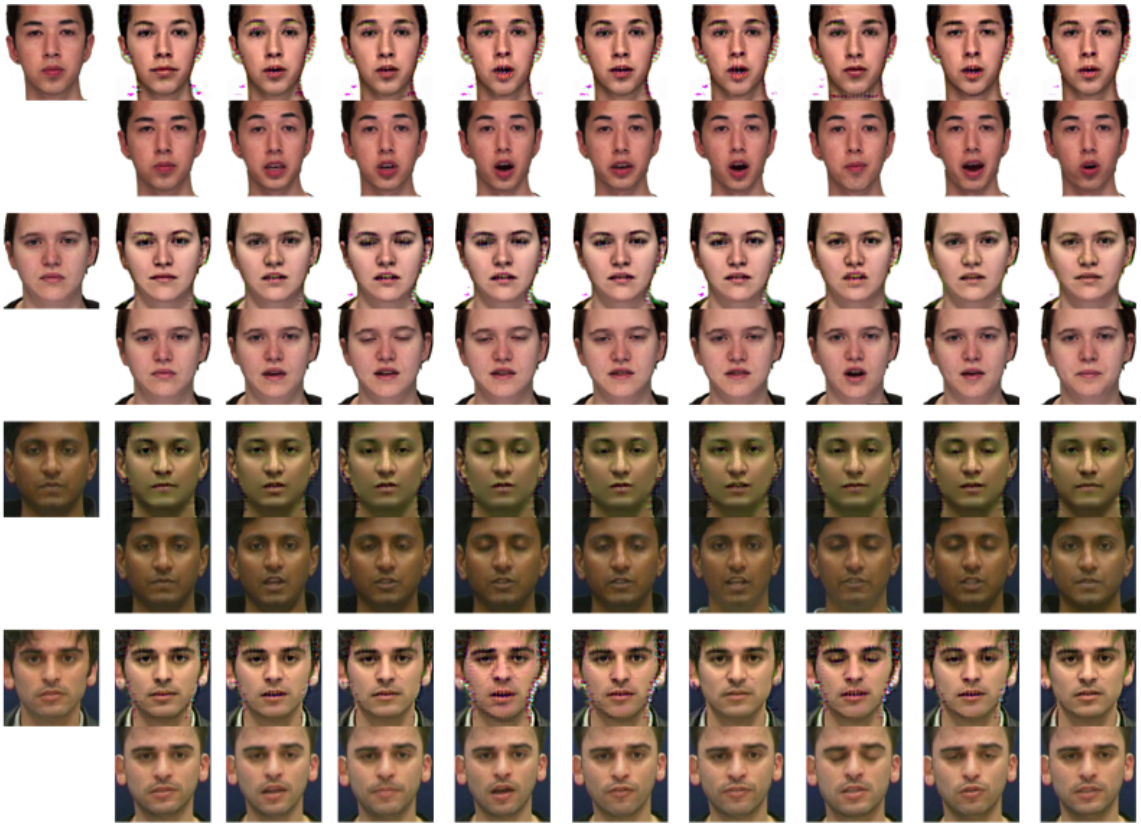


Figure 4.5: Samples from four sequences in paired evaluation. For each sequence, the top left is the still source image x_{src}, and it is post-processed. x_{src} is at neutral expression. In each vertical pair, the top image is the hallucinated frame x_{tgt}, while the corresponding target frame y_{tgt} is at the bottom. In the 4^{th} sequence, the source image and the target video were captured at two different occasions.

4.6.2 Inter-class Synthesis

Table 4.3: RMSE of Action Unit intensity in inter-class synthesis.

GATH-DC	GATH-C	GATH
0.587	0.583	**0.579**

In this evaluation, we compare the AU estimation scores returned by OpenFace on the ground truth of a subject, and scores on the corresponding syntheses. The results are shown in Table 4.3.

Figure 4.6: Samples from three sequences, starting with non-neutral expression source images. The model learned to synthesize the ”closed lip” expression when target is neutral.

Once again, the full GATH model outperforms the two baselines.

4.6.3 Qualitative Assessments on In-The-Wild Images

Fig. 4.7 and 4.8 show animated sequences by GATH, in which the source images are sampled from the CelebA and LFW datasets, respectively, with diversity across genders, skin colors, styles etc. Interestingly in Fig. 4.7c, the model even synthesizes eyes beyond the shades.

4.6.4 Template-and-target-free Expression Editing

In this experiment, we perform expression suppression, transforming a face with arbitrary expression back to the neutral pose. Source images are sampled from the CelebA dataset. The qualitative results are illustrated in Fig. 4.9,4.10. We transform the source to neutral expression simply by providing GATH with a zero AU vector. It proves that via our learning framework, the

Figure 4.7: Samples from six animated sequences, with the source images taken randomly from CelebA dataset, with different genders, skin colors, hair styles, etc.

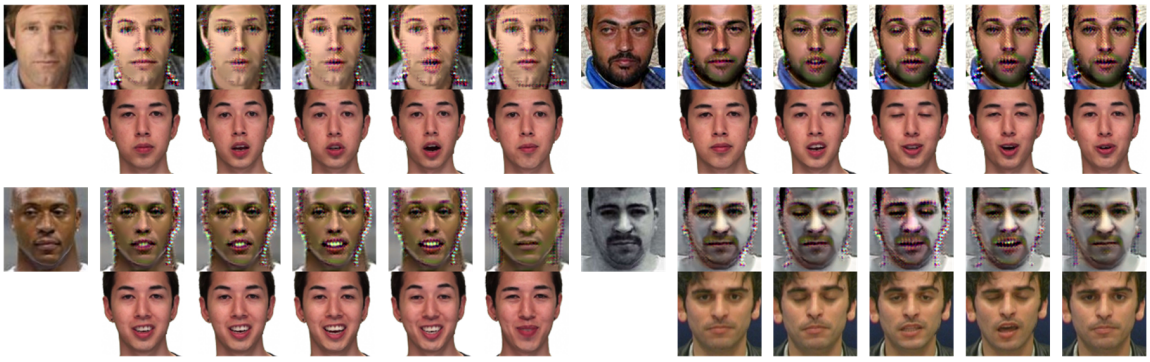


Figure 4.8: Samples from four animated sequences, in which the source images were taken randomly from LFW dataset.

generator has learned to disentangle the identity code from expression. Thus, giving GATH zero AU coefficients equals generating a neutral face of the source actor.

Figure 4.9: Samples from expression suppression editing with our model on the CelebA dataset. In each pair of images, the source is on the left, the suppressed synthesis is shown on the right.

4.6.5 Limitations

Our GATH model has proved to be able to synthesize novel face from arbitrary source. However, there still exists some issues remaining:

- The synthesized image loses its texture dynamic range.
- There is still color noise and distortions in the reconstructed face, especially around the face contour and strong edges.

We will investigate these issues thoroughly to make GATH more robust and generate higher

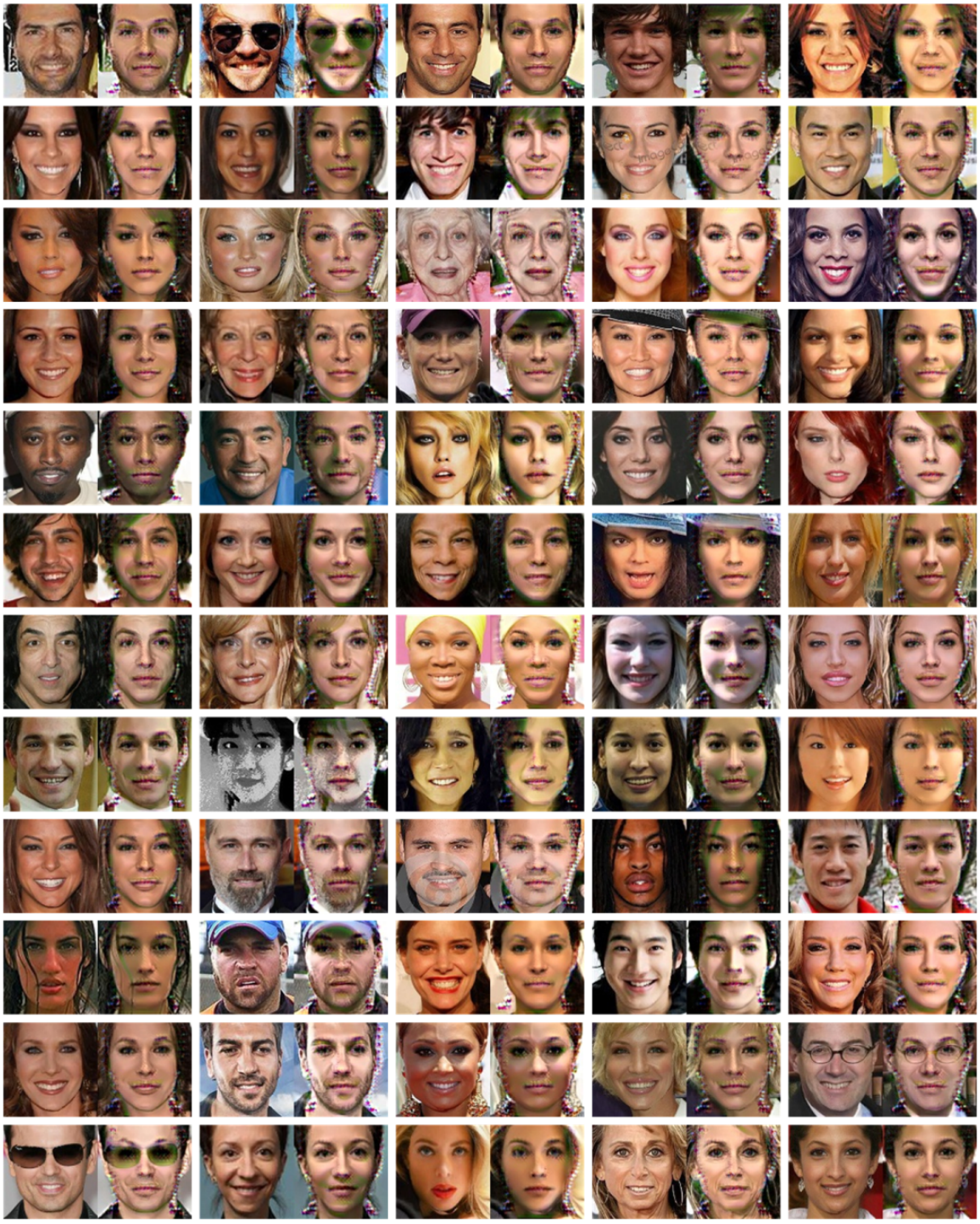


Figure 4.10: More samples from expression suppression editing with our model on the CelebA dataset.

quality face synthesis in future work.

4.7 Related Work

Generative Adversarial Nets (GAN). Proposed by Goodfellow et al. [48], GAN learns the generative model in a minimax game, in which the generator and discriminator gradually improve themselves. Eventually the generator learns to create realistic data able to fool the discriminator. GAN has been widely used in image synthesis with various successes [36, 64, 90, 91, 56, 25]. Moreover, recent works also introduce additional constraints for topic-driven synthesis [75], or use class labels in semi-supervised GAN training [104, 77, 110, 65]. In our approach, a classifier is jointly trained with the discriminator to predict synthesized images into C classes. Consequently, not only the generator learns to generate realistic images, but the synthesis also preserves the identity presented in the source image.

Facial image editing. Facial editing techniques in literature are mostly model-based, using a 3DMM [11, 114], and follow a common approach, in which a 3DMM is fitted to both source and target images, and the target expression is transferred to the source frame by manipulating the model coefficients to calculate a new 3D facial shape, followed by texture remapping [9, 108, 44, 32, 124, 123]. Averbuch-Elor et al. [5] only use 2D face alignment with clever texture warping and detail transfer [69]. Instead of using graphics-based texture warping, Orszewski et al. [78] utilize supervised GAN to synthesize a new albedo in the UV texture space, given matching source and target images. In a different approach, Liu et al. [70] uses conditional GAN to synthesize expression coefficients of a 3DMM given discrete AU labels, followed by standard shape calculation and texture remapping. Based on variational autoencoder (VAE) [61], Yet et al. [126] train an expression flow VAE from matching source-target pairs, and edit the latent code to manipulate facial expression. GANimation [88] learns to transform the image in an unsupervised fashion similar to StarGAN [25], but the expression code has been changed from discreet to continuous.

Deepfakes [4], which has gone viral on the internet recently, uses coupled autoencoders and texture mapping to be able to swap identities of two actors. In contrast to model-based approaches, our model directly generates the facial image from source portrait and target AU coefficients without using a statistical face model or a target video for transferring, forgoing manual texture warping and blending as these tasks are automatically carried out by the deep net. Different from recent GAN-based synthesis models, our method trains the generator from totally unmatched source-target pairs. Unlike GANimation in which AU intensities are randomly sampled during training, in our work target expression parameters are well defined, taken from the natural facial expression manifold. Furthermore, GATH specifically disentangles the identity code from expression in the source portrait, thus it can freely change the intensity of facial expression while keeping the identity invariant. For example our model can perform expression suppression and continuously increase and decrease the expression intensity at ease.

Representation disentangling. It is still an open question of how to design proper objectives that can effectively learn good latent representation from data. Kulkarni et al. [62], Yang et al. [125] propose models that can explicitly separate different codes (object type, pose, lighting) from input images. Those codes can be manipulated to generate a different looking image, e.g. by changing the pose code. Peng et al. [80] propose a recurrent encoder-decoder network for face alignment, that also learns to separate identity and expression codes through a combination of auxiliary networks and objectives. Tran et al. [110] employ semi-supervised GAN to learn disentangled face identity/pose for face frontalization. GATH is similar in spirit to [110], in which the encoder sub-network of the generator learns the latent identity code independently from the arbitrary expression in the source image, and the decoder takes in the combined identity and AU code to generate an animated image from the source portrait.

4.8 Summary

In this chapter, we introduced Generative Adversarial Talking Head, a generative neural net that is capable of synthesizing novel expressive faces from any source portrait, given a vector of action unit coefficients. In our GATH framework, we jointly train a generator with a adversarial discriminator and a classifier, while being supervised by an AU estimator to make the generator learn correct expression deformations, as well as simultaneously disentangle identity features from expressions. Our model directly manipulates image pixels to hallucinate a novel facial expression, while preserving the individual characteristics of the source face, without using a statistical face template or texture rendering.

Extensive experiments on different challenging datasets showed GATH can extract identity code from any given portrait regardless of the facial expression displayed in the image. It was also demonstrated that jointly learning the discriminator and classifier improves synthesis performance of the generator. Furthermore, GATH works directly on AU parameters without the need of a target facial image. Hence, GATH can perform both facial expression enactment and suppression at will as desired by the user, completely template-and-target-free.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Facial expression not only is a natural means of communication between humans, but also, in the coming age of artificial intelligence and virtual persona, it has become an inherent part of human-machine interaction, in which the agent is able to perceive human emotion as well as express its own "feeling" in this mutual communication. Understanding emotion through facial expression is important because it is a partially determining factor in any conversation, be it between humans or especially human-machine, where the agent cannot recognize or create facial expressions on its own. In other words, the intelligent agent needs to be equipped with the capabilities to analyze and synthesize facial expressions. For decades there has been tremendous research effort, this work included, to enable these capabilities algorithmically for the machine.

In this work, we introduced machine learning approaches for facial expression analysis from either video or audio, and facial expression synthesis, including: a real-time 3D face tracking framework for RGBD video, a family of recurrent neural networks for real-time speech-driven facial expression analysis, and finally, a generative deep neural network for arbitrary facial expression synthesis from any given portrait.

First, we proposed a real-time, robust 3D face tracking framework from RGBD video that can simultaneously capture head movements, facial expressions and adapt expression blendshapes to match identity of the tracked subject in unconstrained environments. Our tracker is driven by a

highly efficient person-agnostic 3D shape regressor based on random forest and linear regression and trained on standard image data. The subsequent joint 2D+3D optimization intelligently registers the 3D face shape to both color and depth data more accurately. This combination in our approach addresses several issues that commonly exist in other face trackers. Learning 3D regression makes the tracker robust to lower-quality input and not rely on depth data, which is noisy at large distance. On the other hand, registering 3D shape to depth data resolves the inherent depth ambiguity in 3D alignment, thus our method can register and reconstruct more accurate, better 3D face blendshape, improve the tracking performance in general. We further extend our 3D shape regressor to support profile-to-profile face alignment, making it robust to extreme head poses. Experimental results on real and synthetic datasets showed that our proposed face tracker is comparable to or outperforms state-of-the-art face tracking methods in alignment accuracy and tracking reliability.

Next, we proposed different recurrent neural network models for real-time facial expression analysis from speech. Specifically, our models estimate facial action unit intensities of a speaker which is carried in the audio signal. These facial action unit intensities not only depict the sound-uttering actions via lip deformations, but also implicitly present the affective states of the speaker, such as raising eyebrows when being amused, or smiling when she is happy. In the proposed baseline model, handcrafted acoustic features are used to predict facial actions. We also show that it is more advantageous to learn acoustic feature representation from audio input directly in our end-to-end models, which indeed improved the quality of facial action estimation. Quantitative and qualitative experiments on diverse and challenging audiovisual corpora of different actors across a wide range of facial expressions, voices and languages showed that our proposed models can predict facial action from speech reasonably well, and they can generalize to unseen speech patterns, thanks to the use of low-level features.

Finally, we proposed a novel deep generative neural network, GATH, that can synthesize any

desired facial expression specified by action unit weights on any given arbitrary portrait. In particular, our model directly manipulates image pixels, making the subject in the portrait express various, continuous facial expressions controlled by AU parameters, while maintaining her personal characteristics. This is because our model can learn to disentangle identity code from expressiveness features, so that it can generate novel facial expressions regardless of the expression that the subject portrays in the photo. Furthermore, our model is trained from unpaired data, where the source and target images are of different persons, and the desired expressive image of the source subject does not exist. In order to effectively learn such model, we propose a novel weakly supervised adversarial learning framework that consists of a generator, a discriminator, a classifier and an action unit estimator. Not relying on any face template and target image, our synthesis model enables extremely flexible, template-and-target-free facial expression editing. As demonstrated in our experiments on in-the-wild datasets, our model can perform both facial expression enactment and suppression on arbitrary portrait of any subject, while preserving her identity.

5.2 Future Work

Although our proposed approaches for facial expression analysis and synthesis have performed well for their intended purposes, there still exists some limitations. We suggest the following potential improvements as a guideline:

- Our proposed tracker only models 3D facial shape and completely ignores textures, hence the reconstructed blendshape only present geometric surface details. If a fully textured face model is required, we have to perform an additional texture mapping step. Hence, it suggests that we can also incorporate parametric facial texture into our framework. Furthermore, using dense pixel color information from texture may improve the accuracy of 3D face alignment.

- The 3D shape regressor in our proposed tracker takes features encoded in leaves of the random forest as input to regression. However, if we want to model more complex, “deeper”

features, we would have to increase the depth of the forest, consequently growing exponentially the number of leaves, or in other words, the dimension of feature vector. It would make the size of the regression matrix too big, even though the amount of calculation remains the same and the whole regressor is still efficient. It suggests that replacing our proposed 3D regressor with a deeper neural net may improve face alignment accuracy, at the cost of requiring more computational power.

- Our speech-driven recurrent neural networks were able to predict lip deformations rather well as shown in our experiments. However, eyebrows-related actions are still not estimated as effectively. We will experiment with other network architectures and objective losses to overcome this problem.

- Our synthesis model, GATH, can produce rather accurate facial expressions specified by action unit parameters when compared to the expression in the target image, as shown in our experiments. However, GATH still created color artifacts in the output image. A possible solution is to make the AU Estimator purer, by designing a deeper network.

- Currently, GATH can only work with frontal faces. In the future, we will incorporate head pose into the model, so that GATH can generate a novel face with any expression and pose. We will also modify the generator into a Variational Autoencoder, such that GATH can generate any random yet realistic identity.

References

[1] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu. Convolutional neural networks for speech recognition. *IEEE Transaction on Audio, Speech, and Language Processing*, 22(10), October 2014.

[2] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, and G. Penn. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012.

[3] J. Ahlberg. An updated parameterized face. Technical report, Image Coding Group, Dept. of Electrical Engineering, Linkoping University, 2001.

[4] U. Author. Deepfakes. `https://github.com/deepfakes/faceswap`.

[5] H. Averbuch-Elor, D. Cohen-Or, J. Kopf, and M. F. Cohen. Bringing portraits to life. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia*, 36(196), 2017.

[6] T. Baltrušaitis, P. Robinson, , and L.-P. Morency. Openface: an open source facial behavior analysis toolkit. In *IEEE Winter Conference on Applications of Computer Vision*, 2016.

[7] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross. High-quality single-shot capture of facial geometry. *ACM Transactions on Graphics (TOG)*, 29(4), 2010.

[8] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross. High-quality passive facial performance capture using anchor frames. *ACM Transactions on Graphics (TOG)*, 29(4), 2010.

[9] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. In *SIGGAPH*, pages 187–194, 1999.

[10] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, 1999.

[11] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Eurographics*, pages 641–650, 2003.

[12] T. Bänziger, M. Mortillaro, and K. Scherer. Introducing the geneva multimodal expression corpus for experimental research on emotion perception. *Emotion*, 12(5):1161–1179, 2012.

[13] S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for realtime facial animation. In *SIGGRAPH*, 2013.

[14] C. Bregler, M. Covell, and M. Slaney. Video rewrite: driving visual speech with audio. In *SIGGRAPH*, pages 353–360, 2007.

[15] A. Buades, B. Coll, and J.-M. Morel. Non-Local Means Denoising. *Image Processing On Line*, 1:208–212, 2011.

[16] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang. 3d deformable face tracking with a commodity depth camera. In *Europ. Conf. Comput. Vision (ECCV)*, 2010.

[17] C. Cao, D. Bradley, K. Zhou, and T. Beeler. Real-time high fidelity facial performance capture. In *SIGGRAPH*, 2015.

[18] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. In *SIGGRAPH*, 2014.

[19] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3d shape regression for real-time facial animation. In *SIGGRAPH*, 2013.

[20] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. FaceWarehouse: A 3D Facial Expression Database for Visual Computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, March 2014.

[21] C. Cao, H. Wu, Y. Weng, T. Shao, and K. Zhou. Real-time facial animation with image-based dynamic avatars. *ACM Transactions on Graphics (TOG)*, 35(4), 2016.

[22] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *CVPR*, 2012.

[23] Y. Cao, W. C. Tien, P. Faloutsos, and F. Pighin. Expressive speech-driven facial animation. *ACM Transactions on Graphics*, 24(4):1283–1302, 2005.

[24] B.-C. Chen, C.-S. Chen, and W. H. Hsu. Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. *IEEE Transactions on Multimedia*, 17(6):804–815, 2015.

[25] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *arXiv*, 2017.

[26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Deep Learning and Representation Learning Workshop*, 2014.

[27] M. Cooke, J. Barker, S. Cunningham, and X. Shao. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120:2421–4, 12 2006.

[28] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Trans. Pat. Anal. Mach. Intel.*, 23:681–684, 2001.

[29] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their tranining and applications. *Comput. Vis. Image Underst.*, 61:39–59, 1995.

[30] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor. View-based active appearance models. *Image and Vis. Comput.*, 20(9):657–664, 2002.

[31] E. Cosatto, J. Ostermann, H. P. Graf, and J. Schroeter. Lifelike talking faces for interactive services. *Proc IEEE*, 91(9):1406–1429, 2003.

[32] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfster. Video face replacement. *ACM Transactions on Graphics (TOG)*, 30(6), 2011.

[33] M. Dantone, J. Gall, G. Fanelli, and L. V. Gool. Real-time facial feature detection using conditional regression forests. In *CVPR*, 2012.

[34] D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *Int. J. Comput. Vis.*, 38(2):99–127, July 2000.

[35] L. Deng, O. Abdel-Hamid, and D. Yu. A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[36] E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015.

[37] C. Ding, L. Xie, and P. Zhu. Head motion synthesis from speech using deep neural network. *Multimed Tools Appl*, 74:9871–9888, 2015.

[38] P. Ekman and W. Friesen. Facial action coding system: A technique for the measurement of facial movement. *Consulting Psychologists Press*, 1978.

[39] P. Ekman and W. V. Friesen. Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17:124–129, 1971.

[40] P. Ekman, W. V. Friesen, and P. Ellsworth. Emotion in the human face: Guidelines for research and a review of findings. *New York: Permagon*, 1972.

[41] P. Ekman, W. V. Friesen, and M. O'Sullivan. Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of Personality and Social Psychology*, 53:712–717, 1987.

[42] T. Ezzat, G. Geiger, and T. Poggio. Trainable videorealistic speech animatio. In *SIGGRAPH*, pages 388–397, 2002.

[43] B. Fan, L. Xie, S. Yang, L. Wang, and F. K. Soong. A deep bidirectional lstm approach for video-realistic talking head. *Multimed Tools Appl*, 75:5287–5309, 2016.

[44] P. Garrido, L. Valgaerts, O. Rehmsen, T. Thormahlen, P. Perez, and C. Theobalt. Automatic face reenactment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[45] P. Garrido, L. Valgaerts, C. Wu, and C. Theobalt. Reconstructing detailed dynamic face geometry from monocular video. *ACM Transactions on Graphics (TOG)*, 32(6), 2013.

[46] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

[47] B. Glocker, O. Pauly, E. Konukoglu, and A. Criminisi. Joint classification-regression forests for spatially structured multi-object segmentation. In *ECCV 2012 - 12th European Conference on Computer Vision*. Springer, 2012.

[48] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozairy, A. Courville, and Y. Bengioz. Generative adversarial nets. In *NIPS*, 2014.

[49] S. Haq, P. Jackson, and J. Edge. Audio-visual feature selection and reduction for emotion classification. In *Proc. Int. Conf. on Auditory-Visual Speech Processing (AVSP'08), Tangalooma, Australia*, Sept. 2008.

[50] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2015.

[51] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[52] S. Hochreiter and J. Scmidhuber. Long short-term memory. *Neural Comput*, 9(8):1735–1780, 1997.

[53] Y. Hoshen, R. J. Weiss, and K. W. Wilson. Speech acoustic modeling from raw multichannel waveforms. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015.

[54] G. B. Huang, M. Ramesh, T. Berg, and E. 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.

[55] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32 nd International Conference on Machine Learning*, 2015.

[56] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2017.

[57] J. G. J. Orozco, O. Rudovic and M. Pantic. Hierarchical on-line appearance-based tracking for 3D head pose, eyebrows, lips, eyelids and irises. *Image and Vis. Comput.*, 31(4):322–340, 2013.

[58] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. In *SIGGRAPH*, 2017.

[59] V. Kazemi, C. Keskin, J. Taylor, P. Kohli, and S. Izadi. Real-time face reconstruction from a single depth image. In *3DV*, 2014.

[60] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations*, 2015.

[61] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[62] T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum. Deep convolutional inverse graphics network. In *NIPS*, 2015.

[63] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time-series. *The handbook of brain theory and neural networks*, pages 255–258, 1998.

[64] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2017.

[65] C. Li, K. Xu, J. Zhu, and B. Zhang. Triple generative adversarial nets. In *31st Conference on Neural Information Processing Systems (NIPS)*, 2017.

[66] H. Li, J. Yu, Y. Ye, and C. Bregler. Realtime facial animation with on-the-fly correctives. In *SIGGRAPH*, 2013.

[67] S. Z. Li, Y. Shuicheng, H. Zhang, and Q. Cheng. Multi-view face alignment using direct appearance models. In *Automatic Face and Gesture Recognition*, pages 324–329, 2002.

[68] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[69] Z. Liu, Y. Shan, , and Z. Zhang. Expressive expression mapping with ratio images. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001.

[70] Z. Liu, G. Song, J. Cai, T.-J. Cham, and J. Zhang. Conditional adversarial synthesis of 3d facial action units. *arXiv*, 2017.

[71] S. R. Livingstone, K. Peck, and F. A. Russo. Ravdess: The ryerson audio-visual database of emotional speech and song. In *22nd Annual Meeting of the Canadian Society for Brain, Behaviour and Cognitive Science (CSBBCS)*, 2012.

[72] K. Low. Linear least-squares optimization for point-to-plane ICP surface registration. Technical Report TR04-004, Department of Computer Science, University of North Carolina at Chapel Hill, 2004.

[73] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30 nd International Conference on Machine Learning*, 2013.

[74] I. Matthews and S. Baker. Active appearance models revisited. *Int. J. Comput. Vis.*, 60(2):135–164, 2004.

[75] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv*, 2014.

[76] C. Mutto, P. Zanuttigh, and G. Cortelazzo. Microsoft kinect range camera. In *Time-of-Flight Cameras and Microsoft Kinect*, SpringerBriefs in Electrical and Computer Engineering, pages 33–47. Springer US, 2012.

[77] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning*, 2017.

[78] K. Olszewski, Z. Li, C. Yang, Y. Zhou, R. Yu, Z. Huang, S. Xiang, S. Saito, P. Kohli, and H. Li. Realistic dynamic facial textures from a single image using gans. In *ICCV*, 2017.

[79] D. Palaz, R. Collobert, and M. Magimai-Doss. Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks. In *Interspeech*, 2013.

[80] X. Peng, R. S. Feris, X. Wang, and D. Metaxas. A recurrent encoder-decoder network for sequential face alignment. In *ECCV*, 2016.

[81] H. X. Pham, S. Cheung, and V. Pavlovic. Speech-driven 3d facial animation with implicit emotional awareness: a deep learning approach. In *The 1st DALCOM workshop, CVPR*, 2017.

[82] H. X. Pham and V. Pavlovic. Hybrid On-line 3D Face and Facial Actions Tracking in RGBD Video Sequences. In *22nd International Conference on Pattern Recognition, ICPR*, pages 4194–4199, 2014.

[83] H. X. Pham and V. Pavlovic. Robust real-time 3d face tracking from rgbd videos under extreme pose, depth, and expression variations. In *3DV*, 2016.

[84] H. X. Pham, V. Pavlovic, J. Cai, and T. jen Cham. Robust real-time performance-driven 3d face tracking. In *ICPR*, 2016.

[85] H. X. Pham, Y. Wang, and V. Pavlovic. End-to-end learning for 3d facial animation from speech. In *International Conference on Multimodal Interaction*, 2018.

[86] H. X. Pham, Y. Wang, and V. Pavlovic. End-to-end learning for 3d facial animation from speech. In *International Conference on Multimodal Interaction*, 2018.

[87] H. X. Pham, Y. Wang, and V. Pavlovic. Generative adversarial talking head: Bringing portraits to life with a weakly supervised neural network. *CoRR*, abs/1803.07716, 2018.

[88] A. Pumarola, A. Agudo, A. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[89] Y. Qian, Y. Fan, and F. K. Soong. On the training aspects of deep neural network (dnn) for parametric tts synthesis. In *ICASSP*, pages 3829–3833, 2014.

[90] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representation*, 2016.

[91] S. Reed, Z. Akata, X. Yan, L. Logeswaran, H. Lee, and B. Schiele. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, 2016.

[92] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *CVPR*, 2014.

[93] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *ICCV*, 2003.

[94] S. Romdhani and T. Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *CVPR*, 2005.

[95] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A. rahman Mohamed, G. Dahl, and B. Ramabhadran. Deep convolutional neural networks for large-scale speech tasks. *Neural Network*, 64:39–48, 2015.

[96] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015.

[97] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals. Learning the speech front-end with raw waveforms cldnns. In *Interspeech*, 2015.

[98] S. Sako, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Hmm-based text-to-audio-visual speech synthesis. In *ICSLP*, pages 25–28, 2000.

[99] G. Salvi, J. Beskow, S. Moubayed, and B. Granstrom. Synface: speech-driven facial animation for virtual speech-reading support. *URASIP journal on Audio, speech, and music processing*, 2009.

[100] C. Sanderson and B. Lovell. Multi-region probabilistic histograms for robust and scalable identity inference. *Lecture Notes in Computer Science (LNCS)*, 5558:199–208, 2009.

[101] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, 2011.

[102] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[103] B. M. Smith, J. Brandt, Z. Lin, and L. Zhang. Nonparametric context modeling of local appearance for pose-and expression-robust facial landmark localization. In *CVPR*, pages 1741–1748, 2014.

[104] J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *International Conference on Learning Representation*, 2016.

[105] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Schlizerman. Synthesizing obama: learning lip sync from audio. In *SIGGRAPH*, 2017.

[106] F. Tarrés and A. Rama. GTAV face database. `https://gtav.upc.edu/research-areas/face-database`.

[107] S. Taylor, T. Kim, Y. Yue, M. Mahler, J. Krahe, A. G. Rodriguez, J. Hodgins, and I. Matthews. A deep learning approach for generalized speech animation. In *SIGGRAPH*, 2017.

[108] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[109] D. Thomas and R.-I. Taniguchi. Augmented blendshapes for real-time simultaneous 3d head modeling and facial motion capture. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[110] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2017.

[111] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *Interspeech*, 2016.

[112] L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Transactions on Graphics (TOG)*, 31(6), 2012.

[113] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages I–511 – I–518, 2001.

[114] D. Vlasic, M. Brand, H. Pfster, and J. Popović. Face transfer with multilinear models. *ACM Transactions on Graphics (TOG)*, 24, 2005.

[115] A. Wang, M. Emmi, and P. Faloutsos. Assembling an expressive facial animation system. *ACM SIGGRAPH Video Game Symposium (Sandbox)*, pages 21–26, 2007.

[116] L. Wang, X. Qian, W. Han, and F. K. Soong. Synthesizing photo-real talking head via trajectoryguided sample selection. In *Interspeech*, pages 446–449, 2010.

[117] L. Wang, X. Qian, F. K. Soong, and Q. Huo. Text driven 3d photo-realistic talking head. In *Interspeech*, pages 3307–3310, 2011.

[118] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. In *SIGGRAPH*, 2011.

[119] Z. Wu, S. Zhang, L. Cai, and H. Meng. Real-time synthesis of chinese visual speech and facial expressions using mpeg-4 fap features in a three-dimensional avatar. In *Interspeech*, pages 1802–1805, 2006.

[120] L. Xie and Z. Liu. Realistic mouth-synching for speech-driven talking face using articulatory modeling. *IEEE Trans Multimed*, 9(23):500–510, 2007.

[121] X. Xiong and F. D. la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013.

[122] X. Xiong and F. D. la Torre. Global supervised descent method. In *CVPR*, 2015.

[123] F. Yang, L. Bourdev, E. Shechtman, J. Wang, and D. Metaxas. Facial expression editing in video using a temporally-smooth factorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[124] F. Yang, J. Wang, E. Shechtman, L. Bourdev, , and D. Metaxas. Expression flow for 3d-aware face component transfer. *ACM Transactions on Graphics (TOG)*, 30, 2011.

[125] J. Yang, S. Reed, M.-H. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *NIPS*, 2015.

[126] R. A. Yeh, Z. Liu, D. B. Goldman, and A. Agarwala. Semantic facial expression editing using autoencoded flow. *arXiv preprint arXiv:1611.09961*, 2016.

[127] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In *8th IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–6. IEEE, Sept 2008.

[128] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *ICCV*, 2013.

[129] X. Yu, Z. Lin, J. Brandt, and D. Metaxas. Consensus of regression for occlusion-robust facial feature localization. In *European Conf. Computer Vision (ECCV)*, Sep 2014.

[130] H. Zen, A. Senior, and M. Schuster. Statistical parametric speech synthesis using deep neural networks. In *ICASSP*, pages 7962–7966, 2013.

[131] X. Zhang, L. Wang, G. Li, F. Seide, and F. K. Soong. A new language independent, photo realistic talking head driven by voice only. In *Interspeech*, 2013.

[132] F. Zhou, J. Brandt, and Z. Lin. Exemplar-based graph matching for robust facial landmark localization. In *ICCV*, 2013.

[133] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[134] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[135] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886, 2012.

[136] K. Zuiderveld. Contrast limited adaptive histogram equalization. *Graphic Gems IV*, pages 474–485, 1994.