

©Copyright 2019

Deepali Aneja



# Learning-Based Techniques for Facial Animation

Deepali Aneja

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

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Linda Shapiro, Chair

Alex Colburn

Wilmot Li

Program Authorized to Offer Degree:  
Paul G. Allen School of Computer Science and Engineering



University of Washington

**Abstract**

Learning-Based Techniques for Facial Animation

Deepali Aneja

Chair of the Supervisory Committee:

Professor Linda Shapiro

Paul G. Allen School of Computer Science and Engineering

For decades, animation has been a popular storytelling technique. Traditional tools for creating animations are labor-intensive, requiring animators to painstakingly draw frames and motion curves by hand. An alternative workflow is to equip animators with direct real-time control over digital characters via performance, which offers a more immediate and efficient way to create animation. Even when using these existing expression transfer and lip sync methods, producing convincing facial animation in real-time is a challenging task. In this work, we present several deep learning techniques to model and automate the process of perceptually valid expression retargeting from humans to characters, real-time lip sync for animation, and building an emotionally aware embodied conversational agent. We also present the findings from user studies and some promising future directions in this domain.



# TABLE OF CONTENTS

	Page
List of Figures . . . . .	iv
List of Tables . . . . .	ix
Chapter 1: Introduction . . . . .	1
1.1 Contributions . . . . .	2
1.2 Organization . . . . .	3
Chapter 2: Expression Retargeting in 2D . . . . .	5
2.1 Introduction . . . . .	5
2.2 Methodology . . . . .	10
2.3 Experimental Results . . . . .	19
2.4 Evaluation . . . . .	20
2.5 Comparison with Character Animator . . . . .	25
2.6 Summary . . . . .	26
Chapter 3: Expression Retargeting in 3D . . . . .	27
3.1 Introduction . . . . .	27
3.2 Data Acquisition and Pre-processing . . . . .	32
3.3 Methodology . . . . .	33
3.4 Results . . . . .	39
3.5 Summary . . . . .	46
Chapter 4: Expression Retargeting on Stylized Animal Characters . . . . .	47
4.1 Introduction . . . . .	47
4.2 Data Collection . . . . .	49
4.3 Methodology . . . . .	50

4.4	Results . . . . .	53
4.5	Summary . . . . .	56
Chapter 5: Lip Sync for Live 2D Animation . . . . .		57
5.1	Introduction . . . . .	57
5.2	Approach . . . . .	62
5.3	Experiments . . . . .	70
5.4	Results . . . . .	73
5.5	Limitations . . . . .	80
5.6	Summary . . . . .	80
Chapter 6: Conversational Style Matching Agent . . . . .		81
6.1	Introduction . . . . .	81
6.2	Hypotheses . . . . .	86
6.3	Intelligent Agent Design . . . . .	87
6.4	Conversational Style Matching . . . . .	90
6.5	User Study . . . . .	93
6.6	Results . . . . .	97
6.7	Discussion . . . . .	98
6.8	Design Guidelines . . . . .	101
6.9	Summary . . . . .	102
Chapter 7: High-Fidelity Open Embodied Avatar . . . . .		103
7.1	Introduction . . . . .	103
7.2	The Avatar . . . . .	105
7.3	Applications . . . . .	108
7.4	Limitations . . . . .	111
7.5	Access and Citation . . . . .	111
7.6	Summary . . . . .	112
Chapter 8: Socially Intelligent Virtual Agent . . . . .		113
8.1	Introduction . . . . .	113
8.2	SIVA - The Agent . . . . .	117
8.3	Style Matching and Synthesis . . . . .	122

8.4	User Study	124
8.5	Results and Discussion	127
8.6	Limitations	131
8.7	Summary	131
Chapter 9:	Conclusion	133
	Bibliography	136



## LIST OF FIGURES

Figure Number	Page
2.1 Expressions are surprisingly difficult to create for professional animators. Three professional animators were asked to make the character appear as surprised as possible. None of the expressions achieved above 50% recognition on Mechanical Turk with 50 test subjects. . . . .	6
2.2 Overview of our pipeline. Feature extraction using CNNs and transfer learning builds a model of expression mapping. . . . .	7
2.3 DeepExpr yields clearer expressions than other approaches when tested on MT. From left to right each generated face was intended to clearly convey an expression: Anger from MPEG-4 scored 20% clarity for anger. Anger from HapFACS scored 8% clarity for anger. Fear from HapFACS scored 20% clarity for fear. Fear using FACSGen scored 6% clarity for fear. Anger and fear faces retrieved with our approach, both scored over 85% clarity. . . . .	8
2.4 Facial landmark detection and geometric feature extraction. (a) Face detection on a human input image. (b) Detected facial key points shown as blue dots from a state-of-the-art face alignment method alongwith the defined geometric measurements. . . . .	12
2.5 Examples of registered faces from CK+, DISFA, KDEF, and MMI databases showing disgust, joy, anger, and surprise emotion from left to right. . . . .	13
2.6 Outline of the CNN architecture. The convolutional layers, max pooling layers and fully connected layers are denoted as CONV, POOL and FC followed by the layer number. Human expression image trained model (left), Stylized character expression image trained model (middle) and fine-tuned character trained model (right) are shown. In the transfer learning step, the last fully-connected layer (FC7_character) is fine-tuned using stylized character data. . . . .	15
2.7 Multiple retrieval results for the joy query image . . . . .	18
2.8 Results from our combined approach - DeepExpr and Geometric features. Left-most image in each row is the query image and all six characters are shown portraying the top match of the same expression - anger, fear, joy, disgust, neutral, sad and surprise (top to bottom). . . . .	19

2.9	Character-to-character (top row) and human-to-human (bottom row) expression transfer results. The character expression has been recognized as “surprise” by the character-expression-trained network and the human expression was recognized as “Sad” by the human-expression-trained network. . . . .	20
2.10	Best match results from our DeepExpr approach compared to only geometric feature based retrieval for Disgust (top) and Fear (bottom). . . . .	21
2.11	Average (%) confusion matrix for human expression recognition for six cardinal expressions and neutral. . . . .	22
2.12	Comparison of similarity ranking results: Human expert ranking (top row), MT subjects ranking (middle ranking) and our DeepExpr method (bottom row). Note that the best rank of our DeepExpr method matches the MTurk ranking. . . . .	23
2.13	Correlation rank order result charts with (a) expert and (b) MT tests . . . . .	24
2.14	Expression matching results for fear query image. Ch result scored 41% clarity for fear and DeepExpr result scored 83% clarity for fear. . . . .	26
3.1	Example of inaccurate expression transfer. (a) Expression transfer from human (top right) to a character using Faceshift Studio. (b) Mechanical Turk testers perceive the human expression as sadness, while the character expression is perceived as neutral and a mixture of other expressions. The character expression has neither expression clarity nor geometric consistency. . . . .	28
3.2	Overview of our multi-stage deep learning system ExprGen which takes 2D images of human facial expressions as input and generates 3D parameters of stylized character rigs at runtime. See Section 3.2 for details of the dataset and preprocessing and Section 3.3 for details of training the 3D-CNN and Character Multi-Layer Perceptron (C-MLP). . . . .	29
3.3	Two new characters added to FERF-DB - ‘Mathilda’(left) and ‘Tuna’ (right) . . . .	30
3.4	The HCNN and SCNN are fused together to form the f-CNN, which is trained to produce a similarity score between human and primary character expressions. . . .	35
3.5	Comparison of best matches for training the f-CNN based on geometry and perception. (a) Human input (E), (b) Positive match (E+), (c) Negative match with incorrect geometry (E-), (d) Negative match with incorrect perception (E-). . . . .	36
3.6	Human to primary character expression transfer for human expression transition from neutral to joy, from neutral to surprise, from neutral to sadness, and from neutral to anger based on both perceptual and geometric similarity. (a) Human input expression frames (1-12), (b) Mapped expressions on ‘Mery’, and (c) Mapped expressions on ‘Ray’, (d) Expression recognition results between human (solid lines) and transferred expressions on ‘Mery’ (dashed lines) for different expressions. . . . .	41

3.7	Confusion matrix for perceived transferred expression recognition (%) for seven expression classes. . . . .	42
3.8	Consistent human expression transfer to primary characters. This example shows three different humans with the same expression (joy) transferred to the primary character ‘Aia’. . . . .	43
3.9	Quantitative comparison of expression transfer results of Faceware (blue bars) and ExprGen (red bars). . . . .	43
3.10	Qualitative comparison of expression transfer results of Faceware and ExprGen (left to right: input human expression, Faceware output and ExprGen output). . . . .	44
3.11	Primary to Secondary character expression transfer results (left to right: anger, disgust, fear, joy, sadness and surprise). (a) ‘Mery’s’ expression classes, Expressions transferred to (b) ‘Bonnie’, (c) ‘Tuna’, and (d) ‘Cody’. . . . .	45
3.12	Error cases in obtaining training examples for new secondary characters. (a) Matching is perceptually valid (both expressions are disgust) but geometrically incorrect, (b) Matching is perceptually invalid (expression on left is fear and on right is surprise) but geometrically correct. . . . .	46
4.1	Overview of our expression retargeting which takes 2D images of human facial expressions (left), generates expressions on the universal rig [6] (middle) and re-targets the expression to drive a 3D stylized animal character. . . . .	47
4.2	Hugh’s cardinal expression from left to right: anger, disgust, fear, joy, sadness and surprise. . . . .	49
4.3	Expression retargeting pipeline. First, we learn a low-dimensionality representation of Hugh expression space, and then learn the shared latent space between Hugh and the character using the keyposes. Finally, we apply the Hugh-to-character mapping to generate expression transfer results. . . . .	50
4.4	Comparison of Latent Dimension relevance learned by PCA and BGPLVM: (a) PCA relevance given by eigenvalues, (b) BGPLVM relevance given by ARD kernel weights. . . . .	54
4.5	The ARD Kernel weights learned for each observation space when MRD model trained for Hugh and Cody expression spaces. . . . .	55
4.6	Expression retargeting results on Cody and Noodle. . . . .	56
5.1	Real-Time Lip Sync. Our deep learning approach uses an LSTM to convert live streaming audio to discrete visemes for 2D characters. . . . .	58
5.2	Chloe’s Viseme Set. Additional associated sounds in parentheses. . . . .	62

5.3	Lip Sync Model. We use a unidirectional single-layer LSTM with a temporal shift $d$ of 6 feature vectors (60ms) (a). The audio feature $a$ consists of MFCC, log mean energy, and their first temporal derivatives (b). . . . .	64
5.4	Data Augmentation. Each reference recording has an associated hand-animated viseme sequence. We automatically time warp other recordings of the same sentence to align with each reference recording (a). This procedure allows us to create new input-output training pairs for every successfully warped recording. . . . .	68
5.5	Analysis of Lip Sync Styles. Histograms of viseme usage (a) and raw transition counts (b) show that different animators prefer different visemes and aim for different levels of articulation. . . . .	71
5.6	Impact of Data Augmentation. Augmenting the data results in a significant increase in accuracy, with diminishing returns after four augmented batches. . . . .	73
5.7	Characters. We used Chloe and the Wizard for our human judgement experiments, and we show lip sync results with the other characters in our video summary and supplemental materials. . . . .	74
5.8	Human Judgements. Our method was significantly preferred over all commercial tools, including offline methods. Our full model was also preferred over versions trained with no augmented data ( <i>OursNoAug</i> ) and two thirds of the augmented data ( <i>Ours2/3</i> ). However, the preference over <i>Ours2/3</i> was quite small, which suggests that this amount of data may be sufficient to train an effective model. . . . .	76
5.9	The rankings from our user study organized by system. The thick grey middle line is the median, the box represents 50% of the data between the first and third quartiles, and the whiskers mark the extremes. . . . .	78
6.1	Architecture of our dialogue agent. The microphone input is first processed to extract voiced segments. The voiced segments are then passed to neural speech recognition and voice prosody recognition. The text output from the speech recognition was input to a neural dialogue generation model and also to the LUIS intent recognition engine. The recognized speech and generated dialogue response were both sent to the conversational style extractor. Finally, a rule-based dialogue manager selected the utterance to synthesize. . . . .	85
6.2	Transcript of interactions between users and the agent for 3 different tasks. Emphasized responses were scripted responses based on detected intents of the user. . . . .	89
6.3	Setup for the user study . . . . .	94
6.4	Left: Rating of trust in the agent for all subjects ( $N_{control} = 15$ , $N_{matching} = 15$ ). Middle: Rating of trust in agent for high consideration subjects ( $N_c = 5$ , $N_m = 9$ ). Right: Rating of trust in agent for HC subjects ( $N_c = 9$ , $N_m = 6$ ). . . . .	96

6.5	A mildly significant interaction effect ( $p=0.079$ ) was observed between conversational style and the experimental condition. However, this was driven almost entirely by the high consideration participants (see Figure 6.4)	99
7.1	We present an open embodied avatar with lip syncing and expression capabilities that can be controlled via simple python interface. We provide examples of how to combine this with publicly available speech and dialogue APIs to construct a conversational embodied agent.	105
7.2	The locations of controllable bone positions on the avatar. Each position can be controlled with 6 degrees of freedom. See Table 7.1 for descriptions.	106
7.3	We present two pipelines for retargeting an expression from a human to the avatar. A) Expression Synthesis via Action Units Controls. Our pipeline takes a human video as an input, recognizes 12 FACS Action Units from the user’s detected face, and synthesizes the same expression on the avatar’s face. B) Expression Synthesis via Bone Position Controls. Our pipeline takes a human video as an input, detects the user’s face, generates 3D parameters on the primary character ‘Ray’ through a 3D-CNN, and synthesizes the same expression on the avatar’s face. See Section 3.2 for details for the Action Unit Recognizer, Expression Synthesizer, 3D-CNN and Character Multi-layer Perceptron (C-MLP).	109
8.1	SIVA’s Architecture: The system uses a fusion of both microphone and webcam inputs to drive the agent. Audio is segmented based on voice activity detection, and then speech and prosody recognition are performed. Generated dialogue is passed on to the Conversational Style manager, and along with the prosody style variables, conversational style aligned speech is synthesized, which is used for recognizing phonemes. The video pipeline detects the face to perform expression recognition, and head pose tracking. The output from the text sentiment recognizer, along with emotional expression probabilities, are used to generate expressions. Finally, the outputs from both the pipelines are composited together to synthesize SIVA’s expressive face and lip sync.	118
8.2	(a) SIVA’s non-verbal behavior and emotional expression plot across the user study conditions (high numbers mean a better rating). Interaction plot for 2-way ANOVA result for (b) Animacy. * shows statistical significance and (c) Anthropomorphism. The interaction between the Conversational Style and User Study Condition is significant for Animacy. In all figures, * shows statistical significance.	127

## LIST OF TABLES

Table Number	Page
2.1 Human Network configuration . . . . .	14
2.2 Average retrieval score for each expression across all characters using only geometry and DeepExpr features. . . . .	21
2.3 Average expression recognition accuracy (%) for each expression across all characters using Ch and DeepExpr. . . . .	26
3.1 Network Configuration . . . . .	34
3.2 Average (%) expression recognition accuracy for 2D images of human and stylized character expressions compared with the ground truth labels respectively. Note that the characters have higher expression clarity than humans due to their simpler geometry. . . . .	40
4.1 Reconstruction error for PCA AND BGPLVM models for High expression space. .	55
6.1 Summary of Participation . . . . .	93
7.1 The avatar has the bone position, facial action unit, and phoneme controls. M=middle, L=left, R=right, U=upper, L=lower, C=corner, I=inner, O=outer . . . . .	107
8.1 SIVA's behavior is controlled depending on whether the user was speaking or SIVA is speaking. . . . .	120
8.2 SIVA's expressions and the action unit (AU) combinations to create these expressions.	123

## ACKNOWLEDGMENTS

I would like to thank my advisor Linda Shapiro for her encouragement, guidance, and advice throughout my doctoral studies. I am really thankful for the trust and freedom she has given me while supporting my independence. Additionally, I would like to thank the members of my reading committee, Alex Colburn, with whom I collaborated on many projects and received valuable feedback, and Wilmot Li, who mentored me during my internship at Adobe Research and has always inspired me with his enthusiasm for research, insight, and knowledge of the field. I am also very grateful to Steven Tanimoto, for serving on my dissertation committee, and John Kramlich for serving as the graduate school representative in my exams.

I would like to express sincere appreciation to my collaborators Daniel McDuff and Mary Czerwinski for giving me a wonderful opportunity to intern at Microsoft Research and bringing a real-world perspective to my work, and to Barbara Mones and Gary Faigin for their vital feedback and invaluable creative inputs towards my research.

I owe thanks to my co-authors Siddharth Vora, Bindita Chaudhuri, Beibin Li, Sachin Mehta, and Rens Hoegen, and all the animators for their contributions towards my research. Bouncing ideas back and forth with you all has been the most enjoyable part of my research.

A big thank you to all past and present members of the GRAIL and Multimedia groups, especially to Ezgi Mercan, Bilge Soran, Shu Liang, Yao Lu, Sara Rolfe, Yuguang Li, Shima Nofallah, Meredith Wu and Nicholas Nuechterlein for providing a highly collaborative atmosphere around the lab. I am thankful to my dear friends Srinivasan Iyer, Koosha Khalvati, Maaz Ahmad, Shrainik Jain, Ravi Karkar, Arushi Prakash, and Eunice Jun for making grad school so much fun. I would also like to thank all of the CSE community for making this department such an excellent place to pursue a Ph.D., especially to Elise DeGoede who was extremely helpful.

I would like to express my gratitude to the organizations that have generously funded my research. In particular, I thank the Adobe Research, Microsoft Research, the Facial Expressions Research Group, and the UW Reality Lab.

I would like to take this opportunity to thank my close friends Vasudha Vij, Christopher Norwood, Elizabeth Hebener, Vignesh Chandrasekhar, Shruti Gangakhedkar, Devashish Bhatia and Shradha Nayyar for always cheering for me.

I am deeply grateful for having two sets of parents and siblings, one from birth and one from my marriage. They have showered me with unconditional love and support throughout this journey. Thanks to Swaraj Aneja and Sujata Nijhawan for being the strongest women that I know and always inspiring me to do my best work. Thanks to Madan Lal Aneja and Rajesh Kumar Nijhawan for teaching me to never give up and making sure that I had a constant supply of snacks from India at my desk. Thanks to Vikrant Aneja, Deepa Aneja and Surabhi Nijhawan for always being there.

Special thanks to my husband, Aman Nijhawan, who has been my pillar of strength and a constant source of motivation. It would not have been possible without you.

## Chapter 1

### INTRODUCTION

Animation is one of the most pervasive media for content creation today, and its applications include many areas such as films, gaming, education, healthcare, virtual/augmented reality, communication, and online marketing. Traditionally, creating animation has been a compute and labor-intensive process. This is especially true for intricate parts such as facial animation. However, with recent advances in performance-based movement capture tools, animation has transitioned from appearing in pre-rendered content to being created in real-time. This has enabled a myriad variety of use cases for using animated avatars in real-time communication. In this dissertation, we focus on enhancing these performance-based facial expression capture tools by creating building blocks to generate engaging facial animations. We validate our methodology by building an expressive virtual agent and evaluate its interactions with the users.

Existing performance-based facial expression capture tools [94, 95] often lack the expressive quality and perceptual validity of animator-created animations, mainly due to their assumption that geometric markers are sufficient for expression transfer. Moreover, it is challenging to use these tools for expression retargeting on stylized characters that go beyond human facial geometry. Engaging facial animation must have perceptually valid expressions to connect with the audience, where a *perceptually valid* expression is a character’s expression that is clearly perceived by humans to be the expression the animator or actor intended to convey.

While facial retargeting gives performers expressive control over the animation, another essential component of almost every live animation performance is speech. As a result, the other critical component of performance-to-animation mapping for live animation is lip sync — transforming an actor’s speech into corresponding mouth movements in the animated character. Convincing lip sync allows the character to embody the live performance, while poor lip sync breaks the illusion of

characters as live participants. Recent approaches [297, 74, 155, 288] have shown that vocal audio contains sufficient information to produce realistic speech animations. However, existing tools utilize learning-based algorithms that require a large amount of training data or offline preprocessing, which makes lip sync for live animation a challenging problem.

To validate the importance of perceptually valid expression retargeting, including lip sync, we explored adding expressiveness to an embodied conversational agent by enhancing its facial expression and lip sync capabilities. Our user study results indicate that the users' found the interactions with the expressive agent more engaging.

## **1.1 Contributions**

The primary contributions of the work are:

- A facial expression transfer approach for retrieving 2D character expressions corresponding to the human facial expressions using perceptual model mapping and human geometry.
- A novel multi-stage deep learning method to retarget human facial expressions to multiple stylized 3D characters.
- A facial expression retargeting approach to learn a shared latent space between a universal humanoid character and non-humanoid character parameters.
- A real-time interactive system to automatically generate live lip sync.
- An embodied conversational agent that can engage in a multi-turn dialogue and adapt to the conversational and expressive style of the user over the course of a conversation.
- An extensive user study to analyse human and expressive embodied conversational agent interactions.

## 1.2 Organization

The remainder of the thesis will cover the following material:

Chapter 2 introduces a novel expression transfer approach from humans to multiple stylized characters in 2D. We first train two Convolutional Neural Networks to recognize the expression of humans and stylized characters independently. Then we utilize a transfer learning technique to learn the mapping from humans to characters to create a shared embedding feature space. We use our perceptual model to retrieve character expressions corresponding to humans. This embedding also allows human expression-based image retrieval and character expression-based image retrieval.

Chapter 3 extends our 2D expression retrieval work to drive 3D stylized characters from human expressions. We present a system to automatically generate 3D stylized character expressions from humans in a perceptually valid and geometrically consistent manner. Our multi-stage deep learning system utilizes the latent variables of human and character expression recognition convolutional neural networks to control a 3D animated character rig. This end-to-end system takes images of human faces and generates the character rig parameters that best match the human's facial expression. Our approach generalizes to multiple characters and allows expression transfer between characters in a semi-supervised manner.

Chapter 4 introduces a universal humanoid character and facial expression retargeting approach that allows learning a high-quality mapping between the universal character and stylized animal characters. We address two main challenges of existing facial expression retargeting methods, the need for a large number of training examples, and going beyond the humanoid geometry for the target characters. Our system utilizes a Gaussian Process based nonlinear model to learn a shared latent space between the source and target expressions.

Chapter 5 presents a deep learning-based interactive system that automatically generates live lip sync for layered 2D characters using a Long Short Term Memory model. Our system takes streaming audio as input and produces viseme sequences with less than 200ms of latency (including processing time). Our contributions include specific design decisions for our feature definition and

model configuration that provide a small but useful amount of lookahead to produce accurate lip sync. We also describe a data augmentation procedure that allows us to achieve good results with a very small amount of hand-animated training data (13-20 minutes).

Chapter 6 presents an end-to-end voice-based conversational agent that can engage in naturalistic multi-turn dialogue and align with the interlocutor's conversational style. The system uses a series of deep neural network components for speech recognition, dialogue generation, prosodic analysis, and speech synthesis to generate language and prosodic expression with qualities that match those of the user.

Chapter 7 introduces an open embodied avatar built upon the Unreal Engine that can be controlled via a simple python programming interface. The avatar has lip syncing (phoneme control), head gesture, and facial expression (using either facial action units or cardinal emotion categories) capabilities.

Chapter 8 presents an expressive, embodied conversational agent that can recognize human behavior during open-ended conversations and automatically align its responses to the conversational and expressive style of the other party. The agent leverages multimodal inputs to produce rich and perceptually valid responses (facial expressions and lip sync) during the conversation.

Chapter 9 concludes the thesis by summarizing the main contributions and suggesting directions for future research.

## Chapter 2

# EXPRESSION RETARGETING IN 2D

### 2.1 Introduction

#### 2.1.1 Motivation

Facial expressions are an important component of almost all human interaction and face-to-face communication. As such, the importance of clear facial expressions in animated movies and illustrations cannot be overstated. Disney and Pixar animators [174, 251] have long understood that unambiguous expression of emotions helps convince an audience that an animated character has underlying cognitive processes.

The viewer’s emotional investment in a character depends on the clear recognition of the character’s emotional state [17]. To achieve lifelike emotional complexity, an animator must be able to depict characters with clear, unambiguous expressions, while retaining the fine level control over intensity and expression mix required for nuance and subtlety [245]. However, explicit expressions are notoriously difficult to create [4], as illustrated in Figure 2.1. This difficulty is in part due to animators and automatic systems relying on geometric markers and features modeled for human faces, not stylized character faces.

We focus our efforts on *stylized* 3D characters, defined as characters that no human would mistake for another person, but would still be perceived as having human emotions and thought processes. Our goal is to develop a model of facial expressions that enables accurate retrieval of stylized character expressions given a human expression query.

To achieve this goal, we created DeepExpr, a perceptual model of stylized characters that accurately recognizes human expressions and transfers them to a stylized character without relying on explicit geometric markers. Figure 2.2 shows an overview of the steps to develop the framework of our model. We created a database of labeled facial expressions for six stylized characters as shown

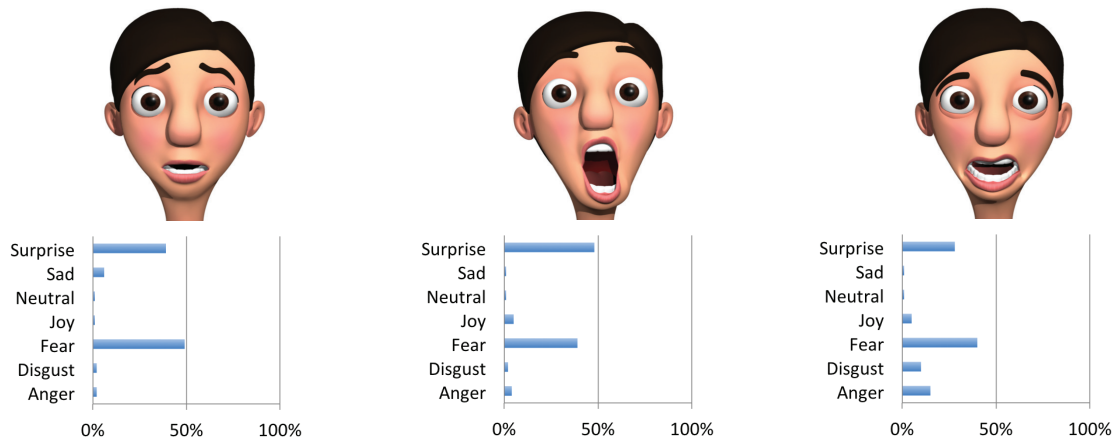


Figure 2.1: Expressions are surprisingly difficult to create for professional animators. Three professional animators were asked to make the character appear as surprised as possible. None of the expressions achieved above 50% recognition on Mechanical Turk with 50 test subjects.

in Figure 2.8. This database with expressions is created by facial expression artists and initially labeled via Mechanical Turk (MT) [33]. Images are labeled for each of six cardinal expressions: joy, sadness, anger, surprise, fear, disgust, and neutral. First, we trained a Convolutional Neural Network (CNN) on a large database of human expressions to input a human expression and output the probabilities of each of the seven classes. Second, we trained a similar character model on an artist-created character expression image database. Third, we learned a mapping between the human and character feature space using the transfer learning approach[230]. Finally, we can retrieve character expressions corresponding to a human using a perceptual model mapping and human geometry.

We make the following contributions:

- A data-driven perceptual model of facial expressions.
- A novel stylized character data set with cardinal expression annotations.
- A mechanism to accurately retrieve plausible character expressions from human expression queries.

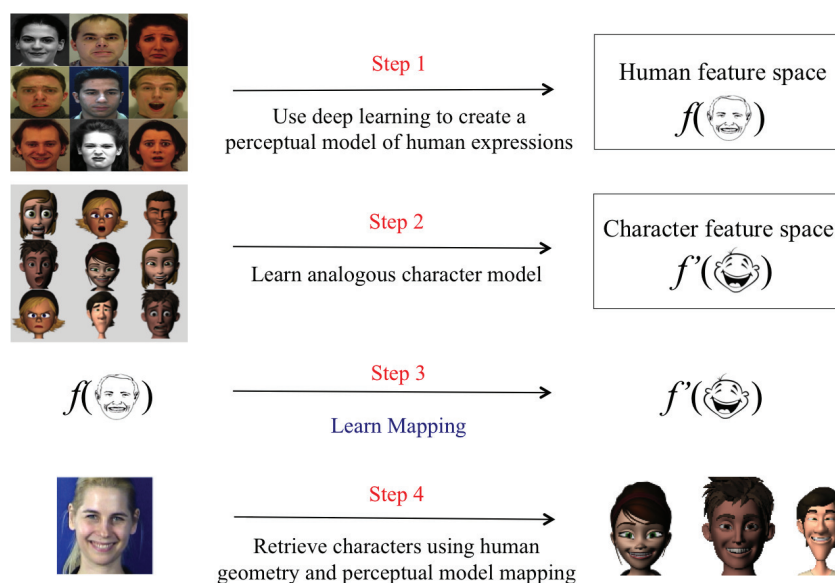


Figure 2.2: Overview of our pipeline. Feature extraction using CNNs and transfer learning builds a model of expression mapping.

Methods and results presented in this chapter were previously published in [6] and more details are provided on the project page: <http://grail.cs.washington.edu/projects/deepexpr/>

### 2.1.2 Related Literature

There is a large body of literature classifying, recognizing, and characterizing human facial expressions. Notably, Paul Ekman’s widely adopted Facial Action Coding System (FACS) [79] is used as a common basis for describing and communicating human facial expressions. The FACS system is often used as a basis for designing characters animation systems [4, 258] and facial expression recognition on scanned 3D faces [263]. However, despite these advances, creating clear facial animations for 3D characters remains a difficult task.

#### *Facial Expression Recognition and Perception*

Though a reliable parameterization of emotion and expression remains elusive, the six cardinal expressions pervade stories and face-to-face interactions, making them a suitable focus for educators

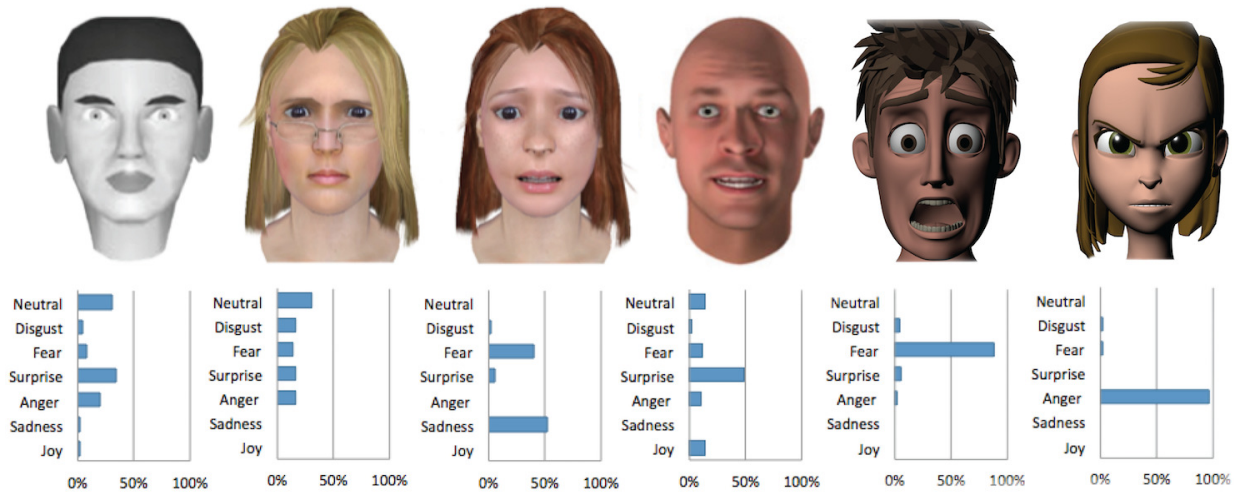


Figure 2.3: DeepExpr yields clearer expressions than other approaches when tested on MT. From left to right each generated face was intended to clearly convey an expression: Anger from MPEG-4 scored 20% clarity for anger. Anger from HapFACS scored 8% clarity for anger. Fear from HapFACS scored 20% clarity for fear. Fear using FACSGen scored 6% clarity for fear. Anger and fear faces retrieved with our approach, both scored over 85% clarity.

and facial expression researchers [1]. To guide and automate the process expression animation, animators and researchers turn to FACS. For example, FACSGen [258] allows researchers to control action units on realistic 3D synthetic faces. Though Roesch et al. confirmed the tool’s perceptual validity settings by asking viewers to rate the presence of emotions in faces developed using action unit combinations found in “real life situations”, we found that their faces were unclear as demonstrated in Figure 2.3.

HapFACS [4], an alternative to FACSGen, allows users to control facial movement at the level of both action units and whole expressions (EmFACS) according to Ekman’s formulas. The strict use of anatomy-based and constrained motion by these systems limits their generalizability to characters with different anatomy and limits their application, because the most believable animation may require the violation of physical laws [174]. Alternatively, the MPEG-4 standard [247] can describe motion in stylized faces by normalizing feature motion to a standard distance. The MPEG-4

standard provides users with archetypal expression profiles for the six cardinal expressions, but like FACS-based systems does not give the user feedback on the perceptual validity of their expression, which may lead to unclear faces. As shown in Figure 2.3, anatomically valid faces generated by these systems did not consistently yield high recognition rates in MT with 50 test subjects.

The results shown in Figure 2.3 support artists' intuition that anatomy-based formulas for expressions must be tailored to each unique face, and necessitate a perceptually guided system to find the optimal configuration for a clear expression. Perceptual models such as Deng and Ma [64] have also been explored for realistic faces with promising results. Deng and Ma polled students' perceptions of the expression of different motion-captured facial configurations and ran Principal Component Analysis (PCA) [150] on the vertices of the meshes of these faces. Using these results, they developed a Support Vector Machine (SVM) [58] model for expression clarity as a function of PCA weights for different areas of the face. They also showed significantly increased expression clarity of generated speech animation by constraining the characters' motion to fit their model. However, the scalability of their procedure is limited by its reliance on on-site subjects and the size and specificity of the seeding dataset. We addressed these limitations by incorporating MT tests in our character expression data collection and training a deep learning model for expression clarity.

### *Feature Extraction and Classification*

Facial expression recognition can be broadly categorized into face detection, registration, feature extraction, and classification. In the detection step, landmark points are used to detect a face in an image. In the registration step, the detected faces are geometrically aligned to match a template image. Then the registered image is used to extract numerical feature vectors as the part of the feature extraction step.

These features can be *geometry based* such as facial landmarks [166, 68], *appearance based* such as Local Binary Patterns (LBP) [278], Gabor filters [189], Haar features [321], Histogram of Oriented Gradients [280], or *motion based* such as optical flow [158] and Volume LBP [340]. Recently, methods have been developed to learn the features by using sparse representations [200, 188]. 3D shape model approach has also be implemented to improve the facial expression recog-

dition rate [144]. A variety of fusion of features has also been utilized to boost up the facial expression recognition performance [290, 329]. They are mostly a combination of geometric and appearance based features. In the current practice of facial expression analysis, CNNs have shown the capability to learn the features that statistically allow the network to make the correct classification of the input data in various ways [214, 334]. CNN features have been fused with geometric features for customized expression recognition [333], and Deep Belief Network have also been utilized to solve the Facial Expression recognition (FER) problem. A recent approach [190] termed “AU (Action Unit)-Aware” Deep Networks demonstrated the effectiveness in classifying the six basic expressions. Joint Fine-Tuning in Deep Neural Networks [151] have also been used to combine temporal appearance features from image sequences and temporal geometry features from temporal facial landmark points to enhance the performance of the facial expression recognition. Along similar lines, we have utilized deep learning techniques as a tool to extract useful features from raw data for both human faces and stylized characters. We then deploy a transfer learning approach, where the weights of the stylized character are initialized with those from a network pre-trained on a human face data set, and then fine-tune with the target stylized character dataset.

In the last step of classification, the algorithm attempts to classify the given face image into seven different classes of basic emotions using machine learning techniques. SVMs can be used for FER tasks [341, 278, 72]. As SVMs treat the outputs as scores for each class which are uncalibrated and difficult to interpret, the softmax classifier gives a slightly more intuitive output with normalized class probabilities and also has a probabilistic interpretation. Based on that, we have used a softmax classifier to recognize the expressions in our classification task using the features extracted by the deep CNNs.

## **2.2 Methodology**

We first describe the data collection approach and design of facial features that can capture the seven expressions: joy, sadness, anger, surprise, fear, disgust, and neutral. Then, we discuss our customized expression recognition and transfer learning framework using deep learning.

### 2.2.1 Data Collection and Pre-processing

To learn deep CNN models that generalize well across a wide range of expressions, we need sufficient training data to avoid over-fitting of the model. For human facial expression data collection, we combined publicly available annotated facial expression databases: extended CK+ [194], DISFA [207], KDEF [196] and MMI [234]. We also created a novel database of facial expressions for six stylized characters : Facial Expression Research Group - Database (**FERG-DB**). Both the databases have labels for the six cardinal expressions and neutral.

**CK+**: The Extended Cohn-Kanade database (CK+) includes 593 video sequences recorded from 123 subjects. Subjects portrayed the six cardinal expressions. We selected only the final frame of each sequence with the peak expression for our method, which resulted in 309 images.

**DISFA**: Denver Intensity of Spontaneous Facial Actions (DISFA) database consists of 27 subjects, each recorded while watching a four minutes video clip by two cameras. As DISFA is not emotion-specified coded, we used the EMFACS system [4] to convert AU FACS codes to expressions, which resulted in around 50,000 images using the left camera only.

**KDEF**: The Karolinska Directed Emotional Faces (KDEF) is a set of 4900 images of human facial expressions of emotion. This database consists of 70 individuals, each displaying 7 different emotional expressions. We used only the front facing angle for our method and selected 980 images.

**MMI**: The MMI database includes expression labeled videos for more than 20 subjects of both genders for which subjects were instructed to display 79 series of facial expressions. We extracted static frames from each corresponding sequence for the six cardinal emotions, resulting in 10,000 images.

We balanced out the final number of the samples for each class for training our network to avoid any bias towards a particular expression.

**Stylized Character Database** We created a novel database (**FERG-DB**) of labeled facial expressions for six stylized characters ('Mery', 'Aia', 'Bonnie', 'Jules', 'Malcolm' and 'Ray'). The

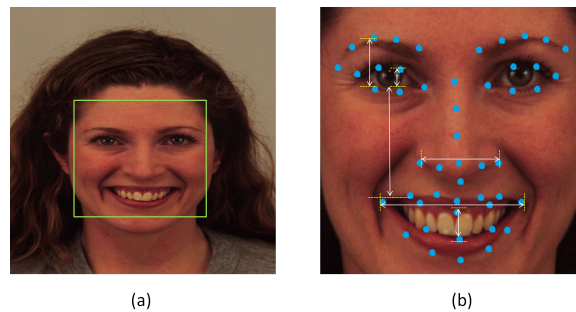


Figure 2.4: Facial landmark detection and geometric feature extraction. (a) Face detection on a human input image. (b) Detected facial key points shown as blue dots from a state-of-the-art face alignment method along with the defined geometric measurements.

animator created the key poses for each expression, and they were labeled via MT to populate the database initially. The number of key poses created depends on the complexity of the expression for each character. We only used the expression key poses having 70% MT test agreement among 50 test subjects for the same pose. On average, 150 key poses (15-20 per expression) were created for each character. Interpolating between the key poses resulted in 50,000 images (around 8,000 images per character). The motivation behind the combination of different characters is to have a generalized feature space among various stylized characters. We obtained the stylized 3D rigs modeled using the Autodesk®MAYA software for all the characters.

### *Data Pre-processing*

For our combined human dataset, we extract 49 facial landmarks [324] to register a face to an average frontal face via an affine transformation and use the landmarks to extract the geometric features including the following measurements: left/right eyebrow height (vertical distance between top of the eyebrow and center of the eye), left/right eyelid height (vertical distance between top of an eye and bottom of the eye), nose width (horizontal distance between leftmost and rightmost nose landmarks), mouth width (left mouth corner to right mouth corner distance), closed mouth measure (vertical distance between the upper and the lower lip), and left/right lip height (vertical distance

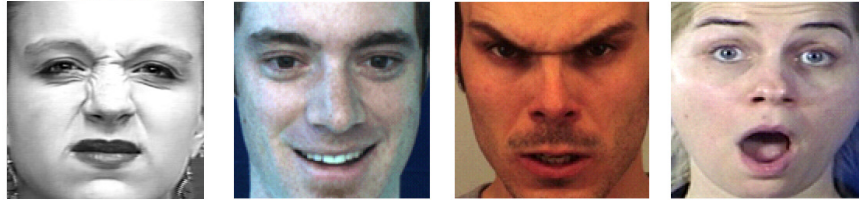


Figure 2.5: Examples of registered faces from CK+, DISFA, KDEF, and MMI databases showing disgust, joy, anger, and surprise emotion from left to right.

between the lip corner from the lower eyelid) as shown in Figure 2.4. Each distance is normalized by the bounding box of the face. Once the faces are cropped and registered, the images are re-sized to  $256 \times 256$  pixels for input to our network training. We use these points to register faces to an average frontal face via an affine transformation. Then a bounding box around the face is considered to be the face region. Geometric measurements between the points are also taken to produce geometric features for refinement of expression retrieval results as described in Section 2.2.2. Once the faces are cropped and registered, the images are re-sized to  $256 \times 256$  pixels for analysis. Figure 2.5 shows examples of registered faces from different databases using this method.

The corresponding 49 landmark points are marked on the neutral expression of the 3D stylized character rig. This supplementary information is saved along with each expression rendering and used later to perform geometric refinement of the result. This step is performed only once per character.

### 2.2.2 Network Training using Deep Learning

With approximately 70,000 images of labeled samples of human faces and 50,000 images for stylized character faces, the datasets are smaller in comparison to other image classification datasets that have been trained from scratch in the past. Moreover, since we have to use a portion of this data set for validation, effectively only 80% of the data was available for training. We performed data augmentation techniques to increase the number of training examples. This step helps in reducing overfitting and improving the model’s ability to generalize. During the training phase, we

Layer type	Patch size/Stride	Output
CONV1	7x7/4	56x56x96
POOL1	3x3/2	28x28x96
CONV2	5x5/1	28x28x256
POOL2	3x3/2	14x14x256
CONV3	3x3/1	14x14x384
POOL3	3x3/2	7x7x384
CONV4	3x3/1	7x7x384
POOL4	3x3/2	3x3x384
FC5		1x1x1024
FC6		1x1x512
FC7		1x1x7

Table 2.1: Human Network configuration

extracted 5 crops of  $227 \times 227$  from the four corners and the center of the image and also used the horizontal mirror images for data augmentation.

### *Training human and character CNN models*

Our human expression network consists of three elements: multiple convolutional layers followed by max-pooling layers and fully connected layers as in [168]. Our character network is analogous to the human CNN architecture and does not require CONV4 for the recognition task as the character images are not very complex. Unlike the human dataset, there are fewer variations in the character dataset (light, pose, accessories, etc.). To avoid overfitting, we limited our model to a fewer number of convolutional parameters (until CONV3). Both networks are trained independently. The details of the network layers are shown in Figure 2.6 and network parameters. Table 2.1 describes the configuration of each layer for a human expression trained network. Our character expression network configuration is analogous to the human expression network architecture, but is trained on the stylized character expression dataset and does not have the CONV4 layer.

All three color channels are processed directly by the network. Images are first rescaled to  $256 \times 256$  and a crop of  $227 \times 227$  is fed to the network. Finally, the output of the last fully connected

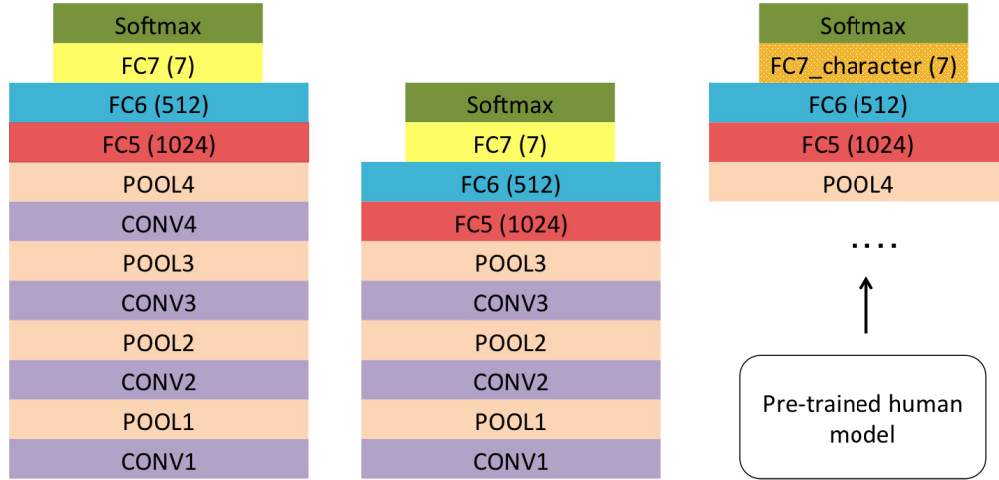


Figure 2.6: Outline of the CNN architecture. The convolutional layers, max pooling layers and fully connected layers are denoted as CONV, POOL and FC followed by the layer number. Human expression image trained model (left), Stylized character expression image trained model (middle) and fine-tuned character trained model (right) are shown. In the transfer learning step, the last fully-connected layer (FC7\_character) is fine-tuned using stylized character data.

layer is fed to a softmax layer that assigns a probability for each class. The prediction itself is made by taking the class with the maximal probability for the given test image.

In the forward propagation step, the CONV layer computes the output of neurons that are connected to local regions in the input (resized to  $256 \times 256$  in the data pre-processing step), each computing a dot product between their weights and a small region they are connected to in the input volume, while the POOL layer performs a downsampling operation along the spatial dimensions. The output of each layer is a linear combination of the inputs mapped by an activation function given as:

$$h^{i+1} = f((W^{i+1})^T h^i) \quad (2.1)$$

where  $h^{i+1}$  is the  $i^{th}$  layer output,  $W^i$  is the vector of weights that connect to each output node and  $f(\cdot)$  is the non-linear activation function which is implemented by the RELU layer given as:

$f(x) = \max(0, x)$  where  $x$  is the input to the neuron. The back-propagation algorithm is used to calculate the gradient with respect to the parameters of the model. The weights of each layer are updated as:

$$\delta^i = (W^i)^T \delta^{i+1} \cdot f'(h^i) \quad (2.2)$$

where  $\delta^i$  is the increment of weights at layer  $i$ . We train our networks using stochastic gradient descent with hyperparameters (momentum=0.9, weight decay=0.0005, initial learning rate=0.01). The learning rate is dropped by a factor of 10 following every 10 epochs of training. The proposed network architectures were implemented using the Caffe toolbox [145] on a Tesla k40c GPU.

### *Transfer Learning*

To create a shared embedding feature space, we fine-tuned the CNN pre-trained on the human dataset with the character dataset for every character by continuing the backpropagation step. The last fully connected layer of human trained model was fine-tuned, and earlier layers were kept fixed to avoid overfitting. We decreased the overall learning rate while increasing the learning rate on the newly initialized FC7\_character layer which is highlighted fine-tuned character trained model in Figure 2.6. We set an initial learning rate of 0.001, so that we don't drastically alter the pre-trained weights. The learning rate is dropped by a factor of 10 following every 10 epochs of training. Our fine-tuned model used 38K stylized character image samples for training, 6K for validation, and 6K for test. The proposed architecture was trained for 50 epochs with 40K iterations on batches of size 50 samples.

### *Distance Metrics*

In order to retrieve the stylized character closest expression match to the human expression, we used the Jensen—Shannon divergence distance [187] for expression clarity and geometric feature distance for expression refinement. It is described by minimizing the distance optimization func-

tion in eq. 2.3 given as:

$$\phi_d = \alpha |\text{JS Distance}| + \beta |\text{Geometric Distance}| \quad (2.3)$$

where JS Distance is given as the Jensen—Shannon divergence distance between FC6 feature vectors of *human* and *character*, and Geometric distance is given as the  $L^2$  norm distance between geometric features of *human* and *character*. Our implementation uses JS Distance as a retrieval parameter and then geometric distance as a sorting parameter to refine the retrieved results with  $\alpha$  and  $\beta$  as relative weight parameters. Details of the computation are given as follows:

**Expression Distance** For a given human expression query image, FC6 (512 outputs) features are extracted from the query image using the human expression trained model and for the test character images from the shared embedding feature space using the fine-tuned character expression model. The FC7 (7 outputs) layer followed by a softmax can be interpreted as the probability that a particular expression class is predicted for a given input feature vector. By normalizing each element of the feature vector by the softmax weight, the FC6 feature vectors are treated as discrete probability distributions. To measure the similarity between human and character feature probability distributions, we used the Jensen—Shannon divergence [187] which is symmetric and is computed as:

$$JSD(H||C) = \frac{1}{2}D(H||M) + \frac{1}{2}D(C||M) \quad (2.4)$$

where  $M = \frac{1}{2}(H + C)$ ,  $D(H||M)$  and  $D(C||M)$  represents the Kullback—Leibler divergence [170] which is given as:

$$D(X||M) = \sum_i X(i) \log \frac{X(i)}{M(i)} \quad (2.5)$$

where X and M are discrete probability distributions.

We used this distance metric to order the retrievals from the closest distance to the farthest in the expression feature space. Our results show that the retrieval ordering matched the query image label, and retrievals were ordered in order of similarity to the query label. To choose the

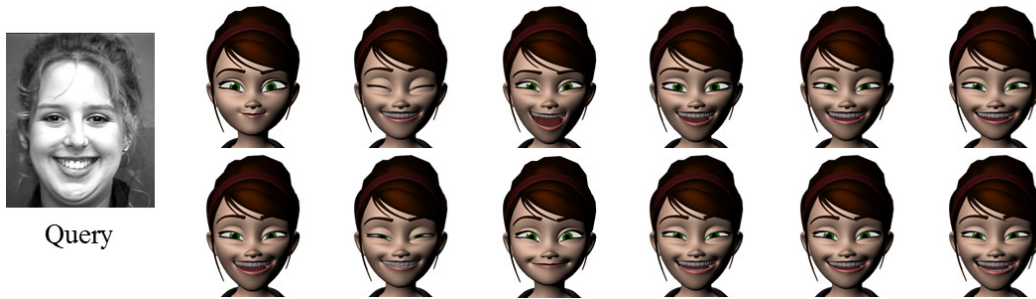


Figure 2.7: Multiple retrieval results for the joy query image

best match out of the multiple retrievals with the same label as shown in Figure 2.7, we added a geometric refinement step as described in the next section.

**Geometric Distance** The JS Divergence distance results in the correct expression match, but not always the closest geometric match to the expression. Figure 2.7 shows the retrieval of the correct label (joy). To match the geometry, we extract geometric distance vectors and use them to refine the result.

We use the facial landmarks as described in Section 2.2.1, to extract the geometric features including the following measurements: the left/right eyebrow height (vertical distance between top of the eyebrow and center of the eye), left/right eyelid height (vertical distance between top of an eye and bottom of the eye), nose width (horizontal distance between leftmost and rightmost nose landmarks), mouth width (left mouth corner to right mouth corner distance), closed mouth measure (vertical distance between the upper and the lower lip), and left/right lip height (vertical distance between the lip corner from respective the lower eyelid). The geometric distance is a normalized space. Each of the distances between landmarks is normalized by the bounding box of the face. After normalization, we compute the  $L^2$  norm distance between the human geometry vector and character geometry vectors with the correct expression label. Finally, we re-order the retrieved images within the matched label based on matched geometry.

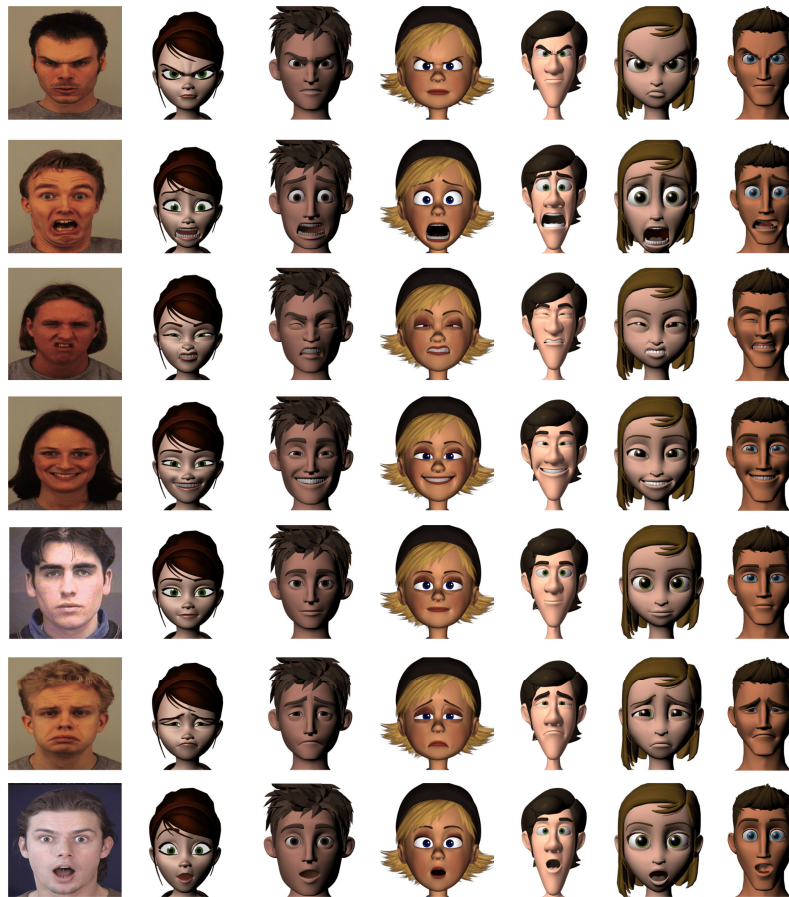


Figure 2.8: Results from our combined approach - DeepExpr and Geometric features. Leftmost image in each row is the query image and all six characters are shown portraying the top match of the same expression - anger, fear, joy, disgust, neutral, sad and surprise (top to bottom).

### 2.3 Experimental Results

The combined DeepExpr features and geometric features produce significant performance enhancement in retrieving the stylized character facial expressions based on human facial expressions. The top results for all seven expressions on six stylized characters are shown in Figure 2.8.

Our character expression trained model can be used to classify and transfer similar expression on different characters. The results for the “surprise” expression transferred to different characters are shown in Figure 2.9 (Top row). This technique can be utilized to transfer the storyboard to a

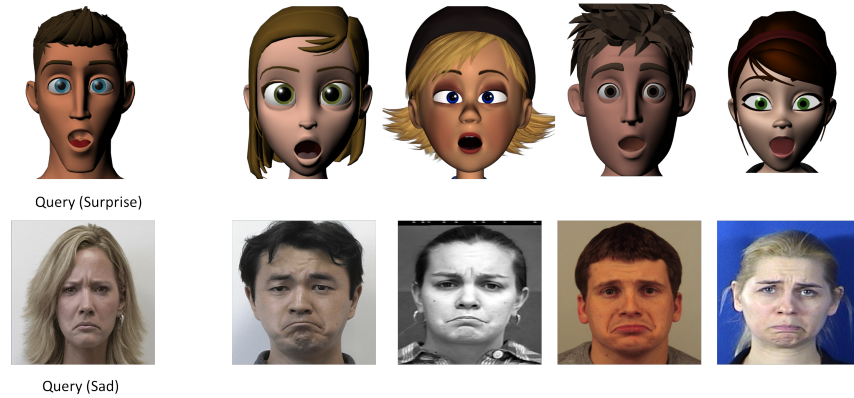


Figure 2.9: Character-to-character (top row) and human-to-human (bottom row) expression transfer results. The character expression has been recognized as “surprise” by the character-expression-trained network and the human expression was recognized as “Sad” by the human-expression-trained network.

different character.

Analogous to character-to-character expression transfer, human expressions can be recognized and used to retrieve the same expression in the human dataset as shown in Figure 2.9 (Bottom row).

## 2.4 Evaluation

### 2.4.1 Expression Recognition Accuracy

For Human Facial Expression Recognition accuracy, we performed a subject independent evaluation, where the classifier is trained on the training set and evaluated on images in the same database (validation and test set) using K-fold cross-validation with  $K=5$ . On average, we used 56K samples for training in batches of 50 samples, 10K samples for validation and 10K for testing. The overall accuracy of human facial expression recognition was 85.27%. In the current practice of human facial expression analysis, CNNs have been deployed to learn the features and perform expression classification [214, 334, 151]. Our models are trained from scratch and our results are comparable to state-of-the-art classification results as represented by the average (%) confusion matrix

Expression	Geometry	DeepExpr
Anger	0.384	0.213
Disgust	0.386	0.171
Fear	0.419	0.228
Joy	0.276	0.106
Neutral	0.429	0.314
Sad	0.271	0.149
Surprise	0.322	0.125

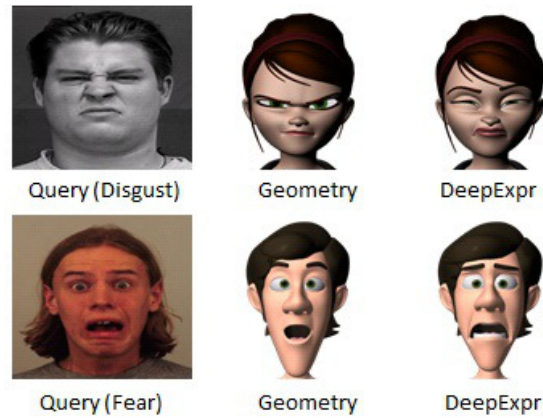


Table 2.2: Average retrieval score for each expression across all characters using only geometry and DeepExpr features. Figure 2.10: Best match results from our DeepExpr approach compared to only geometric feature based retrieval for Disgust (top) and Fear (bottom).

in Figure 2.11. In our human-to-character expression retrieval task, the similarity in human and character feature spaces in the mapping step is given more importance than the actual expression recognition problem. Both of our models (human and character) were trained to achieve an overall expression recognition accuracy is  $> 85\%$  such that the independent expression feature spaces are comparable.

Similarly, for stylized character expression, we used 38K character images for training in batches of 50 samples, 6K for validation, and 6K for testing, and achieved the recognition accuracy of 89.02%. Our aim with human expression accuracy was to achieve a good score on the expression recognition which is close to the state-of-the-art results in order to extract relevant features corresponding to a facial expression.

#### 2.4.2 Expression Retrieval Accuracy

We analyze our retrieval results by computing the retrieval score to measure how close is the retrieved character expression label is to the human query expression label. We also compare our

		Predicted Expression						
		Anger	Disgust	Fear	Joy	Neutral	Sad	Surprise
Actual Expression	Anger	61.6	11.3	7.8	1.9	5.3	7.2	4.6
	Disgust	3.6	82.4	1.5	2.7	6.2	3.1	0.3
	Fear	6.2	5.1	51	7.3	16.2	4.6	9.3
	Joy	0.4	1.8	1.5	87.2	7.4	0.3	1.2
	Neutral	1.3	4.3	8.3	5.5	78.2	1.3	0.9
	Sad	4.2	7.1	5.4	1.6	6.8	73.2	1.5
	Surprise	0.6	0.2	3.1	2.8	1.2	0.3	91.5

Figure 2.11: Average (%) confusion matrix for human expression recognition for six cardinal expressions and neutral.

results with a facial expression expert by choosing 5 random samples from the retrieved results with the same label and rank order them based on their similarity to the query image. The details of analysis are discussed as follows:

### Retrieval Score

We measured the retrieval performance of our method by calculating the average normalized rank of relevant results (same expression label) [222]. The evaluation score for a query human expression image was calculated as:

$$score(q) = \frac{1}{1 - N \cdot N_{rel}} \left( \sum_{k=1}^{N_{rel}} R_k - \frac{N_{rel}(N_{rel} + 1)}{2} \right) \quad (2.6)$$

where  $N$  is the number of images in the database,  $N_{rel}$  the number of database images that are relevant to the query expression label  $q$  (all images in the character database that have the same expression label as the human query expression label), and  $R_k$  is the rank assigned to the  $k^{th}$  relevant



Figure 2.12: Comparison of similarity ranking results: Human expert ranking (top row), MT subjects ranking (middle ranking) and our DeepExpr method (bottom row). Note that the best rank of our DeepExpr method matches the MTurk ranking.

image. The evaluation score ranges from 0 to 1, where 0 is the best score as it indicates that all the relevant database images are retrieved before all other images in the database. A score that is  $>0$  denotes that some irrelevant images (false positives) are retrieved before all relevant images.

The retrieval performance was measured over all the images in the human test dataset using each test image in turn as a query image. The average retrieval score for each expression class was calculated by averaging the retrieval score for all test images in the same class. Table 2.2 shows the final class retrieval score which was calculated by averaging the retrieval scores across all characters for each expression class using only geometry and DeepExpr expression features. The best match results in Figure 2.10 confirm that the geometry measure is not sufficient to match the human query expression with clarity.

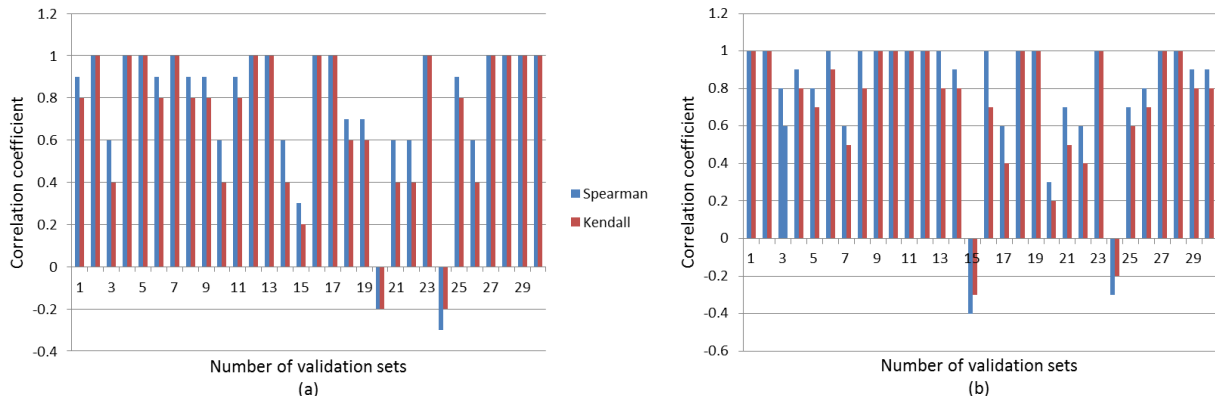


Figure 2.13: Correlation rank order result charts with (a) expert and (b) MT tests

### Comparison

In order to judge the effectiveness of our system, we compared DeepExpr to a human expert and MT test subjects. We asked the expert and the MT subjects to rank five stylized character expressions in order of decreasing expression similarity to a human query image. The facial expression expert, 50 MT test subjects and DeepExpr ranked the same 30 validation test sets. We aggregated the MT results into a single ranking using a voting scheme. We then compared DeepExpr ranking to the results, measuring similarity with two measures. We asked the facial expression expert and the MT subjects to rank five stylized file character expressions in order of decreasing similarity to a human query image. Both measures found a high correlation between DeepExpr ranking compared with the expert and the MT ranking results. Figure 2.12 shows the validation test results from the expert ranking (top row), most probable MTurk ranking (middle row) and our DeepExpr ranking (bottom row) for a “Sad” expression human query corresponding to validation experiment 20<sup>th</sup> in the main text. There is a negative Spearman correlation (-0.2) with the expert but a positive Spearman correlation with the MTurk test subjects (0.3). As there are subtle differences between expressions within the same class (“Sad”), hence there is some confusion in ranking order. In this example, the best rank matches the MTurk Test results but does not match with the expert ranking.

The **Spearman rank correlation coefficient**  $\rho$  measures the strength and direction of the association between two ranked variables [285]. The closer the  $\rho$  coefficient is to 1, the better the

two ranks are correlated. The average  $\rho$  coefficient for the expert rank orderings is  $0.773 \pm 0.336$  and MT tests is  $0.793 \pm 0.3561$ . The most relevant correlation coefficient is between the first rank chosen by the expert and the first rank chosen by DeepExpr as they represent the best match with the query image. The Spearman correlation with expert best rank is 0.934 and with MT best rank is 0.942, which confirms the agreement on selection of the closest match to the human expression.

The **Kendall  $\tau$  test** is a non-parametric hypothesis test for statistical dependence based on the  $\tau$  coefficient [157]. It is a pairwise error that represents how many pairs are ranked discordant. The best matching ranks receive a  $\tau$  value of 1. The average  $\tau$  coefficient for expert validation rank orderings is  $0.706 \pm 0.355$ , and the best rank correlation is 0.910. For the MT ranking, the average Kendall correlation coefficient is  $0.716 \pm 0.343$  and 0.927 is the best rank correlation.

The Spearman and Kendall correlation coefficients of DeepExpr ranking with the expert ranking and MT test ranking for 30 validation experiments are shown in Figure 2.13. Note that more than half the rankings are perfectly correlated, and most of them are above 0.8. Only two of the rankings had (small) negative correlations in both correlation experiments: the order was confusing because of very subtle difference in expressions.

## 2.5 Comparison with Character Animator

Currently to our knowledge, no other system performs stylized character retrieval based on a learned feature set. The closest match to DeepExpr tool is Adobe Character Animator (Ch) [49] which creates 2D animation for characters. We conducted an expression recognition experiment by creating a similar character in Ch with different expressions as layers. We queried three human expression images for each of the seven expressions and asked 50 MT test subjects to recognize the expression for best matches from DeepExpr retrieved images and Ch results. The results are shown in Table 2.3. On an average, joy, neutral and surprise had comparable recognition performance. DeepExpr showed great improvement in recognition of fear and disgust. In Ch, fear was confused with surprise due to the dependence on geometric landmarks of the face showing an open mouth and disgust was most confused with anger. For anger and sad, the closed mouth was most confused

Expression	Ch	DeepExpr
Anger	60	85
Disgust	47	86
Fear	42	81
Neutral	87	88
Joy	95	97
Sad	43	89
Surprise	93	95

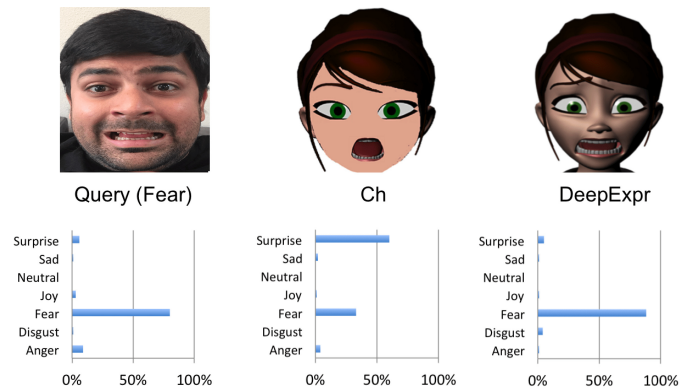


Table 2.3: Average expression recognition accuracy (%) for each expression across all characters using Ch and DeepExpr. Figure 2.14: Expression matching results for fear query image. Ch result scored 41% clarity for fear and DeepExpr result scored 83% clarity for fear.

with neutral in Ch. An example of fear expression MT test is shown in Figure 2.14. DeepExpr achieved higher (83%) expression recognition accuracy as compared to Ch (41%).

## 2.6 Summary

We have demonstrated a perceptual model of facial expression clarity and geometry using a deep learning approach combined with artistic input and crowdsourced perceptual data. Our results are highly correlated with a facial expression expert, in addition to MT subjects and have a higher expression recognition accuracy as compared to Character Animator.

## Chapter 3

# EXPRESSION RETARGETING IN 3D

### 3.1 Introduction

#### 3.1.1 Motivation

Our work is motivated by the goal of enhancing animated storytelling by improving 3D stylized character facial expressions. The importance of believable and accurate animated character facial expressions is readily demonstrated by films and games such as *Polar Express* [338] and *Mass Effect: Andromeda*<sup>1</sup>. In these examples, it is difficult for the audience to connect to the characters and broader storyline due to the character’s facial expressions. The characters in these examples do not exhibit clearly recognizable facial expressions that are consistent with their emotional state in the storyline. Effective characters must have perceptually valid expressions, where a *perceptually valid* expression is a character’s expression that is clearly perceived by humans to be in the intended expression class. Figure 3.1 shows a concrete example of a perceptually invalid expression, in which the human expression did not transfer correctly to the character when tested on Mechanical Turk (MT) for expression clarity with 30 test subjects.

Animator-created character expressions can be expressive and clear but require expertise and hours of work. In order to speed up the animation process, animators often use human actors to control and animate a 3D stylized character using a facial performance capture system. These systems often lack the expressive quality and perceptual validity of animator-created animations, mainly due to their assumption that geometric markers are sufficient for expression transfer. The geometry-based methods and retargeting [204] based on handcrafted descriptors may be unable to take into account the perception of the intended expression when transferred onto a stylized char-

---

<sup>1</sup><https://www.masseffect.com>

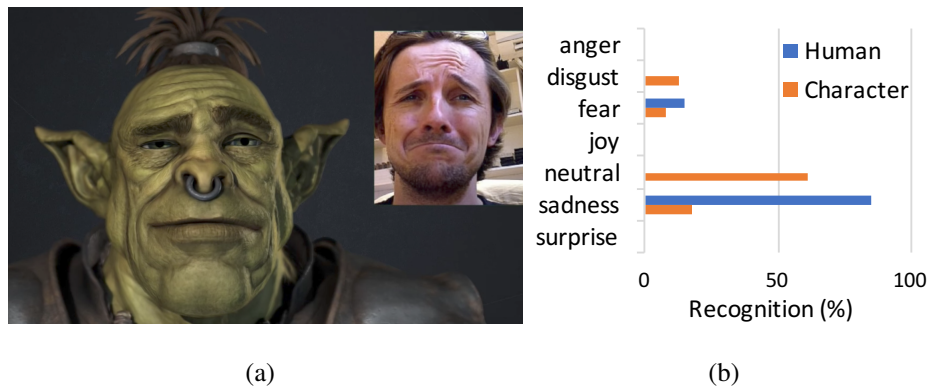


Figure 3.1: Example of inaccurate expression transfer. (a) Expression transfer from human (top right) to a character using Faceshift Studio. (b) Mechanical Turk testers perceive the human expression as sadness, while the character expression is perceived as neutral and a mixture of other expressions. The character expression has neither expression clarity nor geometric consistency.

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

As demonstrated in 2, the geometry-based methods and retargeting based on handcrafted descriptors may be unable to take into account the perception of the intended expression when transferred onto a stylized character. Despite recent advances in modeling capabilities, motion capture, and control parameterization, current methods do not address the fundamental problem of creating clear expressions that humans recognize as the intended expression. Our goal is to learn 3D stylized character expressions from humans in a *perceptually valid* and *geometrically consistent* manner. To this end, we propose an end-to-end system, ExprGen, that takes a 2D image of a human, and predicts the 3D rig parameters of a character, as an extension to our 2D expression retargeting work. It is relatively straightforward to train a regression network from a 2D rendering of a character to its rig parameters, because the training data can be created automatically. It is more challenging to train a network for human faces, because there is no existing dataset mapping

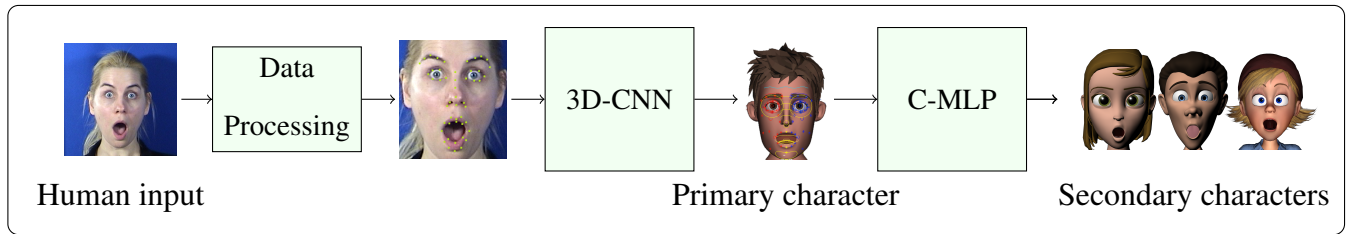


Figure 3.2: Overview of our multi-stage deep learning system ExprGen which takes 2D images of human facial expressions as input and generates 3D parameters of stylized character rigs at runtime. See Section 3.2 for details of the dataset and preprocessing and Section 3.3 for details of training the 3D-CNN and Character Multi-Layer Perceptron (C-MLP).

2D images of human expressions to 3D character rig parameters. Further, it is prohibitively expensive to create one manually. To address this challenge, we leverage publicly available human and character expression datasets with 2D images and expression labels [67, 194, 234, 196, 207, 6]. Each 2D character image in the dataset is associated with its 3D rig parameters.

Our approach is to learn the correspondence between 2D images of human and character expressions and use this correspondence to map human expression images to 3D character rig parameters. We start with recognizing facial expressions for both humans and characters to induce a perceptual metric and constraints for expression generation and matching. Our system then learns a joint embedding to map human expressions to character expressions. This mechanism also accounts for geometric consistency, so that any mapping is both perceptually valid and geometrically consistent. At this point, we can take any human expression image and find similar or dissimilar images in a database of character images. Using this embedding, we train a regression network, 3D-CNN, from a human expression onto the parameters of a specific or *primary* character 3D rig. We avoid explicit human image to 3D rig parameter labeling by using the rig parameters associated with the 2D character images. The 3D-CNN trains on a database of human expression images and the rig parameters of the character expression that most closely matches the expression, using the perceptual and geometric metrics.



Figure 3.3: Two new characters added to FERG-DB - ‘Mathilda’(left) and ‘Tuna’ (right)

Finally, a lightweight mechanism we call a *Character Multi-Layer Perceptron (C-MLP)*, transfers character expressions to other characters. This enables re-use of a primary character rig trained in the previous steps to drive *secondary* characters. Figure 3.2 depicts an overview of our system at run time. Images of human facial expressions are fed into the 3D-CNN to generate expression specific parameters for the primary character rig. The C-MLP uses the generated expression parameters to produce the character expression on other secondary 3D stylized characters.

In order to create such a system, a large amount of expression training data is needed. Fortunately, thousands of images of human and 3D stylized character faces labeled with expressions are readily available for studying facial expressions. In our work, the system learns from the seven broad categories (anger, disgust, fear, joy, sadness, surprise) [82] plus neutral, since there is agreement on their recognition within the facial expression research community, and these seven expressions occur in a wide range of intensities and can blend with each other to create additional expressions. We propose to use this data to create character expressions that exhibit perceptually valid expressions.

Our main contributions are:

- A novel perceptually valid method to map 2D human face images to 3D stylized character rig controls.
- The ability to utilize this mapping to generate 3D characters with clear unambiguous facial expressions.
- A semi-supervised method to enable expression transfer between multiple characters.

We also extended FER-DB by adding two new characters as shown in Figure 3.3 with almost 10,000 images for each new character. All the images are labeled for each of six cardinal expressions: Methods and results presented in this chapter were previously published in [5]

### 3.1.2 Related Literature

#### *Facial Expression Recognition (FER)*

With recent advancement in deep learning research, Convolutional Neural Networks have shown great improvement in the FER tasks [215, 34, 191, 151, 70, 153] and there are a number of fusion algorithms to boost the recognition performance [329, 336, 290, 333]. Many systems are tuned on a single facial expression database, which makes them sensitive to the lighting and particular poses present in that database. These methods focus on engineered features, which lack the generalizability to perform “in the wild” expression recognition. To overcome this limitation, we combine human databases from different sources including a dataset collected in the wild for our training step in order to improve the generalizability and robustness of our trained model. In our work, we use CNNs to learn from expression data by extracting perceptual and geometric features pertaining to the expressions of the input faces. This approach has been shown to perform better in expression recognition and transfer to cartoon characters than geometric features as described in Chapter 2.

#### *Expression Transfer*

Facial animation of stylized characters by retargeting human expressions can be classified into two main categories: 1) parametrization or generating facial animation parameters (based on geometry such as nose width, eye opening etc. [32] or physical control parameters such as muscle action [89]), and 2) motion retargeting [53], which involves mapping motion vectors directly to the target face model [225]. However, dense mesh motion required for motion retargeting may not be available from some input systems. Various other techniques like regression [265, 130], PCA-based linear modeling [57, 205] and blendshape mapping [273, 48, 29] have also been used to learn the mappings between an human and character faces. PCA-based models are insufficient to represent

the particularly detailed variations of a human facial expressions due to their limited dimensionality. In contrast to PCA models, blendshapes have the important advantage that a given facial expression among different people corresponds to a similar set of basis weights. Instead of using a fixed parametric shape model or relying upon the expression specific blendshape weights, our deep learning-based approach learns the perceptual and geometric features together from a database collected from various sources that can flexibly represent the particular facial expressions of the user without the need of depth information.

### *Generalization to Multiple 3D Characters*

Visual storytelling involving multiple stylized characters necessitates facial expression manipulation of multiple characters with minimum human effort. Several marker-based and markerless facial motion capture software packages have been recently developed, including Faceshift Studio [94], Faceware Analyzer and Retargeter [95], Mixamo Face Plus [213], Dynamixyz Performer Suite [73], and Optitrack Expression [229]. The traditional marker-based products create some predefined marker points on the human face and map them to the corresponding points on the 3D character rigs, enabling live tracking of the human facial motion and character facial animation. However, the limited number of marker positions often fail to capture the intended expression. The markerless systems like Faceshift and Faceware capture the blendshapes associated with a set of standard expressions made by the human source and map the blendshapes onto stylized characters. However, all these methods require a significant amount of manual effort in terms of setting up a new character, mapping the expressions from the existing character to a new one, and refining the generated expressions for accurate tracking.

### **3.2 Data Acquisition and Pre-processing**

Our framework uses two databases:

### *Human Expression Database (HED):*

We combine five publicly available labeled facial expression databases to create the HED: (a) Static Facial Expressions in the Wild (SFEW) database [67], (b) Extended Cohn-Kanade database (CK+) [194], (c) MMI database [234], (d) Karolinska Directed Emotional Faces (KDEF) [196], and (e) Denver Intensity of Spontaneous Facial Actions (DISFA) database [207]. The HED consists of approximately 100K labeled images; the number of samples for each class is balanced to avoid bias towards a particular expression.

### *Character Expression Database (CED):*

We use FERF-DB which consists of 55,767 labeled face images of six stylized characters as described in Section 2.2.1. We also use three new characters ('Tuna' [328], 'Mathilda' [282] and 'Cody' [86]).

The data pre-processing for HED is performed as described in Chapter 2. The 3D parameters of each rig are extracted from the CED and used to train a deep neural network for producing the expression specific 3D parameters.

## **3.3 Methodology**

### *3.3.1 Overview*

In order to build a system that can translate human expressions to multiple 3D character expressions, we need several components to handle the human-to-character transfer in 2D, produce parameters for a primary character expression in 3D including both perceptual and geometric similarity, and transfer the expression of a primary character to multiple secondary characters. We build a multi-stage deep learning system ExprGen with two major components: Training from 2D Datasets (Section 3.3.2) and 3D Expression transfer. 3D Expression transfer is composed of two separate components: Human to Character transfer (Section 3.3.3) and Character to Character transfer (Section 3.3.3).

Layer type	Filter size	Stride	Output
CONV-1	11x11	1	64x256x256
CONV-2	1x1	2	64x128x128
CONV-3	5x5	1	64x128x128
CONV-4	1x1	2	64x64x64
CONV-5	5x5	1	64x64x64
CONV-6	1x1	2	64x32x32
CONV-7	3x3	1	64x32x32
CONV-8	1x1	2	64x16x16
CONV-9	3x3	1	64x16x16
CONV-10	1x1	2	64x8x8
CONV-11	3x3	1	64x8x8
Avg. Pooling-12	8x8	1	64x1x1
FC-13			1x7
FC-14			1x7

Table 3.1: Network Configuration

### 3.3.2 Training from 2D Datasets

The goal of this step is to learn a joint embedding between human and primary character expressions based on perception and geometry. Our approach is inspired by the recent success of CNNs to learn the image similarity based on Pseudo-Siamese networks [52, 335]. We extend this concept for expression similarity application by fusing the perceptual and geometric features of humans and characters. We train a Pseudo-Siamese network called the f-CNN with two branches, Human CNN (HCNN) and Shared CNN (SCNN), that do not share weights between them. We first train the HCNN on the human expression dataset (HED) to classify an input human face image into one of the seven expression classes. Then, we train the SCNN on the character expression dataset (CED) by transfer learning [330]. In this process, the last layer of the HCNN is fine-tuned with the CED (for every character) by continuing the backpropagation learning step, creating a shared embedding feature space. The network structures for the HCNN and SCNN are given in Table 3.1.

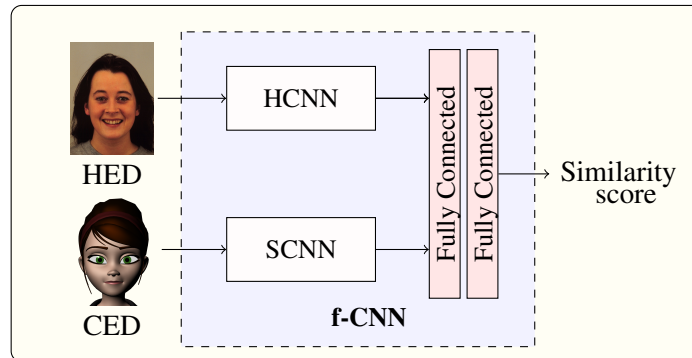


Figure 3.4: The HCNN and SCNN are fused together to form the f-CNN, which is trained to produce a similarity score between human and primary character expressions.

After the network branches are trained to recognize the expressions on humans and characters independently, we combine the fully connected layers of the networks to form a fused-CNN (f-CNN) as shown in Figure 3.4. To achieve this, the outputs from the average pooling layers are concatenated and given to a network of two Fully Connected (FC) layers. To train the f-CNN, we introduce a similarity measure based on the distance between two image encodings as follows. The HCNN and SCNN are used to predict the perceptual expression label of the human input image, and the CED is searched to retrieve the character images having the same predicted label. Then, the Euclidean distance between the geometric feature vector of the human input image and those of all the retrieved character images are computed and ordered based on the distance to the input image. To solve the issue of incorrect geometry match within the same expression class (such as open mouth character joy being matched with closed mouth human joy), triplets  $(E, E+, E-)$  of training images are created where:

- $E$  is a reference human expression image
- $E+$  is a character image similar to the reference human expression image (best geometry match in the search)
- $E-$  is another image that is not geometrically and/or perceptually similar to the reference

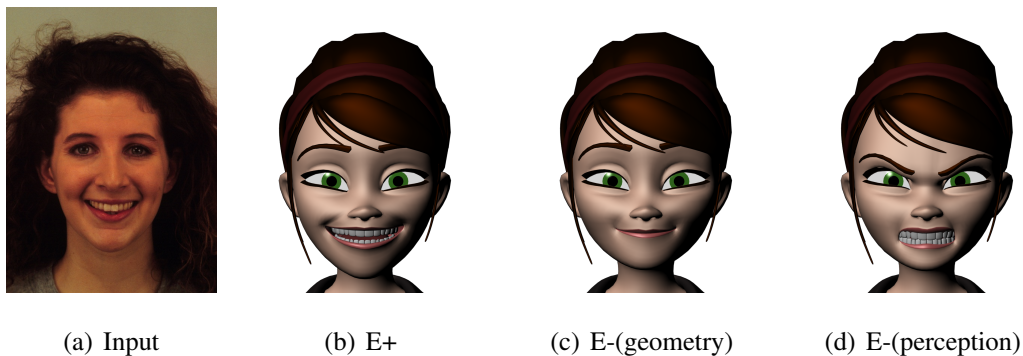


Figure 3.5: Comparison of best matches for training the f-CNN based on geometry and perception. (a) Human input (E), (b) Positive match (E+), (c) Negative match with incorrect geometry (E-), (d) Negative match with incorrect perception (E-).

human expression image. For example, if E is an open mouth joy human expression, then character anger retrieval would be incorrect perceptually and closed mouth human joy would be incorrect geometrically as shown in Figure 3.5.

The f-CNN takes a human expression image and primary character expression image and produces a similarity score by minimizing a loss function. It is trained in a supervised manner to make E+ closer to E than E- using a hinge-based loss term and squared L2-norm regularization term:

$$\min_w \frac{\lambda}{2} \|w\|_2 + \sum_{i=1}^N \max(0, 1 - l_i y_i^{net}) \quad (3.1)$$

where  $w$  are the weights of the neural network,  $y_i^{net}$  is the network output for the  $i^{th}$  training sample, and  $l_i \in \{-1, 1\}$  is the corresponding label (with +1 and -1 denoting a non-matching and a matching pair, respectively). The hinge loss minimizes the distance between the human input expression and a positive character example (matching both geometry and perception) and maximizes the distance between the human input expression and a negative character example (mismatching the geometry and/or perception). Similar to the approach described in [267], triplets are generated online by selecting the hard positive/negative exemplars from within a mini-batch for our training. The f-CNN is trained for the final 2-class classification (similar or not-similar), and the output of this step is the similarity score based on the probability of similarity prediction.

### 3.3.3 3D Expression Transfer

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

#### *Human to Character Transfer*

The f-CNN can be used to retrieve the matching 2D character expressions; however, it requires a database of character images. We aim to control the primary rig by producing rig parameters for any given human image. To control the rig in 3D, we train another CNN called the 3D-CNN which has the same configuration as shown in Table 3.1 except for the dimensionality of the FC layers. Instead of seven probabilities for classifying seven expression classes, the final output is the parameters for the primary character. We initialize the weights of the 3D-CNN by trained HCNN weights so that we can transfer the knowledge learned from the HED, and the model does not overfit the 3D-parameters dataset. The pairs of a human input image and corresponding 3D-parameters for producing a character with similar expression are used for training the 3D-CNN.

All the networks are trained end-to-end using the Torch framework [56] until convergence using stochastic gradient descent with hyper parameters (momentum of 0.9, weight decay of 0.0005 and a batch size of 50) on a single NVIDIA GTX-1080 GPU. In order to make sure that the pre-trained weights are not drastically changed, the learning rates for the SCNN, f-CNN and 3D-CNN are set lower (0.0001) than that of the HCNN (0.001). The learning rate was dropped by a factor of 10 after every 10 epochs of training. Batch normalization was applied [136] after every convolutional layer to reduce the internal-covariate-shift, ReLU as the activation function and drop out with the drop-out ratio of 0.2. To avoid overfitting, our training data is augmented by horizontal flipping, rotating, and random cropping followed by scaling. An 80:10:10 split was used for training, validation and test sets, and 5-fold cross validation was performed.

### *Character to Character Transfer*

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

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

parameters are used to train the C-MLP. Once trained, the C-MLP can input the 3D parameters for a primary character and output corresponding 3D parameters for a selected secondary character.

### **3.4 Results**

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

#### *3.4.1 Expression Recognition Accuracy*

We first evaluated the HCNN and SCNN for the expression recognition task using the HED and CED in a 10-fold cross-validation setting. The HCNN and SCNN obtained average accuracies of 89.71% and 96.82%, respectively. We note that our classification networks perform better than the prior networks trained for a similar classification task (see Section 2.4.1) because of training the HCNN on an additional dataset to learn the features in the wild. The accuracy of our networks increased by about 5% when we did not apply the max pooling step after every convolution layer, indicating that average pooling after all the convolution layers helps the network to preserve the facial appearance and subtle distinctions between each expression, which is lost when max pooling is applied after every convolution. However, our focus is not on the classification accuracy of the trained networks, but on using them to produce 3D-rig parameters. In the remaining experiments we use these networks to learn the expression feature space for humans and stylized characters and use their weights to initialize our final 3D-CNN.

Class	Anger	Disgust	Fear	Joy	Neutral	Sadness	Surprise
Human	76.27	63.81	68.47	94.31	78.03	72.95	92.26
Character	90.45	72.89	79.16	96.39	84.38	79.44	94.87

Table 3.2: Average (%) expression recognition accuracy for 2D images of human and stylized character expressions compared with the ground truth labels respectively. Note that the characters have higher expression clarity than humans due to their simpler geometry.

### 3.4.2 Human to Character Expression Transfer

To evaluate our results for clarity in expression recognition and perceptual accuracy of the transferred expression, we asked 30 Mechanical Turk (MT) test subjects to recognize the input human expression and the generated primary character expression (output of 3D-CNN) for 1000 expression transfer results (approx. 150 for each expression class) on different stylized characters.

We computed the clarity of expression recognition on human and characters independently by comparing the perceived expressions with the ground truth labels. The average (%) expression recognition accuracies for humans and characters are shown in Table 3.2. We observe that the character expression recognition accuracies are higher than humans, since the characters have simpler geometry and stylization can make the expressions relatively easier to perceive. Surprise and joy show high accuracy, while disgust and fear are more difficult for humans to both perceive and act out.

To test the accuracy of the expression transfer we compared human expressions to that of the generated primary character. We assumed the perceived label of the human input as the actual input and the perceived label of the character as the predicted output in its human-character-transfer pair. Our focus is on the perception of the transferred expression on the character rather than matching the ground truth label for humans. Figure 3.7 shows the confusion matrix for transferred expression recognition for each expression class. For a given row (e.g. anger), the columns represent the percentage (averaged over all the perceived human anger expressions) of MT subjects agreeing on

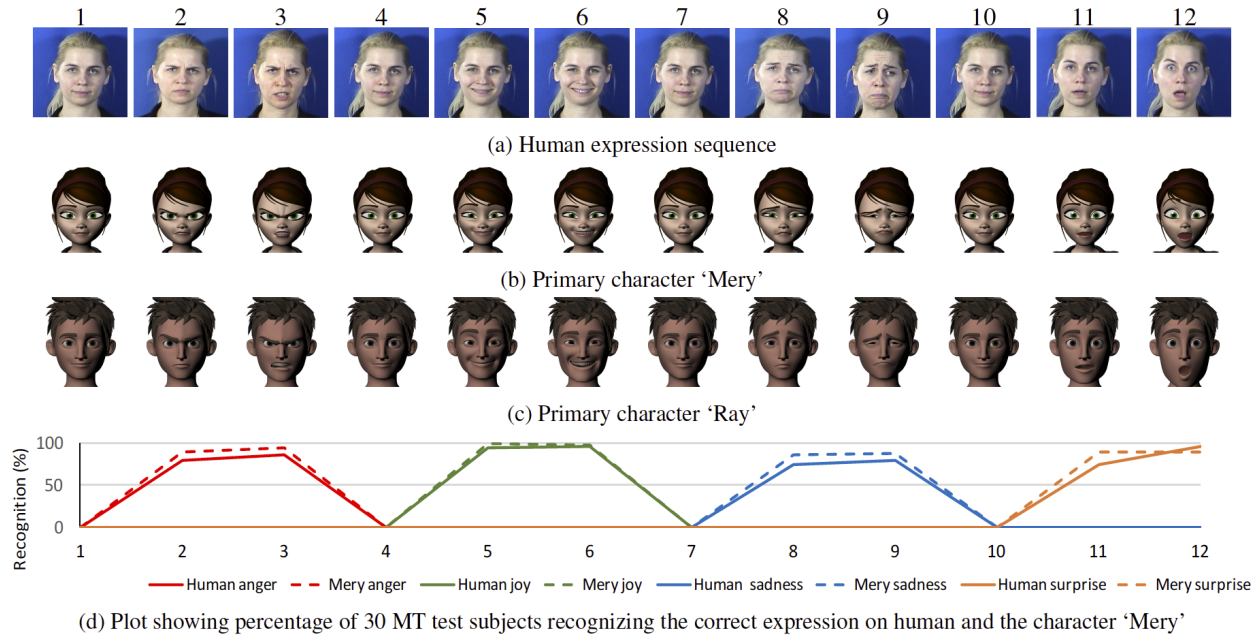


Figure 3.6: Human to primary character expression transfer for human expression transition from neutral to joy, from neutral to surprise, from neutral to sadness, and from neutral to anger based on both perceptual and geometric similarity. (a) Human input expression frames (1-12), (b) Mapped expressions on 'Mery', and (c) Mapped expressions on 'Ray', (d) Expression recognition results between human (solid lines) and transferred expressions on 'Mery' (dashed lines) for different expressions.

the corresponding expression classes for the transferred character expressions. The values show that ExprGen results in accurate transfer of expressions for most of the classes with an average correct perceptual recognition rate of 75.55%. The most common errors are confusion between disgust and anger, between fear and surprise, and between neutral and sadness. These errors are intuitively reasonable since the confused expressions have similar-looking geometric configurations. The least accurate expression transfer was for disgust and fear but as Table 2 shows, these expressions are difficult to recognize for both human and character images.

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

Figure 3.7: Confusion matrix for perceived transferred expression recognition (%) for seven expression classes.

### *Single Human to Multiple Characters*

ExprGen generates expressions for multiple characters with high perceptual validity. The expression transfer results from a human to two stylized characters are shown in Figure 3.6. We tested the expression recognition on input human expressions and transferred expressions on ‘Mery’ (3.6 (b)) using 30 MT test subjects. The plot shown in 3.6 (d) shows high correlation between MT agreement for recognized expressions, which confirms the accurate perception of the intended expression transfer. We also show the generalizability of our algorithm in generating the same expressions on different characters by learning their 3D-rig parameters as shown for ‘Ray’ in Figure 3.6 (c). Our model can be trained for any 3D rig for which we have annotated training data.

### *Multiple Humans to a Single Character*

To illustrate the consistency of our system, we generated the same expression on a single character from different human inputs as illustrated in Figure 3.8, showing that our algorithm is consistent for transferring the expression even when there is variation in the human input examples.

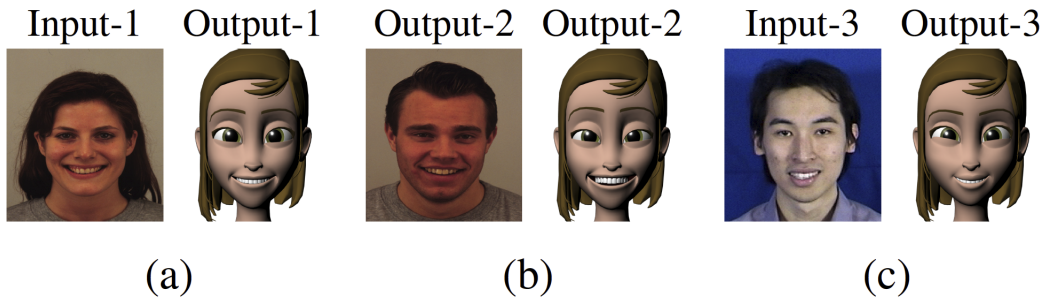


Figure 3.8: Consistent human expression transfer to primary characters. This example shows three different humans with the same expression (joy) transferred to the primary character ‘Aia’.

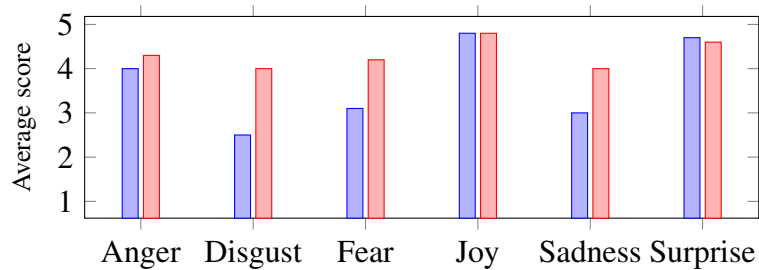


Figure 3.9: Quantitative comparison of expression transfer results of Faceware (blue bars) and ExprGen (red bars).

### Comparison with Faceware

ExprGen generates expressions with greater perceptual validity than popular commercially available software packages. We compared ExprGen with the award-winning<sup>2</sup> Faceware technology [95], because it takes the same input modality as ExprGen. Faceware includes *Analyzer* to convert human facial performance from a sequence of input images into motion capture data and *Retargeter* to map the captured data to the blendshapes of the 3D character face rig by manually creating an expression set for the character. Figure 3.9 shows the comparison of average scores obtained for different expression classes when 30 MT test subjects were asked to rate the closeness of the expression generated on the character to the input human expression on a scale of 1-5, with 5 being

<sup>2</sup><http://facewaretech.com/blog/>

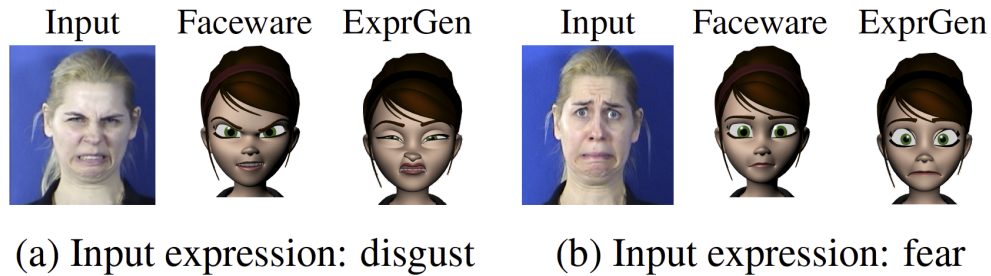


Figure 3.10: Qualitative comparison of expression transfer results of Faceware and ExprGen (left to right: input human expression, Faceware output and ExprGen output).

the closest match. The average score over all classes for ExprGen is 4.31 versus an average score of 3.68 for Faceware. Figure 3.10 shows the expression transfer results of Faceware and ExprGen for two cardinal expressions. These results show that blendshape-mapping-based approaches often produce incorrect expressions (see Figure 3.10(a)) or ambiguous expressions (see Figure 3.10(b)) owing to the limitations of correspondence mapping. We did not compare with the results of Faceshift Studio [94], since it requires a depth camera to capture human facial motion and uses a different approach compared to our 2D human image to 3D stylized character rig mapping.

### 3.4.3 Character to Character Expression Transfer

In order to evaluate the performance of our character-to-character expression model, we selected ‘Mery’ as the primary character, ‘Bonnie’ as the existing secondary character (present in FERGDDB) and ‘Tuna’ and ‘Cody’ (non-human) as the new secondary characters (not present in FERGDDB). Figure 3.11 (a) shows six randomly chosen cardinal expressions on the primary character used as test cases, and Figure 3.11 (b), (c) and (d) show the facial expressions generated on the secondary characters at the output of the C-MLP. The results show that our network accurately learns the relationship between the 3D parameters of the characters, while maintaining the clarity of the expressions. Our network produces surprisingly good results for non-human characters as well, though the C-MLP is trained on only the key poses. However, the training examples for new secondary characters are critical to this approach, and there are two issues in selecting accurate

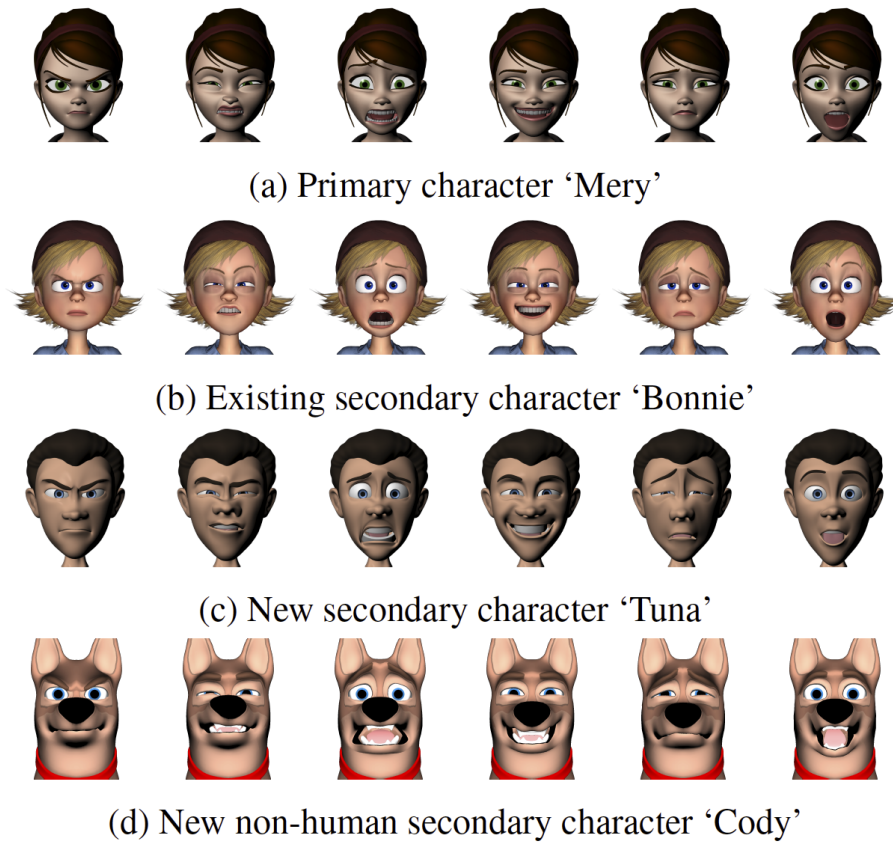


Figure 3.11: Primary to Secondary character expression transfer results (left to right: anger, disgust, fear, joy, sadness and surprise). (a) 'Mery's' expression classes, Expressions transferred to (b) 'Bonnie', (c) 'Tuna', and (d) 'Cody'.

training examples. First, when the new secondary character expression is perceptually valid but a similar expression does not exist for the primary character in the database (see Figure 3.12 (a)), our method retrieves the closest possible match which may be inaccurate. Second, when the new secondary character expression is perceptually ambiguous (see Figure 3.12 (b)), our method tries to find the closest match based on geometric features within the wrong expression classes, which may result in a wrong training example. Our future work will extend the training of new secondary characters by automating the process of generating large numbers of poses for each new character.

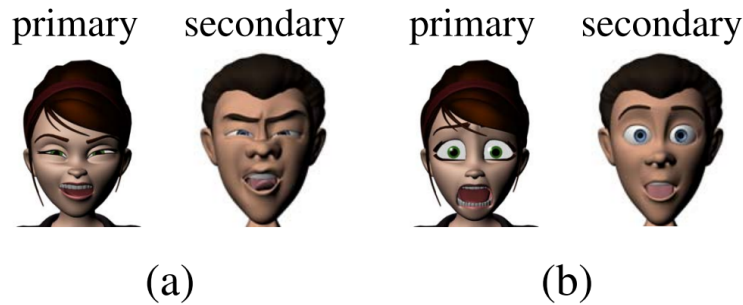


Figure 3.12: Error cases in obtaining training examples for new secondary characters. (a) Matching is perceptually valid (both expressions are disgust) but geometrically incorrect, (b) Matching is perceptually invalid (expression on left is fear and on right is surprise) but geometrically correct.

### 3.5 Summary

We have demonstrated a novel multi-stage deep learning method to transfer human facial expressions to multiple 3D stylized characters that optimizes over expression clarity rather than over geometric markers. The resulting expressions, when validated by Mechanical Turk studies, show that our expression transfer clearly reproduces the input human expressions and generalizes to multiple human source expressions and multiple character targets. ExprGen has several practical applications including visual storytelling, games, social VR experience and human-robot interactions. Our work provides the foundation for several future explorations, including learning expression intensity, adding animation etc. It will be interesting to add the concept of a universal primary character rig that is sufficiently powerful to create a full range of expressions and can be quickly extended to unusual character designs such as one-eyed or fantasy characters.

## Chapter 4

### EXPRESSION RETARGETING ON STYLIZED ANIMAL CHARACTERS

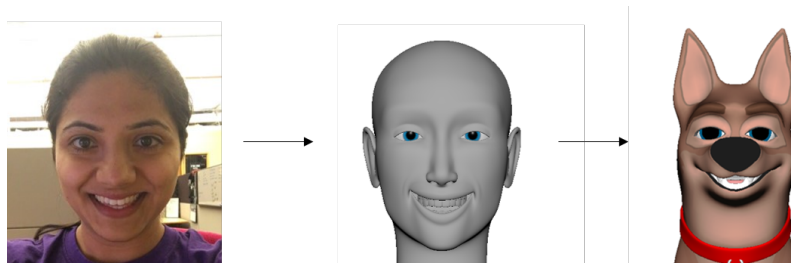


Figure 4.1: Overview of our expression retargeting which takes 2D images of human facial expressions (left), generates expressions on the universal rig [6] (middle) and retargets the expression to drive a 3D stylized animal character.

#### 4.1 Introduction

##### 4.1.1 Motivation

In Chapter 3, we demonstrated a deep learning-based approach that uses images of human facial expressions to generate perceptually valid and geometrically consistent expressions on characters that have similar facial proportions to human faces. This approach also uses a large number of training examples for retargeting the expressions on the primary character.

To address the challenges noted above, our goal is to place exciting stylized characters that go beyond the normal human look and function as expressive surrogates for the user. The users will be able to manipulate these stylized characters as avatars, controlling the avatar’s expressions with their own. For instance, a user might want to control a chicken or an octopus instead of controlling an avatar just looking like that user. Emotional expressions on such characters have not

been heavily studied; most of the work on expression transfer, including our own, have been either human to humanoid or human to stylized characters that have all the features of a human. We want to explore the situation where the avatar has a very different geometry (beak or snout instead of a mouth), or some features are missing.

In our work, we focus on retargeting expressions on stylized animal characters with different facial geometry as compared to humans. Figure 4.1 shows the overview of our system. We introduce a humanoid rig - Androgynous Human (“Andy Hugh”) shown in Figure 4.1 (middle), which is capable of expressing diverse expression variations with different levels of intensities. The expressions on the universal character can be generated using ExprGen as described in chapter 3. In this chapter, we develop a learning-based approach to drive the stylized animal characters from the universal rig. We utilize the Gaussian process (GP) [254] based nonlinear latent variable models [306, 60] to learn the underlying low-dimensional representation from high dimensional Hugh expressions. Finally, we learn a shared latent space, because Hugh expression poses and corresponding animal character expression poses have some underlying nonlinear relationship.

#### 4.1.2 *Related Literature*

While several facial retargeting techniques are developed, most of them do not apply to non-humanoid characters because they assume that the target character has human-like geometry. There has been facial retargeting work on digital avatars that look different from human geometry that use depth information [185], require offline pre-processing with manual correction by the user [38], or use motion capture data as the source for human expression data. [51].

Our learning approach is inspired by extensive applications of Gaussian Process Latent Variable Models (GPLVM) in graphics, vision and robotics for applications including facial expression recognition [85], expression retargeting [28], speech-driven facial animation [63], human body tracking [129], robotic imitation [279], cloth-state estimation [167], and deriving inference in human-robot interaction [318]. Our system is closely related to non-linear example-based methods for facial expression retargeting [28], human motion retargeting [326], and animating interactive characters [14]. The main difference to existing solutions is that our approach introduces a univer-

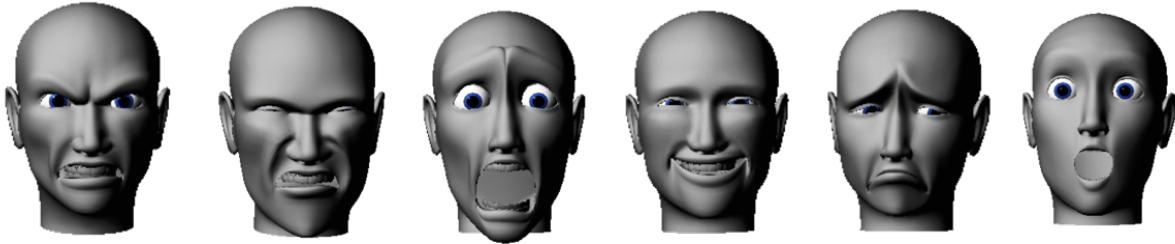


Figure 4.2: Hugh’s cardinal expression from left to right: anger, disgust, fear, joy, sadness and surprise.

sal humanoid character to represent the expression spectrum of the human expression space generating new compound expressions and exploits the Manifold Relevance Determination (MRD) technique [60] to learn the expression latent features in a Bayesian manner, thus avoiding the problem of overfitting with a sparse set of training examples.

## 4.2 Data Collection

We created a humanoid universal character rig “Andy Hugh” (referred to as Hugh) within the Maya [233] environment. The rig was created by team of facial expression experts and animators. It has various face controls for creating facial expression poses. Pictures of Hugh showing different expressions are shown in Figure 4.2. The character’s appearance can be manipulated in several ways to create facial expressions. We used ExprGen as described in Chapter 3 and treated Hugh as the primary character to generate 5500 expression poses from human 2D images. We used 4000 samples for training and 1500 for testing.

After building the Hugh dataset, we worked with the team of animators to choose 30-35 key poses with 5-6 from each expression class (anger, disgust, fear, joy, sadness and surprise). The animator creates the character key poses corresponding to the chosen keyposes for each expression, and they were tested via MT for a perceptually valid match with Hugh. We only used the expression key poses having 70% MT test agreement among 50 test subjects for the same pose. In our work,

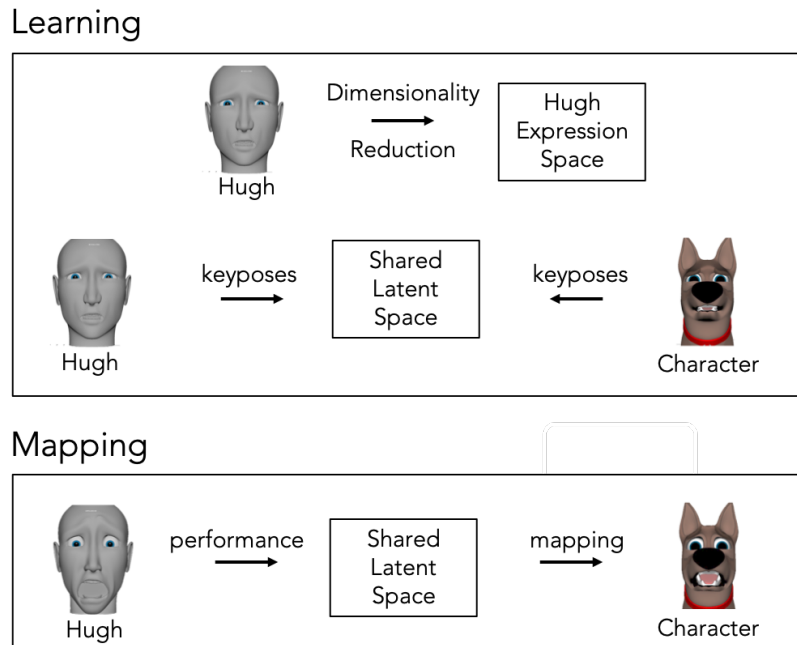


Figure 4.3: Expression retargeting pipeline. First, we learn a low-dimensionality representation of Hugh expression space, and then learn the shared latent space between Hugh and the character using the keyposes. Finally, we apply the Hugh-to-character mapping to generate expression transfer results.

we used two stylized animal character rigs - a dog “Cody” [86] and an otter “Noodle” [257].

### 4.3 Methodology

The overview of our pipeline is shown in Figure 4.3. We perform non-linear dimensionality reduction of Hugh expression observation space by training a BGPLVM. This space can also be used to synthesize novel expressions of Hugh that do not exist in the training data. Then we train an MRD model to learn the shared latent space between Hugh and the character expressions. Finally, we can generate the character expressions corresponding to Hugh expressions using the shared latent mapping.

### 4.3.1 Training BGPLVM

We train a BGPLVM to learn the low-dimensionality representation of the High expression space. We model the relationship between two variables  $\mathbf{X}$  and  $\mathbf{Y}$  as a function  $f$ , i.e.  $f : \mathbf{X} \rightarrow \mathbf{Y}$ . Let us denote  $\mathbf{Y}$  as a matrix of observations with dimensions  $N \times D$ , where  $N$  rows correspond to data-points and  $D$  columns to dimensions as  $\mathbf{Y} = \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N$ ,  $\mathbf{y}_n \in \mathbb{R}^D$ . In the LVM methodology, we assume that these observations come from a latent space  $\mathbf{X}$  in  $N \times M$ ,  $M \ll N$  denoted as  $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ ,  $\mathbf{x}_n \in \mathbb{R}^M$ . The goal of GPLVM is to learn the low-dimensional representation  $\mathbf{X}^{N \times M}$  of the data matrix  $\mathbf{Y}^{N \times D}$ . GPLVM assumes that  $\mathbf{Y}$  is generated from  $\mathbf{X}$  using a non-linear mapping with a GP prior. Although this mapping can be integrated out analytically, the latent variables cannot. Therefore, GPLVM is traditionally optimised using a maximum a posteriori probability (MAP) estimate. The Bayesian GPLVM [306] (BGPLVM) allows for approximately marginalizing out  $\mathbf{X}$ , where we take a Bayesian estimation of GPLVM whose latent variables are integrated out instead of optimized. This process has some advantages: Firstly, it is more robust to overfitting, since we now have a proper distribution for the latent variables (rather than point estimates). Also, we can perform automatic dimensionality detection by using Automatic Relevance Determination (ARD) [254] covariance functions for the GP prior.

The marginal likelihood of the data using variational Bayesian approach is given as:

$$p(\mathbf{Y}|\theta) = \int p(\mathbf{Y}|\mathbf{X}, \theta)p(\mathbf{X})d\mathbf{X} \quad (4.1)$$

However, the integral for marginalization becomes intractable as  $\mathbf{X}$  appears non-linearly in the inverse of the kernel covariance matrix  $\mathbf{K}$ . The variational method allows to compute a Jensen's lower bound on the GPLVM marginal likelihood and the key to obtaining this bound was to introduce auxiliary variables into the model similar to those used in sparse GP regression. This augmented probability model leads to a tractable Jensen's lower bound through the removal of the non-linear factor  $p(Y|X)$  thereby making the approximation tractable. More details are given in

[306]. The prediction of unseen test data  $y^*$  is evaluated as:

$$p(\mathbf{y}^*|\mathbf{Y}) = \frac{\int p(\mathbf{y}^*, \mathbf{Y}|\mathbf{x}^*, \mathbf{X})p(\mathbf{x}^*, \mathbf{X})d\mathbf{X}d\mathbf{x}^*}{\int p(\mathbf{Y}|\mathbf{X})p\mathbf{X}d\mathbf{X}} \quad (4.2)$$

In our work, we use the Radial Basis Function (RBF) kernel and the length scale of the RBF kernel is initialized by the normalized eigen values obtained from the PCA initialization. To train the model, we first optimize by fixing variance parameters by initializing the variance of the kernel and variance for Gaussian likelihood. In the next step, we optimize the model without any constraints, and the model is optimized until there are negligible increments in the likelihood function.

#### 4.3.2 Training MRD

In order to learn a shared latent space between Hugh and the character expressions, we utilize MRD as an extension to BGPLVM. The model is trained over observations from Hugh expression feature space and character expression space using the keyposes. The trained model is then used to infer the character expressions given a test Hugh expression pose. We trained MRD for each character using the key poses created by the animator. For both the stylized animal characters, we used  $\approx 25$  key poses for training and 5 for validation with 5-fold cross-validation. We separated out 5-6 key poses for testing our model.

The goal is to evaluate the shared latent variables and the GP mapping hyper parameters for each observation space  $\theta^{Y,Z}$ . The joint conditional likelihood is obtained by factorizing each observation space, given as:

$$p(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \theta^Y, \theta^Z) = \prod_{\{\gamma=\mathbf{Y}, \mathbf{Z}\}} p(\gamma|\mathbf{X}, \theta^\gamma) \quad (4.3)$$

Similar to BGPLVM, the marginalization of the latent variables for the shared latent space is intractable due to its non-linear appearance in the kernel covariance matrix. An approximate variational inference formulation has been proposed that relies on the use of an augmented probability model similar to BGPLVM. A detailed derivation is given in [306, 60].

After we have a model which is trained to jointly represent two output spaces  $\mathbf{Y}$  and  $\mathbf{Z}$  with a common but factorised input space  $\mathbf{X}$ , we wish to generate a new (or infer a training) set of outputs  $\mathbf{Z}^* \in \mathbb{R}^{N^* \times D_z}$  given a set of (potentially partially) observed test points  $\mathbf{Y}^* \in \mathbb{R}^{N^* \times D_Y}$ .

The inference steps are given as follows:

1. Firstly, we predict the set of latent points  $\mathbf{X}^* \in \mathbb{R}^{N^* \times M}$  that is most likely to have generated  $\mathbf{Y}^*$ . For this, we use an approximation to the posterior  $p(\mathbf{X}^* | \mathbf{Y}^*, \mathbf{Y})$ , that has the same form as for the standard Bayesian GPLVM model [306]
2. Then, we find the training latent points  $\mathbf{X}_{NN}$  which are closest to  $\mathbf{X}^*$  in the shared latent space.
3. Finally, we find outputs  $\mathbf{Z}$  from the likelihood  $p(\mathbf{Z} | \mathbf{X}_{NN})$ . This procedure returns the set of training points  $\mathbf{Z}$  that best match the observed test points  $\mathbf{Y}^*$ .

## 4.4 Results

In this section, we investigate the performance of using non-linear modeling for Hugh expression space estimation and learning the shared latent space between Hugh expression feature space and the character expression feature space. First, we evaluated the performance by only considering the Hugh observation data and by comparing the performance of BGPLVM with a linear latent variable model, Principal Component Analysis (PCA). Then, we compared the performance of MRD with standard regression techniques for inferring character expressions.

### 4.4.1 Quantitative Results

The performance of using BGPLVM and MRD is evaluated using three metrics in all the experiments i.e. the root mean square error (RMSE), normalized root mean square error (NRMSE), and Pearson correlation coefficient (Corr).

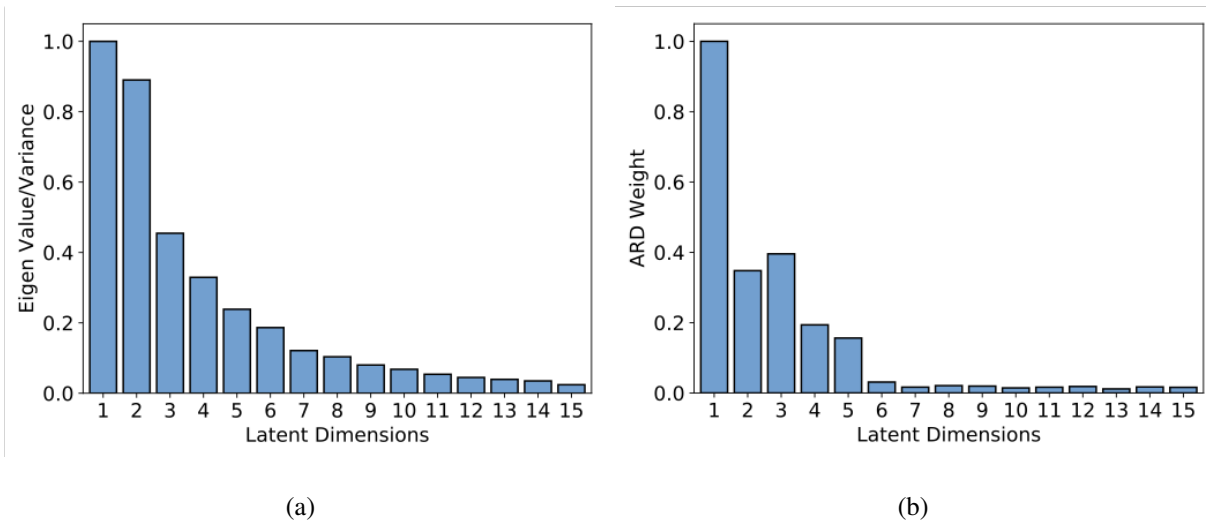


Figure 4.4: Comparison of Latent Dimension relevance learned by PCA and BGPLVM: (a) PCA relevance given by eigenvalues, (b) BGPLVM relevance given by ARD kernel weights.

#### 4.4.2 *Hugh Latent Features Learned*

We performed dimensionality reduction using both BGPLVM and PCA on the Hugh observation space and inspected the learned latent structures for both models. Figure 4.4 demonstrates the relevance of each dimension in the latent space after training. The relevance for PCA is given by eigenvalues and by the ARD kernel weight parameters for BGPLVM. The relevance parameters for both models are normalized such that the most significant dimension has a weight of 1.0 to demonstrate the relative importance between the dimensions. The relevance weights indicate that PCA takes all 15 dimensions to capture Hugh expression space through the linear mapping, whereas BGPLVM captures the underlying features within 5 dimensions using the non-linear GP mapping. To evaluate generalization capability, we compared reconstruction between PCA and BGPLVM as shown in Table 4.1. The results show that BGPLVM has much better performance with significantly ( $p$ -value  $< 0.05$ ) lower RMSE and higher correlation as compared to PCA.

Data	RMSE		Correlation	
	PCA	BGPLVM	PCA	BGPLVM
Train	0.051	0.0149	0.922	0.973
Test	0.064	0.0163	0.871	0.932

Table 4.1: Reconstruction error for PCA AND BGPLVM models for Hugh expression space.

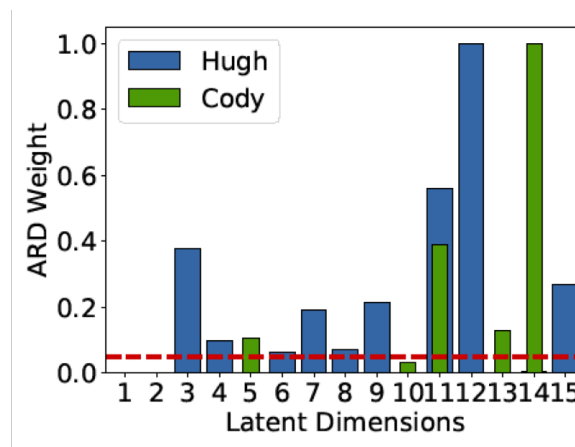


Figure 4.5: The ARD Kernel weights learned for each observation space when MRD model trained for Hugh and Cody expression spaces.

#### 4.4.3 Shared Latent Space Learned

As MRD model is trained for each character, and we show the results for training between Hugh and Cody in Figure 4.5. The result shows the sets of ARD kernel weights that are learned. The threshold on ARD weights is set to 0.05 as shown by the red line leading to shared and private dimensions between the observation spaces.

#### 4.4.4 Qualitative Results

The expression retargeting results for different expressions on Cody and Noodle are shown in Figure 4.6. We used the MRD model to infer the test expression poses for both the stylized animal

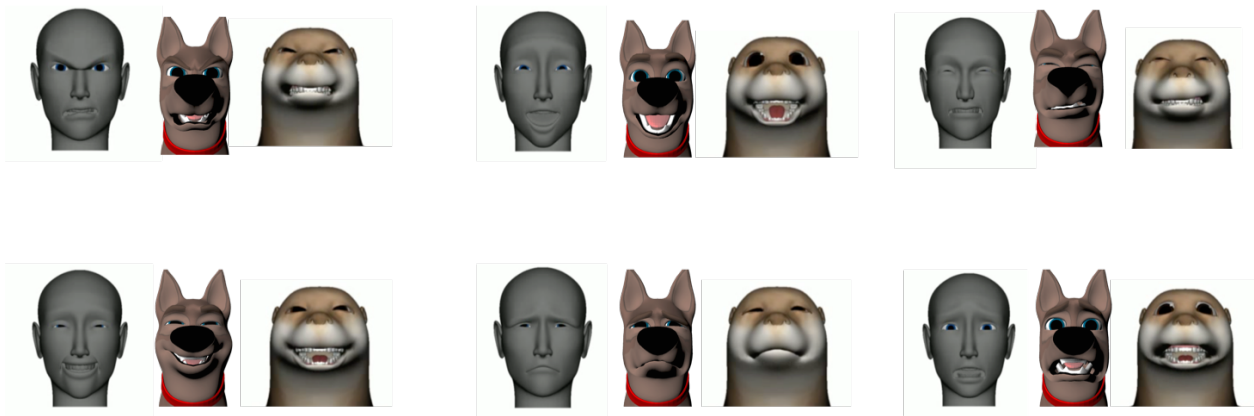


Figure 4.6: Expression retargeting results on Cody and Noodle.

rigs. The results show that the shared latent space is capable to generating the matching expression poses reliably. For future work, we intend to perform more thorough analysis and verify the retargeting results by comparing the predictive performance of our method with other state-of-the-art expression transfer techniques and running MT perceptual studies.

#### 4.5 Summary

Facial expressions inherently lie in a high dimensional configuration space. We hypothesize that human facial expressions can be represented by a much lower expression-specific latent space. We used Bayesian latent variable models to learn the underlying latent features from high dimensional observation spaces. We introduced a universal humanoid character and used a stylized animal expression data to learn a shared latent manifold that captures complementary latent features from both the expression spaces.

## Chapter 5

# LIP SYNC FOR LIVE 2D ANIMATION

### 5.1 Introduction

#### 5.1.1 Motivation

Traditional workflows for creating 2D animations are highly labor-intensive; animators either draw every frame by hand (as in classical animation) or manually specify keyframes and motion curves that define how characters and objects move. However, *live 2D animation* has recently emerged as a powerful new way to communicate and convey ideas with animated characters. In live animation, human performers control cartoon characters in real-time, allowing them to interact and improvise directly with other actors and the audience. Recent examples from major studios include Stephen Colbert interviewing cartoon guests on The Late Show [47], Homer answering phone-in questions from viewers during a segment of The Simpsons [102], Archer talking to a live audience at Comic-Con [11], and the stars of animated shows (e.g., Disney’s Star vs. The Forces of Evil, My Little Pony, cartoon Mr. Bean) hosting live chat sessions with their fans on YouTube and Facebook Live. In addition to these big budget, high-profile use cases, many independent podcasters and game streamers have started using live animated 2D avatars in their shows.

Lip sync for 2D animation is typically done by first creating a discrete set of mouth shapes (visemes) that map to individual units of speech for each character. To make a character talk, animators choose a timed sequence of visemes based on the corresponding speech. Note that this process differs from lip sync for 3D characters. While such characters often have predefined blend shapes for common mouth poses that correspond to visemes, the animation process involves smooth interpolation between blend shapes, which moves the mouth in a continuous fashion. The discrete nature of 2D lip sync gives rise to some unique challenges. First, 2D animators have a constrained palette with which to produce convincing mouth motions. While 3D animators can

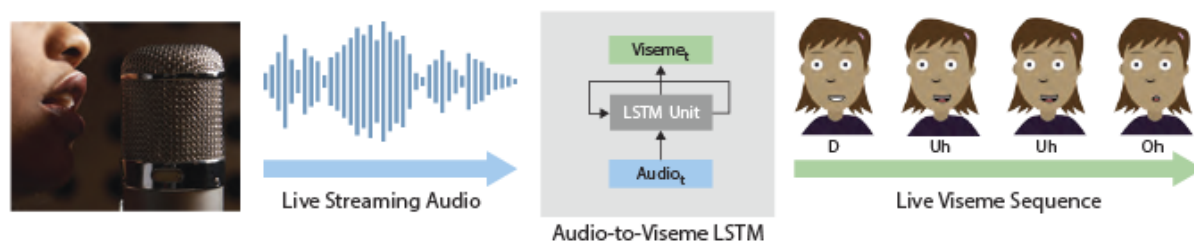


Figure 5.1: Real-Time Lip Sync. Our deep learning approach uses an LSTM to convert live streaming audio to discrete visemes for 2D characters.

slightly modify the mouth shape to produce subtle variations, 2D animators almost always restrict themselves to the predefined viseme set, since it requires significantly more work to author new viseme variations. Thus, choosing the appropriate viseme for each sound in the speech is a vital task. Furthermore, the lack of continuous mouth motion means that the timing of transitions from one viseme to the next is critical to the perception of the lip sync. In particular, missing or extraneous transitions can make the animation look out of sync with the speech. Given these challenges, it is not surprising that lip sync accounts for a significant fraction of the overall production time for many 2D animations. In discussions with professional animators, they estimated five to seven hours of work per minute of speech to hand-author viseme sequences.

Of course, manual lip sync is not a viable option for our target application of live animation. For live settings, we need a method that automatically generates viseme sequences based on input speech. Achieving this goal requires addressing a few unique challenges. First, since live interactive performances do not strictly follow a predefined script, the method does not have access to an accurate transcript of the speech. Moreover, live animation requires real-time performance with very low latency, which precludes the use of accurate speech-to-text algorithms (which typically have a latency of several seconds) in the processing pipeline. More generally, the low-latency requirement prevents the use of any appreciable “lookahead” to determine the right viseme for a given portion of the speech. Finally, since there is no possibility of manually refining the results after the fact, the automatic lip sync must be robust.

We propose a new approach for generating live 2D lip sync. To address the challenges noted above, we present a real-time processing pipeline that leverages a simple Long Short Term Memory (LSTM) [123] model to convert streaming audio input into a corresponding viseme sequence at 24fps with less than 200ms latency (see Figure 5.1). While our system largely relies on an existing architecture, one of our contributions is in identifying the appropriate feature representation and network configuration to achieve state-of-the-art results for live 2D lip sync. Another key contribution is our method for collecting training data for the model. As noted above, obtaining hand-authored lip sync data for training is expensive and time-consuming. Moreover, when creating lip sync, animators make stylistic decisions about the specific choice of visemes and the timing and number of transitions. As a result, training a single “general-purpose” model is unlikely to be sufficient for most applications. Instead, we present a technique for augmenting hand-authored training data through the use of audio time warping [20]. In particular, we ask animators to lip sync sentences from the TIMIT [107] dataset that have been recorded by multiple different speakers. After providing the lip sync for just one speaker, we warp the other TIMIT recordings of the same sentence to match the timing of the first speaker, which allows us to reuse the same lip sync result on multiple different input audio streams.

We ran human preference experiments to compare the quality of our method to several baselines, including both *offline* (i.e., non-live) and *online* automatic lip sync from two commercial 2D animation tools. Our results were consistently preferred over all of these baselines, including the offline methods that have access to the entire waveform. We also analyzed the tradeoff between lip sync quality and the amount of training data and found that our data augmentation method significantly improves the output of the model. The experiments indicate that we can produce reasonable results with as little as 13–15 minutes of hand-authored lip sync data. Finally, we report preliminary findings that suggest our model is able to learn different lip sync styles based on the training data. Video summary and supplementary results at GitHub link: <https://github.com/deepalijaneja/CharacterLipSync2D>

### 5.1.2 Related Literature

There is a large body of previous research that analyzes speech input to generate structured output, like animation data or text. Here we summarize the most relevant areas of related work.

#### *Speech-Based Animation*

Many efforts focus on the problem of automatic lip sync, also known as speech-based animation of digital characters. Most solutions fall into one of three general categories: procedural techniques that use expert rules to convert speech into animation; database (or unit selection) approaches that repurpose previously captured motion segments or video clips to visualize new speech input; and model-driven methods that learn generative models for producing lip sync from speech.

While some of these approaches achieve impressive results, the vast majority rely on accurate text or phone labels for the input speech. For example, the recent JALI system by Edwards et al. [74] takes a transcript of the speech as part of the input, and many other methods represent speech explicitly as a sequence of phones [161, 98, 54, 201, 297, 296, 177]. A text or phone-based representation is beneficial because it abstracts away many idiosyncratic characteristics of the input audio, but generating an accurate transcript or reliable phone labels is very difficult to do in real-time, with small enough latency to support live animation applications. The most responsive real-time speech-to-text (STT) techniques typically require several seconds of lookahead and processing time [313], which is clearly unacceptable for live interactions with animated characters. Our approach foregoes an explicit translation into phones and learns a direct mapping between low-level audio features and output visemes that can be applied in real-time with less than 200ms latency.

Another unique aspect of our problem setting is that we focus on generating discrete viseme sequences. In contrast, most previous lip sync techniques aim to produce “realistic” animations where the mouth moves smoothly between poses. Some of these methods target rigged 3D characters or meshes with predefined mouth blendshapes that correspond to speech sounds [325, 155, 297, 74, 206, 288], while others generate 2D motion trajectories that can be used to deform facial

images to produce continuous mouth motions [39, 30]. As noted earlier, discrete 2D lip sync is not designed to be smooth or realistic. Animators use artistic license to create viseme sequences that capture the essence of the input speech. Operationalizing this artistic process requires different techniques and different training data than previous lip sync methods that aim to generate realistic, continuous mouth motions. In the domain of discrete 2D lip sync, one relevant recent system is Voice Animator [105], which uses a procedural technique to automatically generate so-called “limited animation” style lip sync from input audio. While this work is related to ours, it generates lip sync with only 3 mouth shapes (closed, partly open, and open lip). In contrast, our approach supports a 12-viseme set that is typical for most modern 2D animation styles. In addition, Voice Animator runs on pre-recorded (offline) audio.

Despite these differences in the goals and requirements of previous published lip sync methods, recent model-driven techniques for generating realistic lip sync have shown the promise of learning speech-to-animation mappings from data. In particular, the data-driven method of Taylor et al. taylor2017deep suggests that neural networks can successfully encode the relationships between speech (represented as phones sequences) and mouth motions. Our work explores how we can use a recurrent network that takes advantage of temporal context to achieve high-quality live 2D lip sync.

### *Speech Analysis*

Our goal of converting raw audio input into a discrete sequence of (viseme) labels is related to classical speech analysis problems like STT or automatic speech recognition (ASR). For such applications, recurrent neural networks (primarily in the form of LSTMs) have proven very successful [113, 332, 114]. In our approach, we use a basic LSTM architecture, which allows our model to leverage temporal context from the input audio stream to predict output visemes. However, the low-latency requirements of our target application require a different LSTM configuration than many STT or ASR models. In particular, we cannot rely on any significant amount of future information, which precludes the use of bidirectional LSTMs [112, 98]. In addition, the lack of existing large corpora of hand-animated 2D lip sync data (and the high cost of collecting such data) means

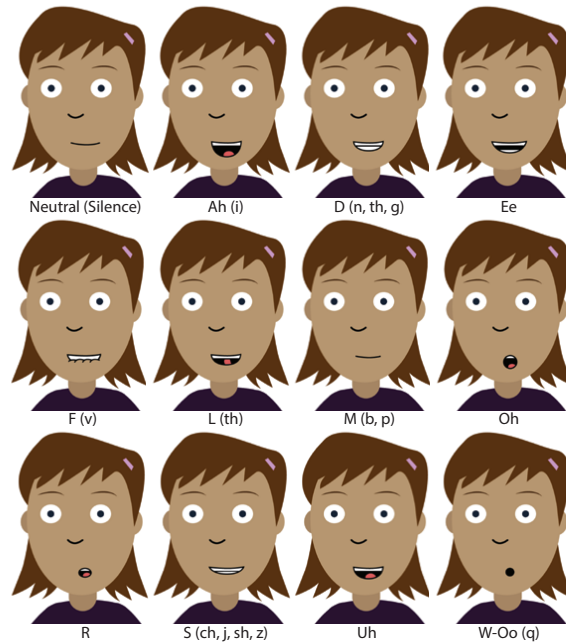


Figure 5.2: Chloe’s Viseme Set. Additional associated sounds in parentheses.

that we cannot rely on training sets with many hours of data, which is the typical amount used to train most STT and ASR models. On the other hand, our output domain (a small set of viseme classes) is much more constrained than STT or ASR. By leveraging the restricted nature of our problem, we achieve a low-latency model that requires a modest amount of data to train.

## 5.2 Approach

We formulate the problem of live 2D lip sync as follows. Given a continuous stream of audio samples representing the input speech, the goal is to automatically output a corresponding sequence of visemes. We use the 12 viseme classes defined by Ch (see Figure 5.2), which is similar to other standard viseme sets in both commercial tools (e.g., ToonBoom [307], CrazyTalk [59] and previous research [74, 91, 40, 178]).

In addition to being accurate, the technique must satisfy two main requirements. First, the method must be fast enough to support live applications. As with any real-time audio processing pipeline, there will necessarily be some latency in the lip sync computation. For instance, simply

converting audio samples into standard features typically requires frequency analysis on temporal windows of samples. To prevent viseme changes from appearing “late” with respect to the speech, live animation broadcasts often delay the audio slightly. The size of the delay must be large enough to produce a good audio-visual alignment where viseme changes occur simultaneously with the audio changes. In fact, some animation literature suggests timing viseme transitions slightly early (1–2 frames at 24fps) with respect to the audio [301]. At the same time, the delay must be small enough to enable natural interactions with other actors and the audience without awkward pauses in the animated character’s responses. We consulted with several live animation production teams and found that 200–300ms is a reasonable target for live lip sync latency; e.g., the live Simpsons broadcast delayed Homer’s voice by 500ms [97] and livestreams often use a 150–200ms audio delay.

The second requirement involves training data. As noted earlier, data-driven methods have proven very successful for various speech analysis problems. However, supervised training data (i.e., hand-authored viseme sequences) is extremely time-intensive to create; we obtained quotes from professional animators estimating five to seven hours of animation work to lip sync each minute of speech. As a result, it is difficult to obtain very large training corpora of hand-animated results. For example, collecting the equivalent amount of training data used by other recent audio-driven models like Suwajanakorn et al. [288] (17 hours) and Taylor et al. [296] (8 hours) would be extremely costly. We aim for a method that requires an order of magnitude fewer data.

Given these requirements, we developed a machine learning approach that generates live 2D lip sync with less than 200ms latency using 13–20 minutes of hand-animated training data. We leverage a compact recurrent model with relatively few parameters that incorporates a small but useful amount of lookahead in both the input feature descriptor and the configuration of the model itself. We also describe a simple data augmentation scheme that leverages the inherent structure of the TIMIT speech dataset to amplify hand-animated viseme sequences by a factor of four. The following sections describe our proposed model and training procedure.

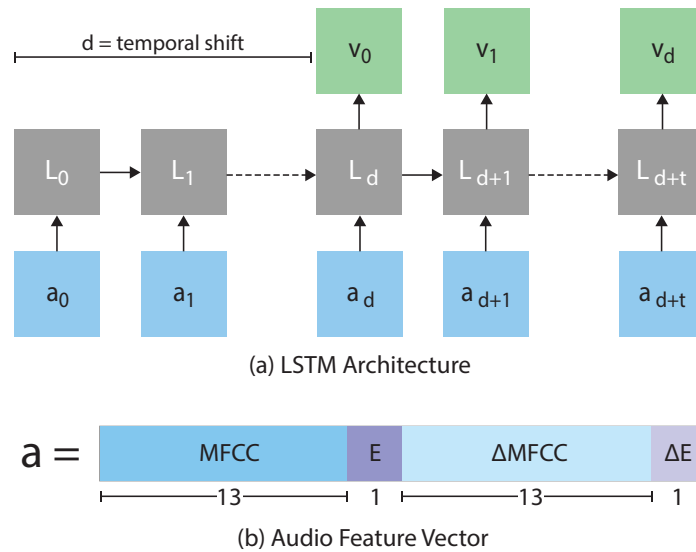


Figure 5.3: Lip Sync Model. We use a unidirectional single-layer LSTM with a temporal shift  $d$  of 6 feature vectors (60ms) (a). The audio feature  $a$  consists of MFCC, log mean energy, and their first temporal derivatives (b).

### 5.2.1 Model

Based on the success of recurrent neural networks in many speech analysis applications, we adopt an LSTM architecture for our problem. Our model takes in a sequence of feature vectors  $(a_0, a_1, \dots, a_N)$  derived from streaming audio and outputs a corresponding sequence of visemes  $(v_0, v_1, \dots, v_N)$  (see Figure 5.3a). The latency restrictions of our application preclude the use of a bidirectional LSTM. Thus, we use a standard unidirectional single-layer LSTM with a 200-dimensional hidden state that is mapped linearly to 12 output viseme classes. The viseme with the maximum score is the model prediction. We note that our initial experiments explored the use of Hidden Markov Models (HMMs) to convert audio observations into visemes, but we found it challenging to pre-define a hidden state space that captures the appropriate amount of temporal context. While the overall configuration of our LSTM does not deviate significantly from previous work, there are a few specific design decisions that were important for getting the model to perform well.

### *Feature Representation*

While it is possible to train a model that operates directly on raw audio samples, most speech analysis applications use mel-frequency cepstrum coefficients (MFCCs) [314] as the input feature representation. MFCCs are a frequency-based representation with non-linearly spaced frequency bands that roughly match the response of the human auditory system. In our pipeline, we process the input audio stream by computing MFCCs (with 13 coefficients) on a sliding 25ms window with a stride of 10ms (i.e., at 100Hz), which is a typical setup for many speech processing techniques. Before computing MFCCs, we compress and boost the input audio levels using the online Hard Limiter filter in Adobe Audition, which runs in real-time.

In addition to the raw MFCC values, some previous methods concatenate derivatives of the coefficients to the feature representation [88, 115, 288]. Such derivatives are particularly important for our application because viseme transitions often correlate with audio changes that in turn cause large MFCC changes. One challenge with such derivatives is that they can be noisy if computed at the same 100Hz frequency as the MFCCs themselves. A standard solution is to average derivatives over a larger temporal region, which sacrifices latency for smoother derivatives. We found that estimating derivatives using averaged finite differences between MFCCs computed two windows before and after the current MFCC window provides a good tradeoff for our application. An additional benefit of this derivative computation is that it provides the model with a small amount of lookahead since each feature vector incorporates information from two MFCC windows into the future. In our experiments, we found that the energy of the audio signal can sometimes be a useful descriptor as well. Thus, we add the log-energy and its derivative as two additional scalars to form a 28-dimensional feature (see Figure 5.3b).

### *Temporal Shift*

Since LSTMs can make use of history, our model has the ability to learn how animators map a sequence of sounds to one or more visemes. However, we found that using past information alone was not sufficient and resulted in chattery viseme transitions. One potential reason for these prob-

lems is that, as noted above, animators often change visemes slightly ahead of the speech [301]. Thus, depriving the model of any future information may be eliminating important audio cues for many viseme transitions. To address this issue, we simply shift which viseme the model predicts with respect to the input audio sequence. In particular, for the current audio feature vector  $x_t$ , we predict the viseme that appears  $d$  windows in the past at  $x_{t-d}$  (see Figure 5.3a). In other words, the model has access to  $d$  future feature vectors when predicting a viseme. We found that  $d = 6$  provides sufficient lookahead. Adding this future context does not require any modifications to the network architecture, although it does add an additional 60ms of latency to the model.

### *Filtering*

Our model outputs viseme predictions at 100Hz. For live animation, the target frame rate is typically 24fps. We apply two types of filtering to convert the 100Hz output to 24fps.

**Removing noise from predictions.** Our model is generally able to predict good viseme sequences. However, at 100Hz, we occasionally encounter spurious noisy predictions. Since these errors are typically very short in duration, we use a small lookahead to filter them out. For any viseme prediction that is different from the previous prediction (i.e., a viseme transition), we consider the subsequent three predictions. If the new viseme holds across this block, then we keep it as-is. Otherwise, we replace the new viseme with the previous prediction. This filtering mechanism adds 30ms of latency.

**Removing 1-frame visemes.** After removing noise from the 100Hz model predictions, we subsample to produce visemes at the target 24fps rate. As a rule, animators never show a given viseme for less than two frames. To enforce this constraint, we do not allow a viseme to change after a single frame. This simple rule does not increase the latency of the system since it just remembers the last viseme duration.

These mechanisms reduce flashing artifacts that sometimes arise when directly subsampling the 100Hz model output.

### 5.2.2 Training

Training our lip sync model requires pairs of input speech recordings with output hand-animated viseme sequences. For each input recording, we compute the corresponding sequence of audio feature vectors, run each vector through the network to obtain a viseme prediction, and use back-propagation through time to optimize the model parameters. We use cross-entropy loss to penalize classification errors with respect to the hand-animated viseme sequence. The ground truth viseme sequences are animated at 24fps, so we upsample them to match the 100Hz frequency of our model.

#### *Data Augmentation*

In order for the model to learn the relationships between speech sounds and visemes, the training data should cover the full spectrum of phones and common transitions. Moreover, since we want our model to generalize to arbitrary input voices, it is important for the training set to include a large diversity of speakers. However, as noted above, hand-animated lip sync data is extremely expensive to generate, which makes it difficult to collect a large collection of input-output pairs that exhibit both phonetic and speaker diversity.

To address this problem, we leverage a simple but important insight. We do not have to treat phonetic and speaker diversity as separate, orthogonal properties. If we select a set of phonetically diverse sentences and record multiple different speakers reading each sentence, then we can obtain a corpus of speech examples that is diverse along both axes but with a useful structure that we can exploit for data augmentation. In particular, if we manually specify the lip sync for one speaker's recording of a given sentence, then it is likely the case that the same sequence of visemes could be used to obtain a good lip sync result for the other recordings of the sentence, provided that we can align the visemes temporally to each recording. Fortunately, the TIMIT dataset, which has been used successfully to train many speech analysis models, has exactly this structure. The subset of 450 unique *SX* sentences in TIMIT is designed to be compact and phonetically diverse, and the corpus includes 7 recordings of each sentence by different speakers. Overall, the recordings span 630 speakers and 8 dialects.

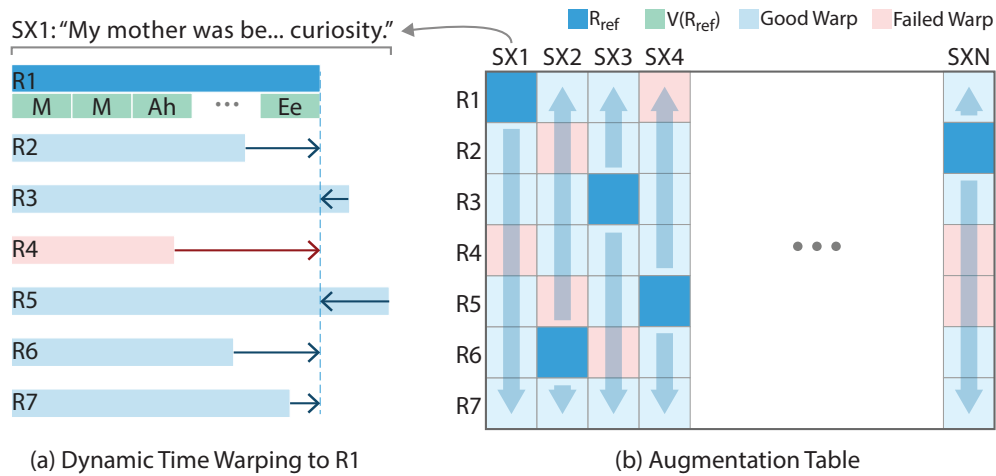


Figure 5.4: Data Augmentation. Each reference recording has an associated hand-animated viseme sequence. We automatically time warp other recordings of the same sentence to align with each reference recording (a). This procedure allows us to create new input-output training pairs for every successfully warped recording.

Based on this insight, our data augmentation works as follows. We select a collection of reference recordings of unique  $SX$  sentences and obtain the corresponding hand-animated viseme sequences.

For each reference recording  $R_{\text{ref}}$ , we apply dynamic time warping [20] to align all other recordings of the same sentence to  $R_{\text{ref}}$  (Figure 5.4a). We use the warping implementation in the Automatic Speech Alignment feature of Adobe Audition. Since warping generally works better from male-to-male and female-to-female voices, we only run the alignment between recordings with the same gender. To filter out cases where the alignment fails, we discard any warped recordings whose durations are significantly different from  $R_{\text{ref}}$ . Finally, we associate each  $R_{\text{ref}}$  and the successfully aligned recordings with the same hand-animated viseme sequence  $V(R_{\text{ref}})$  to use as training pairs for our model (Figure 5.4b). This fully automated procedure allows us to augment our data by roughly a factor of 4 based on the distribution of male-female speakers and the success rate of the Automatic Speech Alignment.

### *Selecting Batches*

The TIMIT corpus consists of 450 phonetically-compact sentences ( $SX$ ), 1890 phonetically-diverse sentences ( $SI$ ) and 2 dialect “shibboleth” sentences ( $SA$ ). The dataset is partitioned into training and test sets. Of the 450  $SX$  sentences, 330 sentences are in the training set  $SX_{\text{train}}$ . Since the  $SX$  sentences are already designed to provide good coverage of phone-to-phone transitions (with an emphasis on phonetic contexts that are considered difficult or particularly interesting for speech analysis applications), we could generate our training data by simply choosing one recording for every sentence in  $SX_{\text{train}}$  and obtaining a corresponding viseme sequence. However, we wanted to partition our training data into equivalent batches in order to run experiments evaluating how different amounts of data affect the performance of our model. To do this, we first scored all the  $SX_{\text{train}}$  recordings by counting the number of distinct individual phones and phone-to-phone transitions in each recording, using the phone transcriptions provided by the TIMIT dataset. For each sentence, we chose the male and female recordings with the maximum scores. Then, we generated batches of recordings by choosing subsets that include similar distributions of high and low scoring recordings and an even mix of male and female speakers. In the end, we produced six batches of 50  $SX$  recordings which we used for training our models. For our validation set, we also created a batch of 50 recordings with a random distribution of recordings from  $SI_{\text{train}}$ ,  $SA_{\text{train}}$ , and the subset of  $SX_{\text{train}}$  sentences not used in any of the previously-generated six training batches. We obtained hand-animated viseme sequences for all seven batches.

### *5.2.3 Model Latency*

At prediction time, the inherent latency of our model comes from the lookahead in the feature vector computation (33ms), the temporal shift between the input audio and output viseme predictions (60ms), and the 100Hz filtering, which takes into account future viseme predictions (30ms). In total, this amounts to 123ms between the time an audio sample arrives in the input stream and when the corresponding viseme is predicted. As noted earlier, animators sometimes show visemes slightly before the corresponding sounds (usually one to two frames at 24fps, or 40-80ms). The

processing time required to run audio samples through our entire pipeline, including the Hard Limiter filter before we compute feature vectors, is 1–2ms measured on a 2017 MacBook Pro laptop with a 3.1GHz Intel Core i5 processor and 8GB of memory. Thus, the total latency in the system is approximately 165-185ms.

### 5.3 Experiments

We conducted several experiments to understand the behavior of our model and the impact of our main design choices. For this quantitative analysis, we compute the per-frame accuracy of the viseme prediction at 24fps, after the filtering step in our pipeline.

#### 5.3.1 Datasets

We collected training data by hiring two professional animators (A1, A2) to lip sync a set of speech recordings using Character Animator. For consistency, they all used the default Chloe character that comes with the application. Chloe includes the same set of 12 visemes that our model uses (see Figure 5.2). We gave A1 and A2 seven batches of recordings each (six for training, one for validation), which we generated as described in Section 5.2.2 The six training batches represented about 20 minutes of speech in total. After propagating the hand-generated viseme sequences to the aligned *SX* recordings using our data augmentation procedure, we obtained approximately 80 minutes of training data per animator.

To gain more insight on the differences in lip sync style, we recruited two other animators (A3, A4) and asked all four to lip sync an additional 27 TIMIT recordings (25 from the *SX* recordings and 2 from the *SA* recordings in TIMIT). These results allow us to analyze how different animators time transitions and choose visemes for the same recordings.

#### 5.3.2 Differences in Style

The statistics of the viseme sequences generated by the four different animators for the same 27 recordings reveal clear differences in lip sync style. In terms of overall viseme choices, different

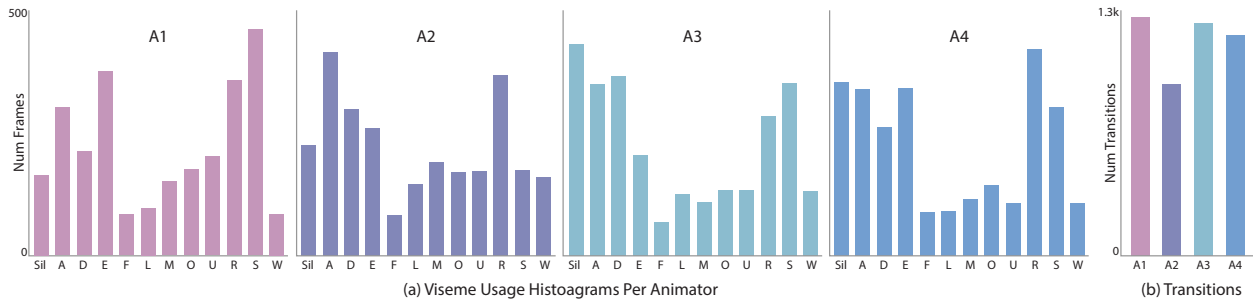


Figure 5.5: Analysis of Lip Sync Styles. Histograms of viseme usage (a) and raw transition counts (b) show that different animators prefer different visemes and aim for different levels of articulation.

animators used different distributions of visemes (Figure 5.5a) and also changed visemes at different rates (Figure 5.5b). For example, A1 and A2 used the Silent viseme far less than A3 and A4, which suggests that they prefer sequences that do not return to the neutral mouth pose. A1 also liked to use the S viseme much more than others. In terms of viseme changes, A2’s relatively low overall transition count suggests that the animator preferred a smoother, less articulated style.

### 5.3.3 Accuracy and Convergence Behavior

We trained separate models using the full datasets that we collected from A1 (*OursA1*) and A2 (*OursA2*). We used the last batch of 50 hand-animated sentences as the validation set and trained on the data from the six *SX* batches. All the networks are trained using the Torch framework [56] until convergence (200 epochs) using the Adam optimizer [164], with a dropout ratio of 0.5 for regularization to avoid overfitting, batch size of 20, and learning rate of 0.001. On a single NVIDIA GTX-1080 GPU, training took less than 30 minutes. For the output layers, we used the softmax activation function for 12 viseme output classification and the cross-entropy error function to compute the classification accuracy. The per-frame viseme prediction accuracy for *OursA1* is 64.37% and *OursA2* is 66.84%.

### 5.3.4 Impact of Lookahead

To evaluate the importance of using future information (albeit a small amount) in our approach, we trained a version of *OursA2* with no temporal shift between observations (feature vectors) and predictions (visemes) and modified the feature vector to include derivatives computed using past MFCC windows only. The per-frame accuracy for the no-lookahead version of *OursA2* is 59.27%, which is significantly lower than the accuracy of *OursA2* (66.84%) which is trained with temporal shift ( $d=6$ ) and using two future windows for MFCC derivative computation. From a qualitative perspective, we notice that the model without lookahead appears to be chattery, with extra transitions around the expected viseme changes.

### 5.3.5 Impact of LSTM context

One advantage of using an LSTM over a non-recurrent network (e.g., the sliding window CNN of Taylor et al. [296]), is that LSTMs can leverage a larger amount of (past) context without increasing the size of the feature vector. While longer feature vectors can cover more past context, they result in larger networks that in turn require more data to train. To investigate how much context our model actually uses for viseme prediction, we trained different versions of *OursA2* with data that artificially limits the amount of context the LSTM can leverage. Our initial experiments showed that the model performance does not improve with more than one second of context, so we segmented the A2 training data into uniform chunks of several durations (200ms, 400ms, 600ms, 800ms, 1sec) and trained our LSTM on each of these five datasets. The per-frame viseme prediction accuracies are 24.63%(200ms), 37.08%(400ms), 56.44%(600ms), 59.72%(800ms) and 64.81%(1sec). The significant increase in accuracy around 600ms suggests that our model is mainly using around 600–800ms of context, which corresponds to 60–80 MFCC windows. In other words, these results suggest that a non-recurrent model may need to use much longer feature vectors (and thus, much more training data) to achieve comparable viseme prediction results.

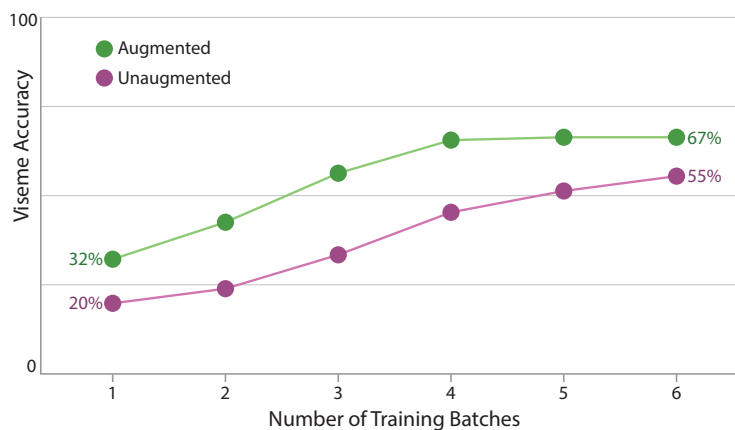


Figure 5.6: Impact of Data Augmentation. Augmenting the data results in a significant increase in accuracy, with diminishing returns after four augmented batches.

### 5.3.6 Impact of Data Augmentation

Finally, we investigate the effect of our data augmentation technique by training versions of *OursA2* with various amounts of data. Specifically, we consider an unaugmented dataset that only has the hand-animated viseme sequences, and our full augmented dataset. We divide the A2 training data into increasing subsets of the 6 hand-animated batches and train the model on both the unaugmented and augmented subsets. As expected, our data augmentation allows us to achieve much higher accuracy for the same amount of animator work (see Figure 5.6). Moreover, there is a clear elbow in the accuracy for the augmented data at around 4 batches, which corresponds to roughly 13 minutes of hand-animated lip sync. In other words, an animator may only need to provide this amount of data to train a new version of our lip sync model. We further validate this claim in Results Section with human judgement experiments that compare the full model with the version trained using 4 augmented batches.

## 5.4 Results

To evaluate the quality of our live lip sync output, we collected human judgements comparing our results against several baselines, including competing methods, hand-animated lip sync, and



Figure 5.7: Characters. We used Chloe and the Wizard for our human judgement experiments, and we show lip sync results with the other characters in our video summary and supplemental materials.

different variations of our model. In informal pilot studies, we saw a slight preference for A2’s lip sync style over A1, so we used the *OursA2* results for these comparisons. We also conducted a small preliminary study comparing the stylistic differences between *OursA1* and *OursA2* results.

In addition to these comparisons, we applied our lip sync model to several different 2D characters (see Figure 5.7) that come bundled with Character Animator. Our video summary and supplemental materials show representative lip sync results using these characters. We also include real-time recordings that shows the system running live in a modified version of Ch. For these recordings, we delay the audio track by 200ms to account for the latency of our model. As noted earlier, this type of audio delay is standard practice for live animation broadcasts.

#### 5.4.1 Comparisons with Competing Methods

We are not aware of any previous research efforts that directly support 2D (discrete viseme) lip sync for live animation. Thus, we compared our method against existing commercial systems. The predominant tool for live 2D animation (including live lip sync) is Character Animator (Ch), which

was used for the live Simpsons episode, the recurring live animation segments on The Late Show, and to our knowledge, all of the recent live animated chat sessions on Facebook and YouTube. In addition to live lip sync, Ch also includes a higher quality offline lip sync feature. For traditional non-live cartoon animation, ToonBoom (TB) is an industry standard tool that also provides offline lip sync. We compared our results using A2’s model (*Ours*) against the Ch online lip sync (*ChOn*), and the offline output from both Ch (*ChOff*) and TB (*TBOff*).

### *Procedure*

To compare our model against any one of the competing methods, we selected a test dataset of recordings, and for each one we generated a pair of viseme sequences using the two lip sync algorithms. We applied the lip sync to two characters (Chloe and the Wizard, shown in Figure 5.7) that are drawn in distinct styles with visemes that look very different. For each character, we presented pairs of lip sync results to users and asked which one they prefer. We used Amazon Mechanical Turk (AMT) to collect these judgements. Based on pilot studies, we found that showing the lip sync results side-by-side with separate play controls made it easy for users to review and compare the output. Since our method uses the same set of visemes as Ch, we were able to generate direct comparisons between our model and both the online and offline Ch algorithms. TB uses a smaller set of eight visemes for their automatic lip sync. To generate comparable results, we mapped a subset of our viseme classes (S- $\dot{\iota}$ D, L- $\dot{\iota}$ D, Uh- $\dot{\iota}$ Ah, R- $\dot{\iota}$ W-Oo) to the TB visemes based on TB’s published phone-to-viseme guide and then used this subset to generate lip sync from TB. For our model, we mapped each viseme that is not in the TB subset to one of the TB visemes and used this mapping to project our lip sync output to the TB subset.

### *Test Set*

For our test dataset, we randomly chose 25 recordings from the TIMIT test set, using the same criteria as our training batch selection process to ensure even coverage of phones and transitions. To increase the diversity of our test set, we composed an additional 10 phonetically diverse sentences

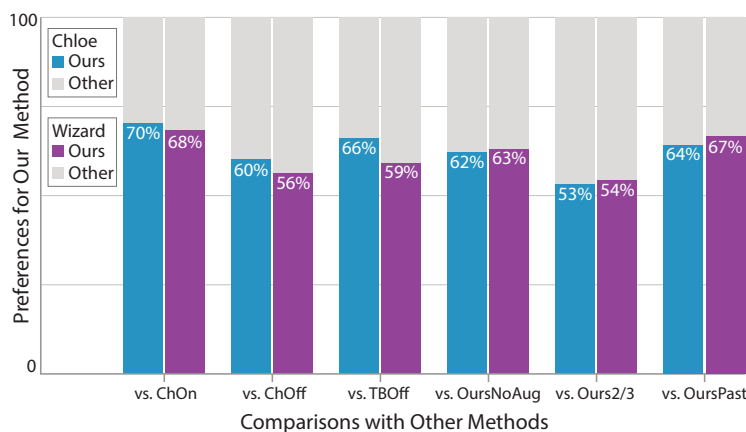


Figure 5.8: Human Judgements. Our method was significantly preferred over all commercial tools, including offline methods. Our full model was also preferred over versions trained with no augmented data (*OursNoAug*) and two thirds of the augmented data (*Ours2/3*). However, the preference over *Ours2/3* was quite small, which suggests that this amount of data may be sufficient to train an effective model.

and recorded a man, woman and child reading each one. We also recorded a voice actor reading each sentence in a stylized cartoon voice. We randomly chose 25 of these non-TIMIT recordings for testing. All test recordings were between 3–4 seconds. None of these recordings were used for training. We used the same test set and procedure for all the comparisons described in the following sections.

### Findings

We collected 20 judgements for every recording (10 for each puppet), which resulted in 1000 judgements for each competing method. The left side of Figure 5.8 summarizes the results of the comparisons with Ch and TB. Our lip sync was preferred in all cases, and these differences were statistically significant (at  $p = 0.05$ ) based on the Binomial Test. We are especially encouraged that our results outperformed even the offline Ch and TB methods, which do not support live animation. Moreover, we did not see much difference between the results for the non-TIMIT versus TIMIT

test recordings, which suggests that our model generalizes to a broader spectrum of speakers. Qualitatively, we found the *ChOff* and *TBOff* results to be overly smooth (i.e., missing transitions) in many cases, while the *ChOn* output tends to be more chattery. We also saw a few cases where *TBOff* uses visemes that clearly do not match the corresponding sound. Our submission video shows several direct comparisons that highlight these differences.

#### 5.4.2 Comparisons with Different Model Variations

Our data augmentation experiments (Impact of Data Augmentation Section) suggest that our model should already perform well using just four out of the six hand-animated training batches. To validate this conjecture, we compared the output of our full model (*Ours*) against a version trained with four augmented batches of hand-animated data (*Ours2/3*). As a baseline, we also compared *Ours* with a model trained on all six batches *without* data augmentation (*OursNoAug*). Similarly, we compared the output of our no-lookahead model (*OursPast*) to *Ours* in order to validate the impact of lookahead on the perceived quality of the resulting lip sync.

#### Findings

The right side of Figure 5.8 shows the comparison results for the different versions of our model. Not surprisingly, *Ours* is clearly preferred over *OursNoAug*. The lack of augmented data results in lip sync with both incorrect viseme choices and a combination of missing and extraneous transitions. On the other hand, the preferences between *Ours* and *Ours2/3* are much more balanced, which suggests that we may only require about four batches (13 minutes) of hand-animated data to train an effective live lip sync model. *Ours* was also distinctly preferred over *OursPast* showing the benefit of the small amount of lookahead in our full model.

#### 5.4.3 Matching Animator Styles

While most high quality lip sync shares many characteristics, there are some stylistic differences across different animators, as noted earlier. To investigate how well our approach captures the

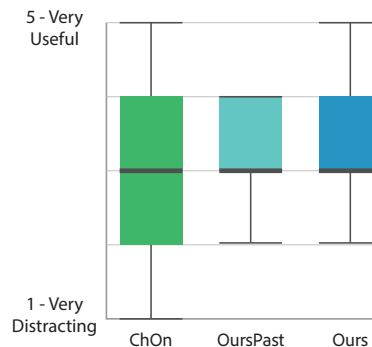


Figure 5.9: The rankings from our user study organized by system. The thick grey middle line is the median, the box represents 50% of the data between the first and third quartiles, and the whiskers mark the extremes.

style of the training data, we conducted a small experiment comparing the outputs of *OursA1* and *OursA2*. We randomly chose 19 hand-animated viseme sequences from each animator that were not part of the training sets for the two models. For each hand-animated result, we generated lip sync output from *OursA1* and *OursA2* using the corresponding speech recording and then presented the two automatic results to the animator along with their own hand-animated sequence as a reference. We then asked the animator to pick which of the model-generated results most resembled the reference. We used the Chloe character for this experiment.

### *Findings*

Each animator chose the “correct” result (i.e., the one generated by the model trained on their own lip sync data) more often than the alternative. A1 chose correctly in 12/19 and A2 chose correctly in 15/19 comparisons. While these are far from conclusive results, they suggest that our model is able to learn at least some of the characteristics that distinguish different lip sync styles.

#### 5.4.4 Impact on Performers

The experiments described above evaluate the quality of lip sync that our model outputs as judged by people who are viewing the animation. We also wanted to gather feedback on how our lip sync techniques affect *performers* who are controlling live animated characters with their voices. In particular, we wondered whether the improved quality of our lip sync or the small amount of latency in our model would have an impact (positive or negative) on performers. To this end, we conducted a small user study with nine participants comparing three lip sync algorithms: *ChOn*, *OursPast*, and *Ours*. To minimize the differences between the conditions, we implemented *OursPast* and *Ours* within Ch. We used a within-subject design where each participant used all three conditions (with the order counterbalanced via a 3x3 Latin square) to control Chloe’s mouth movements. To simulate a live animation setting, we asked each participant to answer 6 questions (two per condition) as if they were being interviewed as Chloe. During the performance, we used the relevant lip sync method to show the participant live feedback of Chloe’s mouth being animated. At the end of the session, we asked participants to rate each condition based on the effectiveness of the live feedback, on a scale from 1 (very distracting) to 5 (very useful) . We also solicited freeform comments on the task. Each session lasted roughly 20 minutes.

#### Findings

We summarize the collected ratings for each condition using a box and whisker plot, as shown in Figure 5.9. The data does not show any discernible difference in how participants rated the usefulness of the live feedback across the different algorithms. In particular, the latency of our full model did not have a noticeable negative impact on the performers. The comments from participants suggest that the cognitive load of performing (e.g., thinking of how to best answer a question) makes it hard to focus on the details of the live feedback. In other words, the results of this study suggest that the quality of live lip sync is mainly relevant for viewers (as shown in our human judgement experiments) rather than performers.

## **5.5 Limitations**

There are two main limitations with our current method that stem from our source of training data. The TIMIT recordings all contain clean, high-quality audio of spoken sentences. As a result, our model performs best on input with similar characteristics. While this is fine for most usage scenarios, there are situations where the input audio may contain background noise or distortions due to the recording environment or microphone quality. For example, capturing speech with the onboard microphone of a laptop in an open room produces noticeably lower quality lip sync output than using even a decent quality USB microphone in a reasonably insulated space. Note that the production teams for almost all live broadcasts already have access to high end microphones and sound booths, which eliminates this problem. In addition, we noticed that vocal input that is very different from conversational speech (e.g., singing, where vowels are often held for long durations) also produces suboptimal results.

We do not believe these are fundamental limitations of our approach. For example, we could potentially collect more training data or, better yet, employ additional data augmentation techniques to help the model learn how to better handle a wider range of audio input. To support singing, we may also need to include slightly different audio features. Of course, we would need to conduct additional experiments to confirm these conjectures.

## **5.6 Summary**

Our work addresses a key technical challenge in the emerging domain of live 2D animation. Accurate, low-latency lip sync is critical for almost all live animation settings, and our extensive human judgement experiments demonstrate that our technique improves upon existing state-of-the-art 2D lip sync engines, most of which require offline processing. Thus, we believe our work has immediate practical implications for both live and even non-live 2D animation production.

## Chapter 6

# CONVERSATIONAL STYLE MATCHING AGENT

### 6.1 Introduction

#### 6.1.1 Motivation

Personal assistants (e.g., Alexa, Siri, Cortana, and Google Now) and “bots” interacting in natural language (e.g., Messenger, Skype, and Sina Weibo), have created new platforms for human-computer interaction. In the U.S., nearly 50 million (or 1 in 5) adults are estimated to have access to a smart speaker (e.g., Amazon Echo) for which voice is the primary interface. Many more have access to an assistant on their smartphones or smartwatches.

Despite the prevalence of these devices and the considerable investment in research and development devoted to them, it is still not very natural to engage with AI agents in extended interactions and open-ended conversations. In part, this is because the agents are unable to have multi-turn conversations with the users and fail to adapt to the social behaviors of humans, although some work on improving interactions has been done with embodied agents [84, 111]. This large gulf in expectations is perhaps part of the reason why conversational agents are only used for very simple tasks and often disappoint users [195].

While agents’ abilities to engage in dialogue has been studied quite extensively [22, 308], the conversational style of these agents has received less attention. However, it has been shown that people’s perceptions of conversational agents are influenced by the interaction styles of the agents [24, 203, 243]. If an agent follows more of the social norms used by people, the conversation should be considered more natural by those that interact with the agent. Niederhoffer and Pennebaker have done studies that show that people align their linguistic style in social interactions [223]. Linguistic mimicry, the adoption of another’s language style, is a behavior that has social benefits [232] where greater empathy is associated with mimicry in conversations. Therefore, designers of an artificially

empathetic conversational agent explicitly identified conversational style matching as a desirable feature; however, it was not included in their published design [218]. In other words, style matching in conversational agent design warrants further exploration.

The conversational style includes prosody, word choice, and timing, for example. We distinguish style (the “how”), from any topical information transferred (the “what”); we can provide the same information with different styles. Prior work defined conversational style parameters amenable to computation and that were able to resemble style matching mechanisms between humans [302]; we were influenced by this work. The purpose of the current study was to understand if an intelligent, conversational agent could perform conversational style matching “on-the-fly” and, if so, how this affects the perception of such an agent.

Designing an end-to-end conversational agent is complex. It requires speech recognition, dialogue generation, and speech synthesis. Furthermore, all these need to operate in real-time, without any unnatural delay in the timing of responses between the agent and the human. We designed a conversational agent that is capable of multi-turn dialogue without human intervention. Via a generative neural language model, the agent is capable of relatively open-ended discourse and can respond to utterances from the interlocutor in an understandable way. By combining this with automatic speech, paralinguistic sensing capabilities, and prosodic control of the agent’s synthesized speech, we created a novel conversational agent system that is capable of real-time style matching.

The contributions of this work are as follows:

- We present an end-to-end automated conversational agent capable of conversational style matching.
- We examine the effect of conversational style matching on perception where we show that conversational style increases a user’s trust in an agent, compared to an agent without conversational style matching.
- We provide design guidelines for conversational systems with conversational style matching.

Methods and results presented in this chapter were previously published in [7] and [124].

### 6.1.2 *Related Literature*

#### *Conversational style*

In terms of understanding human-to-human conversation, Deborah Tannen's theory on conversational style is the most widely used [294, 295]. Tannen defines style as "the use of specific linguistic devices, chosen by references to broad operating principles or conversational strategies." Tannen's conversational style framework categorizes styles on an axis ranging from High Consideration (HC) to High Involvement (HI). An HC interaction style emphasizes consideration and independence. It is characterized by long pauses and hesitations and the use of moderate paralinguistic characteristics. An HI interaction style, on the other hand, emphasizes interpersonal involvement and interest. It is characterized by speaking faster and more loudly, overlapping with the other speaker, and with fewer pauses between conversational turns.

Most of the related research on conversational style matching is based upon the work of Tannen, who described several markers that define conversational style [295]. These markers include: Topic (e.g., personal pronoun usage, persistence), Pace (e.g., speech rate, pauses), Expressive Paralinguistics (e.g., pitch and loudness shifts) and Genre (e.g., story framing). Based on the usage of these stylistic markers, people can be placed on an axis ranging from HC to HI.

In more recent work, Shamekhi et al. looked at conversational style in human-agent interactions [277]. They examined whether there was a particular conversational style (HI or HC) that was preferred for a conversational agent in a structured conversation. However, rather than finding one specific style that worked best, they found that participants liked whichever agent matched their own conversational style.

Thomas et al. looked at the conversational style of participants while performing simple information-seeking task [303, 302]. This task was similarly structured to how a person might use an intelligent agent such as Cortana or Siri. They found that even in these tasks, participants aligned their conversational styles over time. It took less effort to complete tasks for participants who aligned their style compared to those who did not. To summarize, people prefer partners that match their style and actively align style during the conversation.

Conversational style matching can be seen as a type of entrainment, as it relates to people synchronizing their behavior within an interaction [122, 19]. Entrainment has been used in related work to generate more realistic and accurately timed reactions [143] or backchannels [134]. Levitan et al. implemented acoustic-prosodic entrainment in a conversational avatar and observed an increase in the perceived reliability and likability of the system [179]. In a text-chat interface, Scissors et al. [270, 269] found that lexical entrainment was associated with trust between partners. Specifically, pairs with high trust exhibited greater repetition of shared terms than did pairs with lower trust [270]. Subsequent work found that this was driven primarily by certain types of terms (e.g., those with higher positive emotional valence), and that not all similarity increased trust. The similarity in negative terms was associated with decreased trust [269]. We were inspired by this work and wanted to build upon it using an automated agent capable of voice-based interactions.

### 6.1.3 Agents and Trust

Conversational Agents have been studied extensively, with trustworthiness being an important aspect of evaluating and designing these systems [139, 260]. Trust has been mentioned as an important goal for a socially intelligent agent. For example, Elofson et al. [87] investigated the role intelligent agents can play in the development of trust within organizations. An important social goal in human-human interactions is establishing trust between both people. Cassell and Bickmore argue that the methods that humans use for establishing trust can also be used by intelligent agents [42]. As shown in their studies with the embodied agent REA, by incorporating small talk, REA was perceived as more trustworthy by extroverted participants than an agent that does not engage in this social behavior [23].

In a field trial with the SARA agent, Pecune et al. found that the interactional features (e.g., frequency of interruptions and number of turns) played a role in fostering both rapport and task performance. The agent's social awareness was especially important due to the short duration of the interactions, as there is likely a social politeness boundary that should not be broken in short interactions [243]. Rapport is characterized by a close and harmonious relationship, and one would think that this quality would also strengthen trust between an agent and a human.

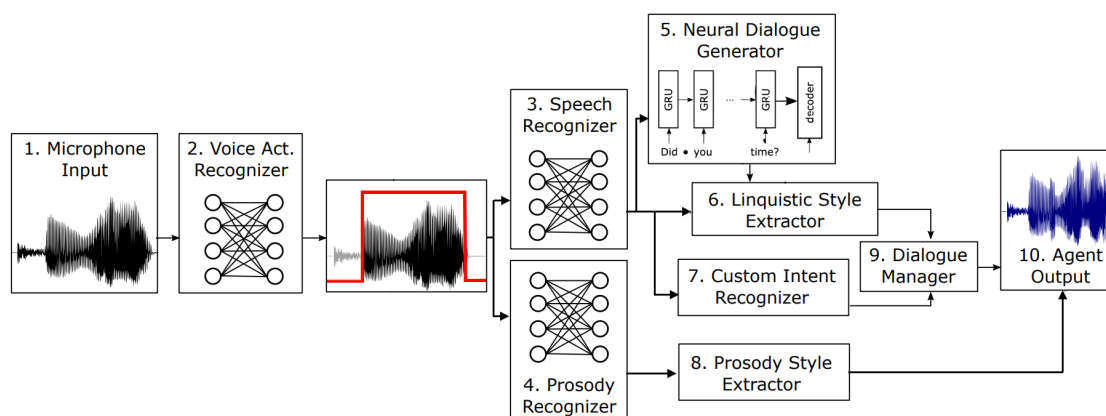


Figure 6.1: Architecture of our dialogue agent. The microphone input is first processed to extract voiced segments. The voiced segments are then passed to neural speech recognition and voice prosody recognition. The text output from the speech recognition was input to a neural dialogue generation model and also to the LUIS intent recognition engine. The recognized speech and generated dialogue response were both sent to the conversational style extractor. Finally, a rule-based dialogue manager selected the utterance to synthesize.

One reason for the link between agents and trust is that agents may sometimes be treated differently than humans. In a study, Gratch et al. found that participants disclosed more to SimSensei, a virtual interviewer for healthcare when presented as an artificial intelligence, rather than an avatar controlled by a human operator in a medical interview [110]. These results suggest that an intelligent virtual agent might be used to obtain more truthful information within medical domains.

Lucas et al. [193] used a multimodal approach to predicting trustworthiness in a negotiation context, using among other conversational features such as sentiment and word count. They found a difference between the objective trustworthiness of a negotiator and how trustworthy they were rated by their opponent (perceived trustworthiness), as people tend to be overly reliant on facial expressions to make their judgments.

## 6.2 Hypotheses

For this study, our goal was to build upon work on conversational style matching using a fully automated, voice-based interface. Based on the work of Thomas et al. [302], we know that people slowly align their conversational style while performing a task. Secondly, based on the work of Shamekhi et al. [277], we have seen that people prefer a conversational agent that matches their own conversational style.

Bolstered by these studies, we desired to verify whether an agent that performs conversational style alignment is preferred to one that does not. We expected participants to prefer this style matching behavior. Additionally, there is some evidence that HC participants are more likely to adjust their range, pitch, and intonation with their partner. We, therefore, suspect that style matching of an agent might be more useful for participants with an HC style [277].

Additionally, Bickmore and Cassell have shown that an agent's perceived trustworthiness can be positively affected by having the agent perform the social dialogue in the form of small talk [23]. We investigate whether conversational style matching further improves this effect. Therefore, our agent uses social dialogue in its interactions, and we measure the perceived trustworthiness of the agent. As such, our hypotheses for this study are as follows:

**H1a.** People prefer interacting with an agent that uses conversational style matching, as opposed to one that does not.

**H1b.** People trust an agent using conversational style matching more than one that does not.

**H2.** Preferences about an agent depend on their conversational style. In particular, people with an HC style prefer style matching.

In order to investigate these research questions, we constructed an intelligent agent that is capable of conducting open-ended conversations. This agent does not require any intervention of a human to converse with people; it responds to user input fully automatically. By using speech recognition and paralinguistic parameter recognition, we can apply conversational style matching in real-time.

We designed an experiment where participants spoke to the agent for about 15 minutes. Participants would either interact with an agent that applied conversational style matching or an agent that did not. Results from this study enabled us to formulate design guidelines for conversational agents, based on our findings.

### **6.3 Intelligent Agent Design**

We built an intelligent conversational agent capable of conducting an open-ended conversation with a person. The agent was built using the open-source Platform for Situated Interaction (PSI) [27]. Figure 6.1 shows the architecture of the agent, and below, we describe the constituent components.

#### *6.3.1 Sensing*

##### *Speech Recognition*

Voice activity was detected with the Windows system voice activity detector (Microsoft, Inc.). Voiced audio was then passed to the Bing speech recognition engine<sup>1</sup>, which returned the recognized textual string.

##### *Paralinguistic Parameter Recognition*

For the voiced audio segments, the paralinguistic speech features were also extracted. Specifically, we detected the fundamental frequency or pitch ( $f_0$ ), and the root mean squared (RMS) energy reflecting the loudness of the voice. These features were extracted using a digital signal processing approach similar to that described by Han et al. [116].

---

<sup>1</sup><https://azure.microsoft.com/en-us/services/cognitive-services/speech/>

### 6.3.2 Synthesis

#### *Dialogue Generation*

We used a data-driven paradigm of conversation generation [283] in which neural models are built, typically from large-scale unconstrained datasets. Our model was trained on social media data that were mined from the Twitter fire hose. The neural models can then return a ranked list of the most likely responses for any text-based input. Despite the diverse and relatively unconstrained dialogue examples, the model is able to learn to conduct simple conversations. However, while these unconstrained models can conduct general chit-chat effectively, they are not goal-driven, nor do they have a deep knowledge-base about the world. Therefore, we added the capability to identify specific intents in the text to our system by using the Language Understanding and Intent Service (LUIS)<sup>2</sup>. LUIS can recognize intents by supplying training examples of them, after which it is able to detect variations of the intents. An example of intent might be “Movies” and a training example might be “Do you like movies?” or “What are your favorite films?”. LUIS was trained to recognize 10 to 15 specific intents (165 intents total across all tasks). When the system recognizes an intent, the system responds with a predefined scripted response using its dialogue manager. These responses are less open-ended and often pose a specific question to the user, allowing us to extend the interaction time participants had with the agent.

Figure 6.2 shows the excerpts from a conversation with the agent collected during our user study. The figure displays both the dialogue generated by the neural model and the scripted responses to detected intents are emphasized. A majority of the utterances in each conversation were generated by the neural dialogue model.

#### *Speech Synthesis*

We use the speech synthesis markup language (SSML) [35] to vary the prosodic qualities of the generated speech (pitch, loudness, and speech rate). Within our system, SSML allows each of these properties to be specified at the utterance level. For both pitch and loudness, the system

---

<sup>2</sup><https://www.luis.ai/home>

### Discussing personal life:

**User:** Hey, I'd like to get to know you better can I ask you some questions?

**Agent:** I'd love to tell you about myself. What do you want to know?

**User:** Can you tell me a little bit about where you are from?

**Agent:** I am from the UK, I live in London.

**User:** When did you move here?

**Agent:** A few years ago..

**User:** How is life here compared to the UK?

**Agent:** Life is good.

### Planning a trip to London:

**User:** Do you know if there are any museums around Buckingham Palace in London?

**Agent:** I am sure there will be some in the area.

**User:** What about parks?

**Agent:** *I know Buckingham Palace is surrounded by some beautiful parks and you can check out the palace or you could go to Hyde Park where you can row a boat. What do you think?*

**User:** I think that sounds nice... I think I'd also like to go for afternoon cream tea...

### Planning a trip to a movie:

**User:** Do you want to watch a movie tonight?

**Agent:** I don't know, what are you watching?

**User:** Do you have a favorite actor?

**Agent:** *I like Meryl Streep, do you have a favorite actor?*

**User:** Erm, that's a good question. ... I am not really sure.

**Agent:** Yes, it is a great question. Neither am I.

**User:** I like Ryan Reynolds. Do you know who Ryan Reynolds is?

**Agent:** I don't think I have heard of him.

Figure 6.2: Transcript of interactions between users and the agent for 3 different tasks. Emphasized responses were scripted responses based on detected intents of the user.

allows variation between five different levels (ranging from very low pitch/loudness to very high pitch/loudness). The speech rate is changed with a floating-point number, where 1 represents standard speed, 2 is double speed, 0.5 is half speed, and so forth.

#### **6.4 Conversational Style Matching**

In order to match the participant's conversational style, the agent leverages its sensing components (speech recognition and the paralinguistic parameter recognition). Then, based on the data collected through these components, the agent calculates several variables that define the user's conversational style. These variables are based on the work of Thomas et al. and have been shown to capture the consideration-involvement dimension of conversational style [302]. The specific features used in this study were selected to be computable at scale in real-time, without intervention. By manipulating the variables in its own responses, the agent performs conversational style matching. The agent matches both the content of its utterances, as well as the prosodic qualities of the spoken response.

##### *6.4.1 Content Variables*

We used four different variables to define a participant's word usage: one variable tracks pronoun usage, two variables measure repetition, and one measures utterance length.

###### *Pronoun use*

The first variable is personal pronoun usage, which measures the rate participants use personal pronouns in their speech (e.g., you, he, she, etc.). This measure is calculated by calculating the usage ratio of these words compared to other words occurring in each utterance.

###### *Repetition*

In order to measure repetition, we used two variables that both relate to the repetition of terms (A term in this context is a word that is not considered a stop word). Repetition can be seen as a

measure of persistence in introducing a specific topic. The first of the variables measure the rate of repeated terms per individual utterance. The second is a ratio of sentences that contained any repeated terms.

#### *Utterance length*

The last content variable is a simple measure of the average number of words per utterance and represents the participant's average utterance length.

#### *6.4.2 Acoustic Variables*

We used three different variables that describe the participant's paralinguistic features in their speech; speech rate, pitch variance, and loudness variance.

#### *Speech rate*

The speech rate variable measures how quickly participants speak. As such, it is somewhat related to the utterance length to define verbosity. The speech rate measures how many words the participant says per second. This is calculated by using the Windows system voice activity detector to determine whether the participant is speaking and then comparing the recognized text with the duration of the voice activity, giving us the average number of words per second for each utterance.

#### *Pitch*

The pitch feature was measured by determining the participant's  $f_0$  on a per utterance basis and was stored as such.

#### *Loudness*

The final feature, loudness, was measured in a similar way as to how the pitch was measure—by determining the detected RMS energy of each utterance.

The above variables were all calculated in real-time while the participant was interacting with the agent.

### *6.4.3 Conversational Style Matching Synthesis*

In order to perform conversational style matching, the agent matches the participant on the previously defined conversational style variables in two different ways. For the word choice and utterance length, the agent adjusts the content of the utterance in order to match the participant more closely. For the prosody variables and speech rate, the agent changes the prosody and pace of its own responses. In all cases, the conversational style variables were aggregated over the last five utterances to reduce the likelihood of dramatic changes in style from one utterance to the next.

#### *Utterance content matching*

In order to match the conversational style to the utterance content, the agent adjusts the ranking order of the responses generated by our dialogue model. This is done by calculating the conversational style variables (word choice and utterance length) of the top 10 responses returned by the dialogue model's beam search algorithm. The answers are then re-ranked based on how closely they match the content variables of the interlocutor. Since the top 10 responses of the dialogue model are generally very close in meaning to each other, we can adapt the conversational style of the response to that of the user, without changing the meaning of the response.

#### *Acoustic matching*

The agent matches the participant's prosody and pace using SSML to adjust the pitch, loudness, and speech rate of its utterances. The matching of prosody and pace is done on a local level, meaning the agent matches the participant's changes, rather than match their overall prosody and pace. In order to do this, the agent tracks the average pitch and loudness of the participant and uses this as a baseline of the participant's speech. It then determines the current style by comparing the values of the last five utterances to this baseline. After establishing a baseline, the agent matches

	Control	Matching
Gender	M = 8, F = 7	M = 8, F = 7
Conv. Style	HC = 5, HI = 10	HC = 9, HI = 6
Age	$\mu=33$ ( $\sigma=10.9$ )	$\mu=31$ ( $\sigma=6.1$ )

Table 6.1: Summary of Participation

any detected increase or decrease in these values. Because of this approach, the agent requires a stable baseline of several utterances before it can perform matching. As such, it does not perform prosody matching on the first four utterances.

## 6.5 User Study

Our user study protocol was approved by the Institutional Review Board (IRB) of Microsoft. The study used between subjects’ design. Participants either interacted with an agent using conversational style matching, or one that did not. To reach 15 minutes of interaction time with the agent, we defined a set of tasks (scenarios) that the participant would discuss with the agent (we define the specific tasks further in the Tasks subsection 8.4.1). After the interaction with the agent, participants filled out several surveys used to investigate our hypotheses (the specific surveys are defined in the Materials subsection 6.5.3). Participants would take approximately 15 minutes to fill out the surveys, resulting in a total time of 30 minutes to complete the study.

### 6.5.1 Participants

We recruited 30 participants (14 females) total. The average age was 32 years ( $SD = 8.77$ ). Although 16 participants were identified as HI speakers and 14 were HC, we were unfortunately not able to balance the study for conversational style as this was a self-reported measure collected during the study. The participants reported varying levels of familiarity with speech-based interfaces. Twenty-two participants owned a smart speaker. Table 6.1 contains a summary of the participants.



Figure 6.3: Setup for the user study

### 6.5.2 Apparatus

During the user study, the participants were seated singly in a room. A speaker (FUGOO Style-S Portable Bluetooth Speaker) was placed on a table opposite them. A microphone was placed on a table next to the participant. The configuration of the apparatus can be seen in Figure 6.3.

### 6.5.3 Materials

We used several self-report survey instruments to capture the experience of the participants. We first captured several participant traits:

#### *Conversational Style*

We recorded the participants' self-reported conversational style using a survey created by Shamekhi et al. [277], based on Tannen's [295] style theory.

### *Personality*

We captured the personality of participants using the Big Five Personality Inventory survey by John et al. [148].

### *System Rating*

The second set of surveys were used to capture the participants' attitude towards the agent and ratings of their interaction with it.

### *Interaction Rating*

To measure how participants experienced the interaction with the conversational agent, we used the same questionnaire as Shamekhi et al. [277]. The interaction questionnaire contains several questions relating to the participant's overall impression of the interaction, as well as some questions on the perceived trustworthiness of the agent. We, therefore, created two composite variables, one relating to interaction and one on the trust rating of the agent.

### *Agent Impressions*

To capture the participants' impressions of the agent, we used the Godspeed questionnaire by Bartneck et al. [15]. This survey was originally used for obtaining ratings on robots, therefore some minor modifications made (we discarded ratings on physical movement).

#### *6.5.4 Tasks*

We defined several tasks for participants to perform during the study. Each task described a scenario that the participant discussed with the agent. For each task, 10 to 15 scripted responses were created, and these responses were triggered when the LUIS intent recognition engine detected a specific intent relating to one of the scripted responses. (see Figure 6.2 for an example of a transcript).

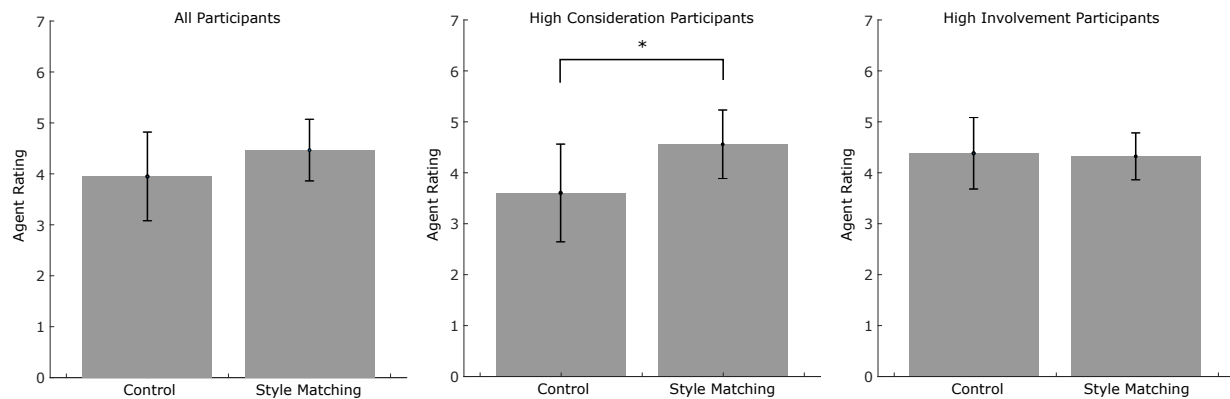


Figure 6.4: Left: Rating of trust in the agent for all subjects ( $N_{control} = 15$ ,  $N_{matching} = 15$ ). Middle: Rating of trust in agent for high consideration subjects ( $N_c = 5$ ,  $N_m = 9$ ). Right: Rating of trust in agent for HC subjects ( $N_c = 9$ ,  $N_m = 6$ ).

### *Planning Lunch*

During this task, the participants were asked to schedule a lunch meeting with the agent. The topics included: where to eat, where to meet, and what to have for lunch.

### *Personal Life*

During this task, the participants were asked to talk with the agent about their personal life. The topics included: where they were from, their likes and dislikes, and their jobs.

### *Discussing a Vacation to London*

During this task, the participant was asked to plan a trip to London, UK. The topics included: places to visit (museums, art galleries, parks, etc.), how to travel around the city and where to stay.

### *Organizing a Party*

During this task, the participants were asked to organize a party with the agent. The topics included: where to host the party, what food to serve, and what music and decorations to have.

### *Planning a Trip to a Movie*

During this task, the participants were asked to plan a trip to see a movie. The topics included: movie preferences (favorite actors, genres, and directors), which movie theater to go to and when to go.

## **6.6 Results**

Overall, the participants spend at least three minutes in each of the five tasks and spoke with the agent for an average of 18 ( $\sigma = 1.94$ ) minutes. This resulted in close to nine hours of interaction with the agent across the 30 participants.

We first analyzed the results of the participants' ratings of the overall interaction (H1a) and the perceived trustworthiness (H1b) of the agent, both composite measures obtained through the interaction survey. We did not find a significant difference between the style matching agent and the control condition for either overall interaction ( $t(28) = .21, p = .837$ ), nor for the trustworthiness rating ( $t(28) = 1.26, p = .22$ ).

Following this, we performed a set of two-way ANOVAs on the effects of the experimental condition (conversational style match or control, H2) and the participants' conversational style (HC or HI), for both the interaction rating and the trustworthiness rating of the agent. For the first ANOVA on the overall interaction rating we did not find a main effect across conditions ( $F(1, 26) = .004, p = .952$ ), nor did we for conversational style ( $F(1, 26) = .239, p = .629$ ). There was not a significant interaction between condition and the participants' conversational style either ( $F(1, 26) = .001, p = .98$ ). As such, we cannot confirm hypothesis 1a.

The second two-way ANOVA on the agent's trustworthiness ratings did not show a significant main effect for the condition ( $F(1, 26) = 2.603, p = .119$ ), nor for the participants' conversational style ( $F(1,26) = .826, p = .372$ ) and as such we cannot confirm hypothesis 1b. However, although the interaction between the condition and the participants' conversational style was not significant, there appeared to be a trend towards significance ( $F(1, 26) = 3.347, p = .079$ ).

We further investigated this interaction trend with a simple effects analysis, which showed that

it appears to be driven by participants with an HC conversational style ( $p = .027$ ). HC participants rated the conversational style matching agent more trustworthy ( $\mu = 4.56$ ,  $\sigma = .24$ ) than the control condition ( $\mu = 3.6$ ,  $\sigma = .33$ ). However, for HI participants there was no difference ( $p = .875$ ), as they rated the style matching agent ( $\mu = 4.3$ ,  $\sigma = .3$ ) similar to the control condition ( $\mu = 4.36$ ,  $\sigma = .23$ ). Thus, showing that the agent's conversational style matching appears to be more important for HC participants. This result provides some support for hypothesis 2. Figure 6.4 shows the mean agent ratings for the trustworthiness scores for the two experimental conditions. On the left, we show the results across all participants, in the middle the results for the participants with high consideration conversational styles and on the right the results for the participants with high involvement conversational style. Figure 6.5 shows an interaction plot between the conversational style and the experimental condition.

Lastly, since we also measured personality with the big five questionnaires, we investigated whether there is a correlation between personality and trust. However, we found no significant correlations, with the highest correlation for the personality types being between extraversion and trust ( $r=.27$ ,  $p=.14$ ).

## 6.7 Discussion

Using our conversational agent and dialogue model, the participants were able to sustain conversations for the expected 15 minutes. Scripting a natural dialogue of this length was novel and challenging. However, the neural language model was able to generate dialogue successfully turns that were appropriate for general chit-chat, while the scripted responses were successful in responding to on-topic utterances and drive the conversation.

We found that participants with an HC conversational style rated the agent as more trustworthy when it matched their conversational style, compared to when it did not, whereas, participants with an HI conversational style rated both agents similarly. We did not find any differences in the rating of interaction quality. However, the small number of participants in our study could limit the significance of our results. We ran 30 participants total spread across 2 conditions. As a comparison, Niederhoffer and Pennebaker ran a study on linguistic style matching between dyads with a total

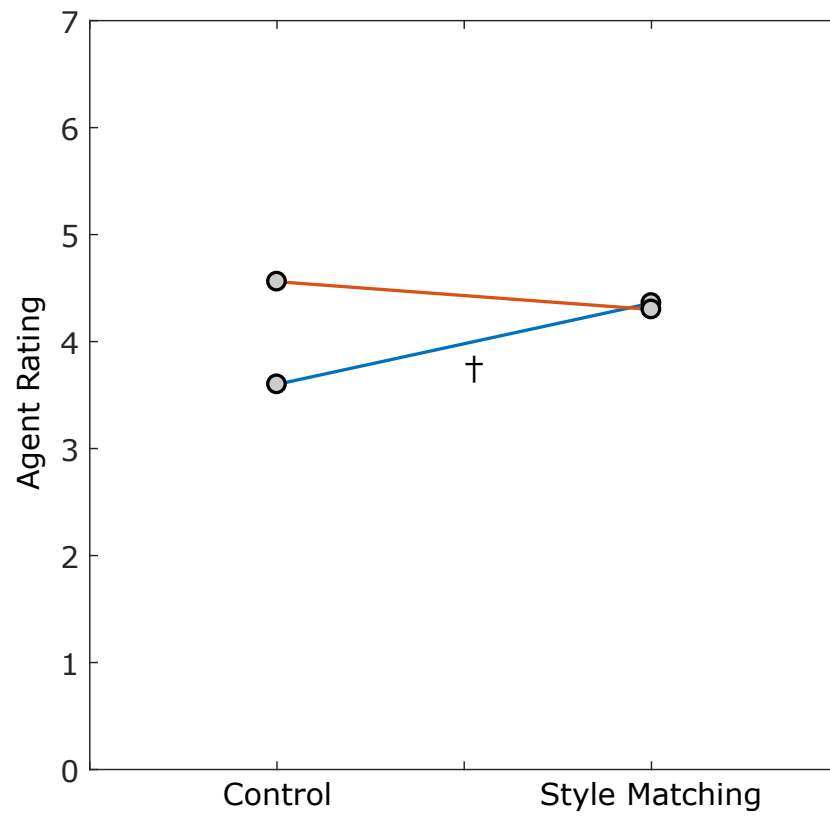


Figure 6.5: A mildly significant interaction effect ( $p=0.079$ ) was observed between conversational style and the experimental condition. However, this was driven almost entirely by the high consideration participants (see Figure 6.4)

of 62 dyads [223]. It is expected that the effect sizes of subtle adaptations in conversational style might be small after a 15-minute interaction. Nevertheless, it could remain an essential design consideration when a system is deployed longitudinally “in-the-wild”.

One possible explanation for the result of HC participants rating a conversational style matching agent as more trustworthy might lie in how these participants are more considerate of their partner while having a conversation. As such, the agent adapting to the participant’s conversational style might be appreciated more by an HC speaker than by an HI speaker, thus resulting in a higher trustworthiness rating. To put it simply, HI speakers might be less troubled by an interlocutor that does not match their style.

We observed some differences in how participants interacted with the agent. Some of these differences might have affected how participants experienced the agent (regardless of condition). For example, for 11 out of the 30 participants, some speech overlap occurred with the agent (see figure 6.2, for example). Overlap can be seen as an aspect of an HI conversational style, but in our study, this was not part of the design and occasionally occurred in both conditions. There are several reasons why speech overlap occurred. For example, it might be because of a participant’s conversational style (i.e., HI participants are more likely to overlap in speech). However, it also happened because of limitations in the design of the system, as the agent responded to every single utterance it recorded. At times, the agent made segmentation errors by splitting up a single utterance into multiple, and as such, would respond to several utterances in a row. If a participant was not aware of this, they might consider that the agent was finished speaking after the first utterance, and then overlap with the agent’s response to the second part of the utterance. Secondly, if these participants did not finish their sentences, the agent would still respond to what it heard. This behavior was not expected by some participants and could also occasionally lead to nonsensical responses of the agent. In general, participants that took longer pauses while interacting with the agent did not experience overlap as often, whereas those that spoke faster would often overlap with the agent (possibly thinking that the agent was finished speaking when it had not). Some of these types of overlap observed could, unfortunately, go on for several rounds of the conversation.

## 6.8 *Design Guidelines*

We define some design recommendations to keep in mind when developing a conversational style matching agent using unconstrained dialogue models. The guidelines are based on our observations of participants interacting with the conversational agent.

For conversational style matching to be most effective, some care needs to be taken. One major pitfall is that the positive effect of the matching condition could easily be overshadowed and undone by the negative effects of other limitations of the system. In some cases, the matching could even exacerbate these negative effects. For example, if an agent gives answers that frustrate its user, this could influence the user's conversational style. This frustration is then matched by the agent, which will most likely affect the experience of the participant. This could create a negative spiral of negative style matching. As such, it is crucial to consider this while designing a system that uses style matching.

Keeping the above in mind, if the system does have some significant flaws, it would be wise to notify the participant of this and not oversell the system's capabilities. For example, if it takes a while for a system to process and answer specific questions and style match, the user needs to be made aware of this. If the participant is informed of this before the interaction, it could create some understanding and perhaps reduce frustration from having to wait for an answer. We have observed this in subsequent studies, which we will report on in the future.

As previously noted, there can be a lot of overlap between the temporal aspects of the speech of the agent and the participant. Although overlap in speech can be common in human-human dialogue, this is usually an aspect of an HI conversational style. Therefore, to build an HC-appropriate conversational agent, this needs to be considered. Several relatively simple changes could deal with these types of issues. One solution is allowing the agent to be interrupted (i.e., the agent stops talking when overlapping speech is detected), thus giving the floor to the participant to continue speaking. Another way of avoiding overlap is to filter out stop words and interjections from the participant. Currently, the system does not do this and therefore responds to any utterance of the participant, which can potentially lead to some confusion on the participant's side, since people

often do not even realize they are using stop words or are interjecting, as we observed.

One other useful addition could be the incorporation of more frequent usage of pauses by the agent. This was not implemented in the current system, but there is a great difference in usage of pauses between HI and HC speakers, and this might have a significant impact on the participant's perception of the system.

## **6.9 Summary**

We designed an end-to-end voice-based conversational agent capable of multi-turn dialogue to study the effect of conversational style matching on participants' perception of the agent. Thirty participants interacted with the agent for approximately 15 minutes. We found that individuals with High Consideration conversational styles were more likely to trust an agent that matched their conversational style, whereas those with the High Involvement style were indifferent. Despite the subtle nature of conversation style matching, we were able to observe an effect on participants' trust ratings after only 15 minutes of agent interaction. This is very encouraging for future agent designs that aim to leverage subtle conversational matching (or entrainment).

## Chapter 7

# HIGH-FIDELITY OPEN EMBODIED AVATAR

### 7.1 Introduction

Virtual agents have many applications, from assistance to companionship. With advancements in machine intelligence, computer graphics, and human-computer interaction, agents are becoming more and more capable of real-time interaction with the user, synthesizing speech and expressions, style matching in conversations, tracking, and completing tasks as assistants. How we represent an intelligent system via a user interface has implications for how the user perceives the system and associates or attributes intelligence with it. One line of research emphasizes that embodied agents offer several advantages over non-embodied dialogue systems. Agent with a physical presence allows the users to look at them. Cassell [41] argues that embodiment can help locate intelligence for users and reconcile both interactional and propositional intelligence. Embodied agents have greater expressive capabilities, and naturally embodied agents have been used to explore the role of social signals [43] and emotions [245] in human-computer interaction.

Improvements in speech recognition [323], dialogue generation [274, 132], emotional speech synthesis [281, 197] and computer graphics have made it possible to design more expressive and realistic conversational agents. However, there are still many uncertainties in how best to design embodied agents. Building a state-of-the-art system for performing research to address these questions is non-trivial.

Open code, data, and models democratize research, helping advance the state-of-the-art by enabling researchers to build upon existing work. LUCIA [135] is an MPEG-4 facial animation system that can copy a real human being by reproducing the movements of passive markers positioned on his face. Several architectures have been proposed for designing and building embodied conversational agents (ECA). AVATAR [264] is an architecture to develop ECAs based on open

source tools and libraries. FACSvatar [309] processes and animates action units in real-time. The framework is designed to be modular, making it easy to integrate into other systems and has low latency allowing real-time animations. FACSHuman [108] is a similar tool that allows animation of a 3-dimensional avatar using the facial action coding system (FACS)[81]. The system, however, does not allow for easily constructing real-time multimodal ECAs.

While these systems have been very valuable, there are some limitations. First, driving the non-verbal behavior of the agent is often not simple. For example, LUCIA [135] requires the user to wear reflective markers. In [208], people rated the realistic characters or very cartoon-like characters as more appealing or pleasant, but not characters that lie in the middle of the ‘abstract to realistic’ scale. GRETA is an autonomous and expressive agent that is somewhat realistic in appearance and has numerous capabilities for social and emotional expression in addition to dialogue [249]. We have designed a freely available high-fidelity embodied avatar that has lip syncing, head gesture, and facial expression capabilities. It can be controlled via a simple python interface, using either bone positions, action units, or basic expressions. The goal is to make it easier for people to develop ECAs.

The contributions of this work are:

- We present a high-fidelity open embodied avatar that is capable of lip syncing and facial expression control via a simple python interface, using either bone positions, FACS or basic expressions.
- We provide an example of an end-to-end conversational agent interface via publicly available APIs.
- We show how expression transfer can be used to control the avatar.

The avatar, code, and models presented in this chapter were previously published in [8], and the software information can be found at <https://github.com/deepalianeja/AvatarSim>.

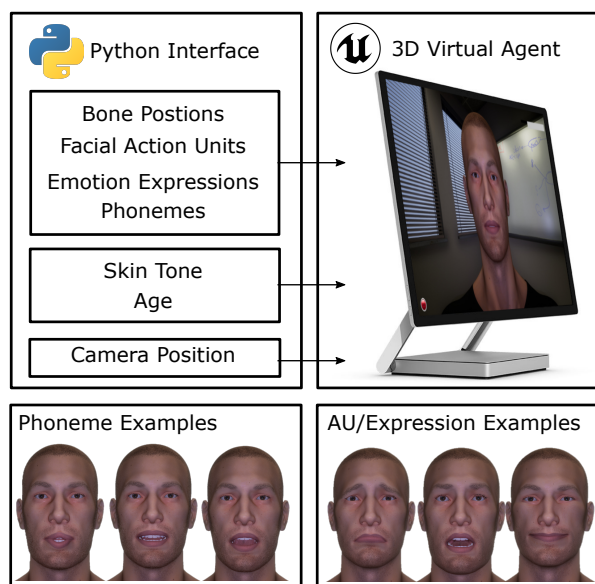


Figure 7.1: We present an open embodied avatar with lip syncing and expression capabilities that can be controlled via simple python interface. We provide examples of how to combine this with publicly available speech and dialogue APIs to construct a conversational embodied agent.

## 7.2 The Avatar

We created the avatar within the AirSim [276] environment using Unreal Engine 4 (UE4) [106]. The avatar, a head, and a torso is placed in a room with multiple light sources (windows, overhead lights, and spotlights). Pictures of the avatar are shown in Figure 7.1. The avatar’s appearance can be manipulated in several ways. The position and orientation of 38 “bone” positions, which are facial locations (or landmarks), can be controlled, each with six degrees of freedom. We also created 24 FACS action unit presets [81] and 19 phonetic mouth shape presets (visemes) for the avatar. Each of these variables can be controlled in real-time using a simple Python interface. Once the avatar executable is running and a client connection formed between the python instance, the positions of bones, the action unit, and phoneme presets or the location of the camera can be updated as frequently as needed. The avatar controls are described as follows:

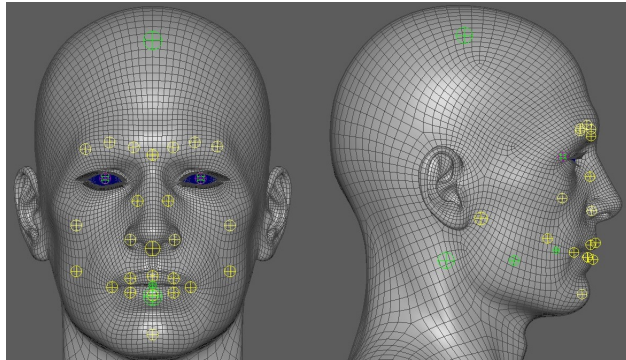


Figure 7.2: The locations of controllable bone positions on the avatar. Each position can be controlled with 6 degrees of freedom. See Table 7.1 for descriptions.

**Bone Position Controls.** The position  $(x_b, y_b, z_b)$  and orientation  $(\sigma_{b1}, \sigma_{b2}, \sigma_{b3})$  of each of the 38 bone positions can be set. Figure 7.2 shows the location of the bone positions and Table 7.1 lists their names. The green points indicate joints that control rigid head and jaw motions. The yellow points indicate landmarks that control specific facial regions (e.g., eyebrows). The bone positions can be referred to via these names (e.g., LUpperEyelid for the left upper eyelid) or by their integer as specified in our documentation.

**Facial Action Controls.** The Facial Action Coding System [81] is the most widely used and objective taxonomy for describing facial behavior. Therefore, in addition to control of bone positions, the avatar has controls for 24 facial action unit presets (see Table 7.1). The FACS action units can be referred to via their names (e.g., FACS\_04 for Brow Lowerer), and the intensity of the facial action units can be controlled continuously via a float value from 0.0f to 1.0f.

**Expression Controls.** A set of basic emotional expressions (anger, contempt, disgust, fear, joy, sadness, and surprise) can be created via combinations of action units (e.g., using EMFACS [103]) or using expression retargeting as described in Section 3. We provide code to enable these. The avatar is also capable of blinking the eyes by controlling the upper and lower eyelid bone position controls.

Bone Positions	Action Units	Phonemes
U. Cheek (L & R)	1: I. Brow Raiser	ae
Mid U. Lip (M, L & R)	2: O. Brow Raiser	b
Mid L. Lip (M, L & R)	4: Brow Lowerer	d
Lip Corner (L & R)	5: U. Lid Raiser	ee
U. Nose (M, L & R)	6: Cheer Raiser	ngnk
O. Brow (L & R)	7: Lid Tightener	eh
Mid Brow (L & R)	9: Nose Wrinkler	ar
I. Brow (L & R)	10: U. Lip Raiser	f
Nostril (L & R)	11: Nasolabial Deep.	g
Cheek Dimple (L & R)	12: Lip C. Puller	h
Eye (L & R)	13: Sharp Lip Puller	i
U. Eyelid (L & R)	14: Dimpler	k
L. Eyelid (L & R)	15: Lip C. Depressor	l
Jaw	16: L. Lip Depressor	m
Chin	17: Chin Raiser	n
Tongue Base	18: Lip Pucker	oh
Tongue Tip	20: Lip Stretcher	ooo
Chest	22: Lip Funneler	oh
Neck	23: Lip Tightener	m
Head	24: Lip Pressor	
Root	25: Lips Part	
Hair	26: Jaw Drop	
	27: Mouth Stretch	
	28: Lip Suck	

Table 7.1: The avatar has the bone position, facial action unit, and phoneme controls. M=middle, L=left, R=right, U=upper, L=lower, C=corner, I=inner, O=outer

**Lip Syncing Controls.** For conversational applications, lip syncing is enabled via phoneme control. In our example scripts, we show how a phoneme classifier [133] can be first used to identify phonemes from an audio segment and then used to control the lips in synchrony with the audio playback. The phoneme presets can be referred to via their names (e.g., `Phoneme_b` for “b” phoneme), and the intensity of the phoneme presets can be controlled continuously via a float value from 0.0f to 1.0f. Nineteen phoneme presets are included, while there are 44 phonemes in English the 19 presets can be used quite effectively for most dialogue.

**Head Gesture Controls.** The head gesture can be varied continuously for head rotations by controlling the yaw, pitch, and roll between  $-1.0f$  and  $1.0f$ . The head orientation needs to be specified as a quaternion in the NED frame.

**Appearance Controls.** The skin tone and age of the avatar can be controlled via independent parameters. The skin tone varied from light (0) to dark (1), and the skin age texture can be varied from youthful (0) to old (1).

**Camera Controls.** The camera position within the virtual environment that governs the perspective from which the agent will be captured can be controlled with six degrees of freedom (position  $(x_c, y_c, z_c)$  and orientation  $(\sigma_{c1}, \sigma_{c2}, \sigma_{c3})$ ).

## 7.3 Applications

### 7.3.1 Conversational Agent

To illustrate how the avatar can be used as a conversational agent, we provide an end-to-end pipeline for conversational dialogue, similar to [125]. The pipeline uses public APIs for natural language generation and speech synthesis from Microsoft. We created a demonstration script that calls a natural language API to generate utterances and then uses a text-to-speech service to return a synthesized voice. We convert this voice into a sequence of visual groups of phonemes (visemes) using PocketSphinx [133]. Finally, we synchronously play the audio and drive the avatar using the phoneme presets to achieve a simple lip syncing example of a conversational agent. Python scripts for running this conversational agent are provided with the agent software.

### 7.3.2 Facial Expression Retargeting

In virtual embodied agent interactions, whether for conversation or not, expressions and head motions are important for creating a more natural and lifelike agent. Our next example shows how

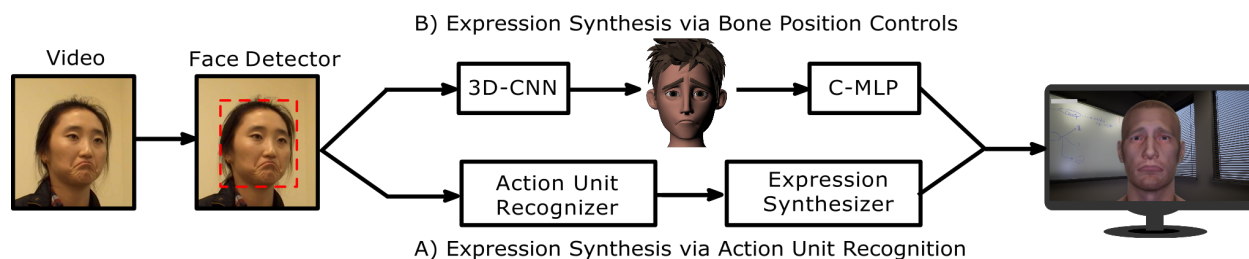


Figure 7.3: We present two pipelines for retargeting an expression from a human to the avatar. A) Expression Synthesis via Action Units Controls. Our pipeline takes a human video as an input, recognizes 12 FACS Action Units from the user’s detected face, and synthesizes the same expression on the avatar’s face. B) Expression Synthesis via Bone Position Controls. Our pipeline takes a human video as an input, detects the user’s face, generates 3D parameters on the primary character ‘Ray’ through a 3D-CNN, and synthesizes the same expression on the avatar’s face. See Section 3.2 for details for the Action Unit Recognizer, Expression Synthesizer, 3D-CNN and Character Multi-layer Perceptron (C-MLP).

expressive faces can be generated. We provide two interfaces for driving the expressions of the avatar (see Figure 7.3), one using bone controls and the other using the FACS presets. In both cases, we utilize existing deep learning-based frameworks to perform the expression transfer, i.e., synthesizing the avatar’s face with the expressions of a human subject in the input video. This creates the effect of facial expression mimicry. We provide scripts and a FACS-based trained model for systematically generating faces. The facial expressions on the avatar can be synthesized as follows:

**Expression Synthesis via Facial Action Units.** We describe an end-to-end pipeline to retarget a human’s facial expression to the avatar’s face using facial action unit controls, as shown in Figure 7.3 (A). The pipeline utilizes a webcam attached to the computer and processes the video frames. First, we detect the face in the input video. We use an open source HoG-based face detector [162]. Next, we analyze the facial region of interest (ROI), defined by the face detection bounding box, using the Facial Action Unit Recognizer. The output from the Action Unit Recognizer is used to synthesize the expressions on the avatar’s face by the Expression Synthesizer.

### *Facial Action Unit Recognizer*

We trained a Convolutional Neural Network (CNN) [119] on a large facial expression dataset [92] for recognizing 12 facial action units - 1, 2, 4, 5, 6, 9, 12, 17, 20, 25, 26 (refer to Table 7.1 for descriptions) and 43 (Eyes Closed). The Action Unit Recognizer returns a 12-dimensional feature vector representing the probability of each action unit. The model was trained using PyTorch [240] to minimize the binary cross-entropy loss with an average F1-score of 0.78. More details about the training and evaluation of the model can be found in [182]. Alternatively, the avatar could be used with an off-the-shelf FACS AU detection SDK, such as [209].

### *Expression Synthesizer*

We use the facial action unit probability scores from the Facial Action Unit Recognizer to control the avatar’s facial action controls and mimic the human facial expressions. The raw probability scores can be used directly to drive the actions, or these values can be smoothed or rounded first. Python scripts for running the expression synthesis via facial action units are provided with the agent software along with the trained Action Unit Recognizer model.

**Expression Synthesis via Bone Position Controls.** We describe an end-to-end pipeline to retarget a human’s facial expression to the avatar’s face using the bone position controls, as shown in Figure 7.3 (B). As with the previous approach, face detection is first performed on the input video and the bounding box used to define the ROI. We then use a multi-stage deep learning system ExprGen (see Chapter Chapter 3), to retarget facial expressions from the human to the avatar. The pipeline takes 2D images of human facial expressions as input and generates 3D parameters of the avatar. We have two separate components in our pipeline: Human to Primary Character expression transfer, including both perceptual and geometric similarity, and Primary Character to Avatar expression transfer, which includes a lightweight method expression retargeting from the primary character to the avatar. The expression transfer result from human to the primary character ‘Ray’ is shown in Figure 7.3 (B) (middle). Then, we train a C-MLP to transfer the expression

of the primary character to the avatar (secondary character). More details about the dataset and training the model are described in Chapter Chapter 2 and Chapter 3.

#### **7.4 Limitations**

Some of the limitations of our work are: The intensity of avatar’s FACS action units is not validated, the synthesis of composite facial expressions using multiple facial action units along with phoneme based lip syncing is currently rule-based and not validated for expression clarity, and the performance between the expression transfer via action units vs. bone position controls is not evaluated. When using the avatar as a conversational agent along with expression retargeting, it can result in implausible facial movement due to the conflict between lip syncing controls and expression controls. To avoid such a scenario, we control the expressiveness of the avatar’s upper face only (above the mouth) while lip syncing. However, the avatar allows for the exploration of more advanced solutions to this problem. In future research, we can use the Facial Action Coding System (FACS) and/or perceptual studies to evaluate the intensity of the retargeted avatar facial expression.

#### **7.5 Access and Citation**

The *avatar* is available for distribution to researchers online. The avatar, code, models and software information can be found at <https://github.com/danmcduff/AvatarSim>. The software is released under the MIT license and a responsible AI license<sup>1</sup>. This allows the use of the software without restriction, including without limiting rights to use, copy or modify it, with the exception of a set of explicit applications. If you find this software helpful or use it in research, please cite this paper. Citation information can be found on the GITHUB page.

---

<sup>1</sup><https://www.licenses.ai/>

## **7.6 Summary**

We present an open high-fidelity embodied avatar with bone, action unit, and phoneme controls. We release code and models to illustrate how the avatar can be controlled like a puppet, via expression retargeting or used to create a simple conversational agent using public APIs. We hope that this resource enables researchers to advance the state of the art in high-fidelity embodied virtual agents. Contributions to the code repository are welcomed.

## Chapter 8

# SOCIALLY INTELLIGENT VIRTUAL AGENT

### **8.1 Introduction**

#### *8.1.1 Motivation*

Conversational interfaces are becoming increasingly popular. Recent advances in speech recognition, generative dialogue models and speech synthesis have enabled practical applications of voice-based inputs. However, these systems are constrained in terms of how they can communicate, because they are limited to vocal interactions, and even those do not reflect the vocal characteristics of the human’s speech. Central to this argument is that it is important to realize how humans engage with each other during normal conversations. Research has also found that embodied agents offer several advantages over non-embodied dialogue systems. One line of reasoning is that an agent that has a physical presence provides visual affordances, so that users know where to focus their attention and can receive cues from body and hand gestures, eye gaze and other expressions [43]. The human body allows us to “locate” intelligence—not only the typical domain knowledge required, but also the social and interactional information we need about conversational parameters, such as turn-taking, taking the floor and interruptions.

In this vision, then, an embodied social agent who converses with the user requires less navigation and searching than traditional user interfaces (because the user knows where to find the desired information). Multimodal gestures, such as deixis, eye gaze, speech patterns, head nods, and other, nonverbal gestures are external manifestations of social intelligence that support trustworthiness [22]. For instance, early research has shown that to attain conversational clarity, people rely more on gestural cues when their conversations are in noisy environments or are otherwise difficult to understand [259]. From this perspective, embodied social agents might be a more natural way for people to interact with computation [43].

In this view, conversational agents provide a familiar mental model for the user. Well-designed or possibly, anthropomorphic features can then help to create a framework for understanding how to work with these agents. Specifically, conversational agents can provide affordances for possible interaction qualities, capabilities, and limits. We argue that, if agent designers can tap into users' natural proclivity toward social interaction, this will also lead to higher levels of affinity for, and interaction with, that agent. If we design agents to not only behave as we expect them to, but also to adhere to social norms and values, then we should be able to amplify trust, which should encourage engagement [23, 270].

Creating convincing dialogue is one of the fundamental challenges for the design of a conversational agent. Dialogue grammars are based on valid sequences of utterances within a branching structure. Frame-based systems leverage entities and a dialogue manager to navigate a conversation. Plan-based approaches assume a goal for each task that is being addressed in the dialogue before deciding what to say next. A weakness of all these approaches is that the dialogue may stray out of the bounds of the domain, into chit-chat for example, in which case the systems are often not able to keep the conversation flowing naturally. The conversational style matching framework described in Section 6.3 is able to successfully generate dialogue turns that are appropriate for small talk and maintain longer conversations. However, the user study results showed that the voice-only matching does not make a significant impact on the ratings of the overall interaction and the perceived trustworthiness of the agent.

Our work builds on a rich background of research into intelligent embodied agents [22, 203, 172, 65, 101, 138, 248]. These systems include agents that are expressive in language, speech, facial and body gestures. However, in the past, agents have relied upon mostly scripted dialogue and/or prescribed "intents" that had to be pre-trained, limiting the opportunities for less constrained interactions. To address the challenges noted above, we present an embodied conversational agent (ECA) that is capable of multi-turn dialogue. Inspired by our previous work on auditory only conversational style matching [124], we designed our ECA to match appropriate facial expressive style along with the linguistic conversational style of the users to provide them with a more immersive experience while interacting with the ECA.

The contributions of this work are :

- We present an ECA that is capable of longer (over 5 minutes) conversations with a user.
- We study how conversational and expressive style matching influences a user’s perception of the emotional behavior and connection with the agent.
- We show how conversational and expressive style matching influences perception of the agent

### 8.1.2 Related Literature

#### *Embodied Conversational Agents*

It has been shown that rapport and information disclosure is higher with assistants that can engage in social dialogue and respond appropriately [22, 23]. REA, an embodied real-estate agent, was designed to be capable of social and task-oriented conversation with a user [22]. Since social goals are important for human-human interaction, they should also serve an important purpose in human-agent interaction. Due to the inclusion of social dialogue in the form of small talk in conversations with users, REA was perceived as more trustworthy (by extroverts) than an agent that did not engage in small talk [23]. SARA (socially-Aware Robot Assistant) [203] is a system capable of multimodal interpersonal understanding, reasoning and generation. In a field trial of SARA’s interactions with users, it was found that the interactional features (e.g., the frequency of interruptions and number of turns) played a role in building rapport with the user and improved task performance.

Perhaps the most common application for ECAs in research has been in healthcare, particularly in counseling and therapy, but also in diagnosis. SimSensei [65] is a virtual counselor which enables a face-to-face conversation and leverages multimodal signal processing through a system called MultiSense. WoeBot [101] is a completely automated, conversational, text-based “psychotherapist” that uses cognitive behavioral therapy to help patients reframe their perceived problems cognitively. ECAs have also been leveraged as social skills and interview trainers [292, 127].

MACH (My Automated Conversation Coach) is an example of such a system that is claimed to provide easy access to social skills training [127]. Harlie [138] is a conversational agent designed for using a user's speech to help diagnose Parkinson's disease. Harlie can be used on a smartphone and can carry out conversations with a user on a variety of topics. Tanaka et al. [291] used a conversational avatar that produced spoken queries to a user based on a quasi-mental state test, a revised Wechsler memory scale, and other neuropsychological questions. Finally, agents have also been used to help with diagnosing depressive disorders [248].

Conversational agents have their limitations as health consultants, however. In a study, [211] examining the responses of common smartphone agents, such as Siri, Google Now, and Cortana, it was found that with specific healthcare questions, ranging from simple, physical-based questions (like "My foot is hurt") to serious statements such as "I want to commit suicide", the agents were extremely inconsistent in recognizing the nature of the health statements, responding in a respectful and sensitive way, and directing the user to the appropriate public service for their health issue. The authors concluded that much more needs to be done to make these agents respond in a meaningful, more humanistic way to the various health statements that are likely to be addressed to these agents. We consider our approach to automatic generation of conversation across a broader swath of topics to move the needle more in the direction of agents recognizing in a more nuanced way a wider breadth of topics.

While all of these systems have proven useful in limited domains, none of them has been developed with an eye toward open-ended conversations across a wide variety of domains or for general productivity tasks. To our knowledge, we believe this work is the first to consider the combined role of conversational and expressive style of the interlocutor to drive interactions and perceptions of an agent and its behavior in order to create a more expressive and animate agent. We performed a laboratory study to explore this design idea with an eye toward providing guidance in developing more natural and likable conversational agents.

### 8.1.3 Style Matching

Style matching, or entrainment, is the phenomenon of a speaker adopting the behaviors or traits of their interlocutor. This can occur through word choice, as in lexical entrainment, that has been observed in human-human conversations [270, 269] and human-agent [277, 124] interactions. It can also occur in non-verbal behaviors such prosodic elements of speech [302], facial expressions and head gestures [43] and other embodied forms [272]. Again, non-verbal matching has been observed to affect human-human interactions [302], human-virtual agent interactions [179, 43] and human-robot interactions [227]. Matching can happen over a relatively short time scale [43]. Thomas et al. [302] observed style matching occurring between strangers performing information seeking tasks after only a couple of minutes [302]. Style matching has numerous benefits that help interpersonal interactions proceed more smoothly and efficiently [46].

To understand the human conversational style (CS), we follow Deborah Tannen's theory [294, 295] as described in Section 6.1.2. Tannen categorizes CS into high consideration (HC) and high involvement (HI). The HC style emphasizes consideration, independence and is characterized by slower and hesitant speech, longer pauses, and the use of moderate paralinguistics. The HI style emphasizes interpersonal involvement, interest, and understanding, and is characterized by faster and louder speech with shorter pauses between conversational turns. We wanted to test how matching of lexical and non-verbal behaviors would impact the perception of a fully autonomous ECA by people with different conversational styles.

## 8.2 SIVA - The Agent

We used the high-fidelity avatar described in Chapter 7.2 to build *SIVA* (Socially Intelligent Virtual Agent), an ECA capable of adapting to the conversational and expressive style of the user over the course of a conversation. The Avatar rig has facial presets for 27 bone joint controls and 20 phonetic mouth shape controls. The internal architecture of *SIVA* was built using the Platform for Situated Interaction (PSI) [27], allowing for online interaction with the Unreal Engine 4 [106] world through Airsim API [276].

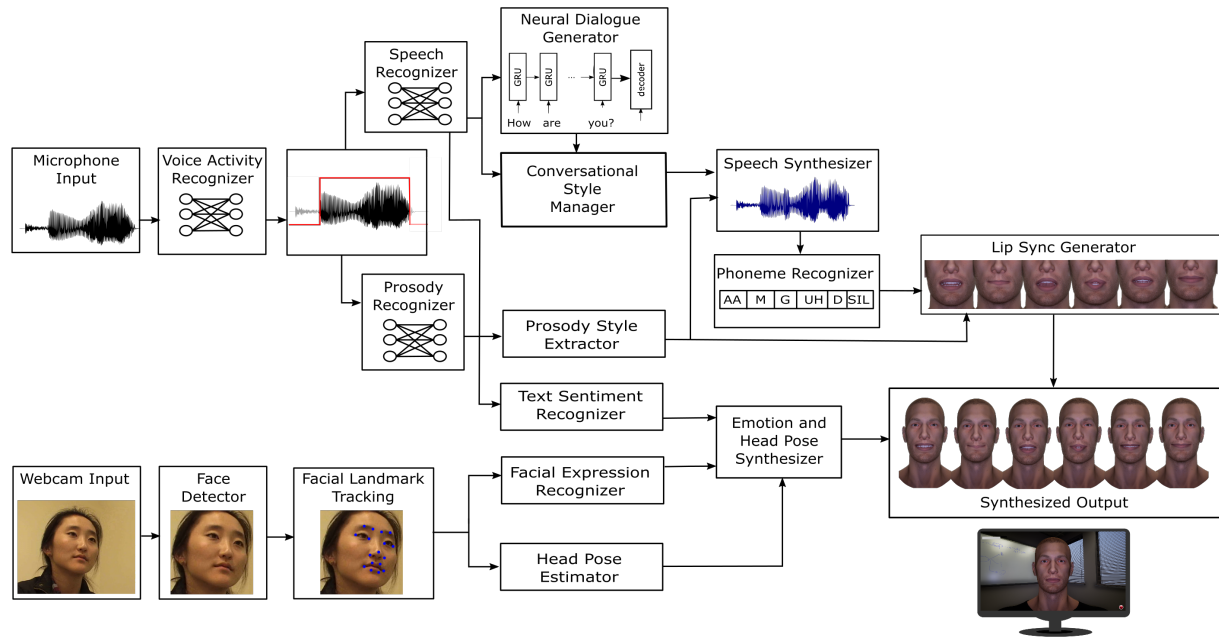


Figure 8.1: SIVA's Architecture: The system uses a fusion of both microphone and webcam inputs to drive the agent. Audio is segmented based on voice activity detection, and then speech and prosody recognition are performed. Generated dialogue is passed on to the Conversational Style manager, and along with the prosody style variables, conversational style aligned speech is synthesized, which is used for recognizing phonemes. The video pipeline detects the face to perform expression recognition, and head pose tracking. The output from the text sentiment recognizer, along with emotional expression probabilities, are used to generate expressions. Finally, the outputs from both the pipelines are composited together to synthesize SIVA's expressive face and lip sync.

The design of *SIVA* is truly multimodal, allowing for the capture of audio-visual features and synthesis of face-to-face conversations with a human. We built components to analyze the audio and video signals from the user for conversational style variables (Section 8.2.1) and expressive style variables (Section 8.2.2), and produce style aligned parameters for the expressive behavior of the agent (Section 8.3). Figure 8.1 shows the overview of the architecture. The following sections describe the main components of the architecture.

### *8.2.1 Conversational Pipeline*

The goal of this pipeline is to analyze the user’s speech for conversational style variables and synthesize *SIVA*’s speech adapting to that style. The audio pipeline utilizes a microphone (typically integrated with the webcam) and processes the audio stream (sampled at 16KHz). Microsoft Windows system voice activity detector detects the voice activity, and Microsoft speech API<sup>1</sup> performs the speech recognition.

### *Neural Dialogue Generator*

We utilize a deep learning-based dialogue generator model that is trained end-to-end on large scale unstructured Twitter conversations. It is capable of generating context-sensitive dialogue in response to a text-based input.

### *Conversational Style Manager*

This component extracts the linguistic style variables [302] from the user’s utterance and supplements the dialogue response with specific intents and scripted responses that the system is trained to recognize using Language Understanding and Intent Service (LUIS)<sup>2</sup>.

---

<sup>1</sup><https://azure.microsoft.com/en-us/services/cognitive-services/speech/>

<sup>2</sup><https://www.luis.ai/home>

State	SIVA's Behavior
When User is Speaking	Facial expressions and head pose based on the user's behavior (mimicking).
When SIVA is Speaking	Lipsyncing with speech and expressions based on the sentiment of the text utterance.

Table 8.1: SIVA's behavior is controlled depending on whether the user was speaking or SIVA is speaking.

### *Prosody Style Extractor*

We extract three different variables that describe the user's paralinguistic features from their speech: speech rate, pitch variance, and loudness variance.

### *Speech Synthesizer*

SIVA's conversational style aligned speech is synthesized using conversational style variables and prosody style variables. We perform the speech style alignment in two ways: (1) Adapting to the word choice variables and utterance length, and (2) Varying the prosody and pace of the response using SSML [35].

### 8.2.2 *Expressive Pipeline*

The expressive pipeline automatically recognizes and quantifies the behavior of the user and synthesizes SIVA's expressive response. This pipeline generates lip syncing and facial expressions based on the current conversational state to provide a perceptually valid interface for a more engaging face-to-face conversation.

SIVA's behavioral response comprises of two states as summarized in Table 8.1. When the user is speaking, and SIVA is listening to the user, we use the Emotion Recognizer and Head Pose Tracker to sense the user's facial expressions and head pose, and the Emotion and Head

Pose Synthesizer to generate *SIVA*'s synchronized non-verbal behavior in real-time. After the user's utterance, when it's *SIVA*'s turn to speak, we utilize a Phoneme Recognizer and Lip Sync Generator to synthesize lip syncing to the synthesized speech, along with facial expressions based on the output of the Text Sentiment Recognizer.

#### *Emotion Recognizer and Head Pose Tracker*

The expressive pipeline utilizes a webcam attached to the computer and processes the video frames. First, we detect the face in the video and extract the landmark positions of key facial features. Next, we use an emotion detection algorithm to analyze the facial region of interest and also track the user's head movement. The EmotionAPI <sup>3</sup> returns eight probabilities for each of the following basic facial expressions: anger, disgust, fear, joy, sadness, surprise, and neutral.

#### *Emotion and Head Pose Synthesizer*

The facially expressive response of *SIVA* mimics the user's facial expression to empathize with the user's affective state during a conversation. The facial expression probabilities from EmotionAPI control *SIVA*'s facial action units' (AUs) presets based on EMFACS [103] emotional facial action coding system (FACS). We also implement the agent's head gestures by controlling the head orientation of the 3D rig, matching the user's head movements. Facial mimicry is something that occurs in human-human interactions [121] and it has been found that mimicking facial expressions and gestures is one way to create more satisfactory interactions with an embodied agent [256]. Therefore, we employ this strategy in our interactions as the primary method of controlling *SIVA*'s behavior.

#### *Phoneme Recognizer*

The Phoneme Recognizer component automatically outputs a corresponding sequence of visual groups of phonemes (visemes) from a given continuous stream of audio samples generated by the

---

<sup>3</sup>Microsoft, Inc.

synthesized speech output. We use PocketSphinx [133] for phoneme recognition which returns probabilities for 39 phonemes + 1 for Silence [62], and map the phonemes to a set of 20 standard visemes [74].

### *Text Sentiment Recognizer*

The detected human user's utterance is fed to the Bing speech recognition<sup>4</sup> engine which returns the recognized text string. We use a natural language processing based model to analyze the text string for sentiments. We train the model on a large proprietary corpus of labeled utterances. We use these sentiments to control *SIVA*'s expressions while speaking.

### *Lip Sync Generator*

The sequence of visemes from the Phoneme Recognizer control *SIVA*'s corresponding viseme facial presets for synthesizing believable lip sync. We smooth out the viseme output by not allowing a viseme to change after a single frame. We also leverage the sentiment output to control the expressiveness of *SIVA*'s upper face while lip syncing. It results in an emotionally aware *SIVA*'s expressive response while speaking.

## **8.3 Style Matching and Synthesis**

*SIVA* performs conversational and expressive style matching based on the interaction style of the user during the conversation. The Audio and Video pipelines collect the input data and analyze it for specific style variables. *SIVA*'s response adapts to the style of the user by learning from these style variables in real-time. The style variables and the speech synthesis using these variables is described as follows.

---

<sup>4</sup><https://azure.microsoft.com/en-us/services/cognitive-services/speech/>








Emotion	Neutral	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Action Units	-	4+5+7+23	9+15+16 +20+26	1+2+4+5+7	6+12	1+4+15	1+2+5+26
SIVA's Expression							

Table 8.2: *SIVA*'s expressions and the action unit (AU) combinations to create these expressions.

### 8.3.1 Conversational Style Variables

*SIVA* performs Conversational style matching on two kinds of style variables as described in Section 6.4: (1) Content of the utterances like the use of pronouns, word repetition, utterance length, and (2) prosodic qualities like speech rate, pitch, and loudness.

### 8.3.2 Expressive Style Variables

The variables used to generate the expressive response for *SIVA* are described as follows.

#### *Expressions and Head movement*

The intensity of the expressiveness measures the user's affect during the interaction with the agent. We compute the facial expression probabilities for anger, disgust, fear, joy, sadness, surprise and neutral, and control the corresponding EMFACS [103] based AU presets of *SIVA* as shown in Table 8.2. Similarly, we generate *SIVA*'s head movement by tracking the user's head orientation (yaw and roll) and control the corresponding head pose values of the agent.

#### *Text sentiment*

The Text sentiment recognizer analyses users' utterances and outputs the probabilities of six sentiments: anger, neutral, joy, fear, sadness, and guilt. This captures the contextual polarity and emotional consciousness from the user's utterance. Similar to facial expression synthesis, we con-

trol the sentiment via corresponding facial AU presets to show the emotion on the upper part of the face when *SIVA* speaks to avoid the conflict between speech generation and facial expressions (e.g., *SIVA* should not express happiness while delivering a negative message).

### *Lip Sync intensity*

The lip sync intensity measure is driven by the loudness from the Prosody Style Extractor. The five levels of loudness are: x-soft, soft, medium, loud, and x-loud. These style variables are calculated from the user’s audio, and *SIVA* responds by lip syncing at the appropriate intensity.

## **8.4 User Study**

To investigate the effect of conversational and expressive style matching on users’ perceptions of the system, we conducted a user study comparing two interaction conditions—participants interacting with *SIVA* using conversational and expressive style matching (experimental) and others interacting with *SIVA* with no style matching (control). The user study protocol was approved by the Institutional Review Board (IRB) of the authors’ institution. We used a between-subjects design where each participant interacted with *SIVA* with only one of the interaction conditions. We designed five conversation tasks (Section tasks) focused on social interaction scenarios with approximately five minutes for each task, leading to over 10 hours of accumulated data. Participants were given two surveys to fill out: before and after the interaction with *SIVA*. For the study, we recruited 30 participants (15 females) between the ages of 23 and 55 (mean = 32.65, std = 8.33). We recruited people of different conversational styles (15 HC, 15 HI) and balanced them in each condition. All participants completed five conversation tasks over the course of  $\sim 25$  minutes.

### *8.4.1 Tasks and Questionnaires*

The users could interact with the agent as they deemed appropriate conversationally, and were given about five minutes to complete each task. We used the same scenario tasks as the conversational style matching user study described in Section [6.5.4](#).

We used the following self-report survey instruments to capture the experience of the users after the completion of all the tasks.

#### *Agent Expressions.*

To evaluate *SIVA*'s expressiveness, we asked the following questions to measure the degree of believable behaviour exhibited by *SIVA*. They were rated on scale of 1-7:

- To what degree did the agent reflect your emotions?
- To what degree did the agent exhibit believable nonverbal behavior?
- To what degree did the agent emotionally connect with you?

#### *Agent Impressions.*

To capture the participants' impressions of *SIVA*, we used the Godspeed questionnaire [15]. This survey was originally used for obtaining ratings on robots, therefore some minor modifications were made (we discarded ratings on movement). The Godspeed Questionnaire is the best validated measurement tool to evaluate perception of social interactions between humans and robots. It measures anthropomorphism, animacy, likability, perceived intelligence and perceived safety. We used it to capture the participants' impressions of *SIVA* during conversations. We asked the following questions on scale of 1-5.

Please rate your impression of the agent on these scales :

##### *Animacy*

- Dead : Alive
- Stagnant : Lively
- Mechanical : Organic
- Artificial : Lifelike
- Inert : Interactive
- Apathetic : Responsive

##### *Anthropomorphism*

- Fake : Natural
- Machinelike : Humanlike
- Unconscious : Conscious
- Artificial : Lifelike

#### *Likeability*

- Dislike : Like
- Unfriendly : Friendly
- Unkind : Kind
- Unpleasant : Pleasant
- Awful : Nice

#### *Perceived Intelligence*

- Incompetent : Competent
- Ignorant : Knowledgeable
- Irresponsible : Responsible
- Unintelligent : Intelligent
- Foolish : Sensible

#### *Perceived Task Load.*

To capture the users' subjective impression of the task load, we used the NASA Task-Load Index (TLX) [117] scale which is a standard in recording subjective task workload. We included a subset of the original NASA TLX subscales (mental demand, temporal demand, overall performance, effort, and frustration level) into the analysis, each of them being rated on a 7-point Likert scale. Here are the Perceived Task Load questions asked:

Please answer the following questions based on the interaction you just had with the agent on scale of 1-7.

- How mentally demanding was the task?
- How physically demanding was the task?
- How hurried or rushed was the pace of the task?
- How successful were you in accomplishing what you were asked to do?
- How hard did you have to work to accomplish your level of performance?

## 8.5 Results and Discussion

During the user study sessions, the interaction between *SIVA* and the participant lasted an average of 18.8 minutes (std: 4 mins), spending an average of 3 mins 45 secs in each of the five tasks. The results of preliminary data analysis showed one outlier in our data. For this participant, *SIVA* head orientation was not initialized accurately, which resulted in negative scoring from the participant's side. For our data analysis, we excluded the data for that participant. We focused on the impact of the visual and conversational style match condition vs. the control condition and the interaction between the participants' conversational styles: High Consideration (HC) or High Involvement (HI) with the agent matching or not.

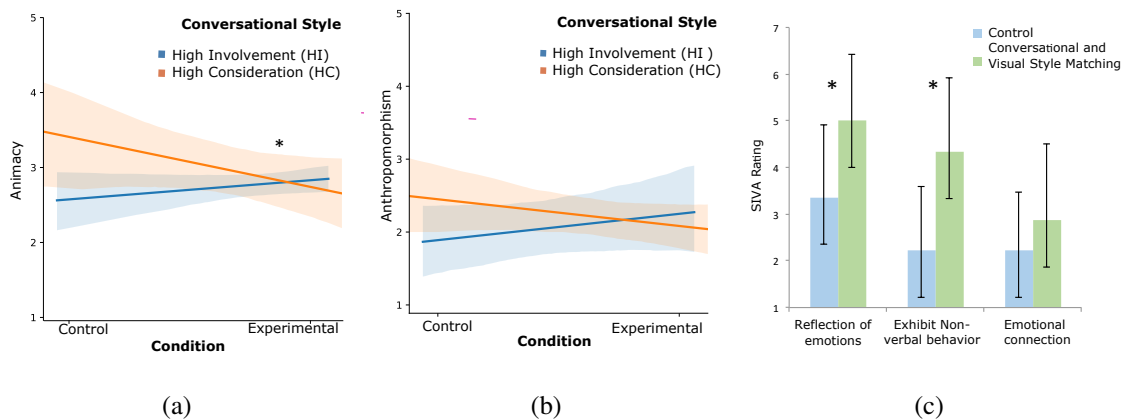


Figure 8.2: (a) *SIVA*'s non-verbal behavior and emotional expression plot across the user study conditions (high numbers mean a better rating). Interaction plot for 2-way ANOVA result for (b) Animacy. \* shows statistical significance and (c) Anthropomorphism. The interaction between the Conversational Style and User Study Condition is significant for Animacy. In all figures, \* shows statistical significance.

### 8.5.1 Perception of Animacy

We analyzed the four Godspeed indices (anthropomorphism, animacy, likeability and perceived intelligence) as composite measures obtained through the survey ratings for each index.

We performed a two-way ANOVA for ratings of each of the indices between the conversational style (HC or HI) and the user study condition (control vs. experimental). Out of the four indices, there was a significant interaction between the effects of condition and style ( $F(2, 25) = 4.38, p = 0.04$ ) for animacy. In this test, the independent variables “Condition” and “Style” did not have statistically significant main effects. As shown in Figure 8.2 (a), the impression of *SIVA*’s “Animacy” increased for HI conversational style participants between the control and the expressive style matching conditions. This result also showed that the trend was reversed for participants with HC conversation style. Thus, employing a lively, expressive style appears to be more important for HI compared to HC participants. For HC participants, the agent’s expressions may have been perceived as a distraction during the conversation. For anthropomorphism, the HI conversational style group rated *SIVA* more natural compared to the HC conversation style group, but the interaction was not significant ( $F(2, 25) = 2.32, p = 0.140$ ), as shown in Figure 8.2 (b). The differences in the interaction for likability and perceived intelligence were not significant, as the two conversational style groups rated similarly on these dimensions.

Overall, these results show that, for HI participants, matching expressions made the agent appear more animated and anthropomorphic; however, for HC participants, this was not the case. It is possible that, since the facial expressions of the agent were not entirely natural, the HC participants found them uncanny and the negative response to this was greater than the desired benefits of expressiveness. Whereas, for the HI participants, the opposite was true. This leads us to believe that the benefits of style matching for embodied agents may depend on, at the very least, the conversational style of the interlocutor, at least while the behaviors of the agent are not entirely naturalistic. This requires further investigation, including other personality and design factors that have yet to be explored. It is a rich area for future research.

### 8.5.2 *Non-verbal Behavior and Emotional Connection*

We evaluated the effectiveness of *SIVA*'s expressions by analyzing the ratings of the emotion questionnaire across conditions using independent sample t-tests. Figure 8.2 (c) shows the mean *SIVA* rating for the three questions from the emotion questionnaire across two user study conditions. The participants rated the visual and style matching condition to be statistically significantly more reflective of their emotions ( $5 \pm 1.41$ ) compared to the control condition ( $3.35 \pm 1.55$ ), ( $t(27) = 2.98, p = 0.006$ ). The results also show that the participants perceived the agent to exhibit reliably more convincing non-verbal behavior for the style matching condition ( $4.33 \pm 1.59$ ) compared to the control condition ( $2.21 \pm 1.37$ ) ( $t(27) = 3.83, p = 0.0006$ ). For the emotional connection with *SIVA* during the conversation, participants rated the style matching condition higher than the control, but we did not find significant differences in ratings across the conditions ( $t(27) = 1.19, p = 0.241$ ). The effect size of these measurements was smaller than for the other questions, and a larger sample may reveal a significant difference in the level of emotional connection experienced.

### 8.5.3 *Perceived Task Load*

A two-way ANOVA was run to examine the effect of condition and conversational style on the perceived task load composite ratings from the NASA TLX survey. There was no significant interaction between the effects of condition and conversational style on perceived task load ratings ( $F(2, 25) = 0.86, p = 0.36$ ) and there were no significant main effects.

### 8.5.4 *Participants' Feedback*

We collected feedback from the participants in the form of comments after their interaction with *SIVA*. In this section, we discuss some of these participant comments and refer to the participants in the format: *ParticipantID (Conversational Style, Condition)*.

In the experimental condition, participants noticed and commented on how the agent was able to reflect their emotions. Participant P12 (HI, E) said "*The agent reflected my expressions, even*

*when they meant I feel weird.*” Whereas in the control participants commented on the lack of expression. Participant P7 (HI, C) said *“Agent had no facial mannerisms that I can remember.”*

With regard to the interaction effect, it was observed that HC conversational style participants said that the conversation in the control condition felt genuine, whereas an HI style participant felt the control condition lacked expressiveness.

HC participant P17 (HC, C) mentioned *“The communication with the agent felt authentic..”*. Another HC participant P19 (HC, C) mentioned *“Responses over the course of the conversation were good.”* Suggesting that the lack of facial expressions and emotion were not troubling. Comparing this to HI participant P9 (HI, C), who said *“It was a bit uncanny that the head and eyes didn’t move even though the mouth did..”*. These comments are in line with our findings about an interaction between conversation style and the experimental condition with regard to animacy ratings. This trend is also supported by another HI conversational style participant with the expectation of visual expressiveness during a conversation, P22(HI, C) said *“The agent only appeared to move his mouth but the top half of his face was not the most believable when it came to speaking with him.. ”*.

However, a few participants also mentioned that the mimicking of the facial expressions by SIVA was not entirely natural. P8 (HI, E) mentioned: *“I thought the way the agent mimic my face and head movements was very creepy. Looks strange when it slightly smiled right when I did.”*. P24 (HC, E) said: *“Kind of weird that he literally mirrored my movements.”* The uncanny nature of the facial expression might explain why HC participants rated the animacy lower when SIVA had facial expressions compared to when it did not. The dialogue alone may have felt more natural to them. We discuss this more in the Limitations section.

Overall, the qualitative results showed that the HC conversational style participants paid more attention to the verbal communication and perceived the control condition to be more natural. On the other hand, HI conversational style participants noticed the lack of expressiveness during their conversations in the control condition, revealing that the visually expressive qualities were possibly more important to them, at least with this current design instantiation.

## **8.6 Limitations**

There are two main limitations to our current system. There is a delay in *SIVA*'s response to the user's utterance. This delay stems from our style matched speech synthesis component, which takes  $\approx 2$  secs to process and answer specific questions. In some cases, when the participant starts speaking before *SIVA*'s response, this leads to a speech overlap between the participant and *SIVA*. For our study, we explained the delay during the experimental briefing so that the participants understood the pace of the conversational tasks.

To avoid turn-taking errors that arise, we could implement a mechanisms where *SIVA* talks only during its turn. Another limitation of our design is in the mimicry of users' expressions by *SIVA*.

Mimicking human expressions limits the emotional expressiveness of *SIVA*. Ideally, the agent's visual response should be empathetic and believable to connect better with the user. We could overcome this limitation by generating *SIVA*'s expressive visual response from a combination of multimodal input signals (audio, physiological, video) for a better understanding of the context and not just mirroring the user's face, e.g., showing the agent as "thinking" before responding, by looking up in the air, etc.

## **8.7 Summary**

Our work addresses a key technical challenge in the emerging domain of interaction with ECAs. Emotional awareness and achieving realism without falling into the uncanny valley is critical for an engaging, open-ended conversational experience with an ECA. This work is taking the required first steps towards a more animated and emotionally responsive ECA, by attempting to match the visual and conversational style of the user. Much more work needs to be done to understand what parameters matter in visual and conversational style matching of ECAs.

Overall, the participants from our study believed the expressiveness of *SIVA* reflected their emotions and exhibited perceptually and emotionally valid, non-verbal behavior during the conversation when compared with an agent that did not. Our study also demonstrated that visually expressive qualities lead to more “animated” impressions of *SIVA* for the participants with an HI conversational style, whereas the participants with the HC conversational style rated the control (non-matching condition) higher when evaluating these qualities.

## Chapter 9

### CONCLUSION

This thesis presents data-driven learning-based techniques for enhanced performance-based facial animation. We developed novel methods to learn and automate the process of perceptually valid expression retargeting from humans to characters, real-time lip-synchronization for animation, and building an emotionally aware embodied conversational agent.

We first tackled the problem of understanding human facial expressions and retargeting them to multiple stylized characters. In Chapter 2, we introduced a novel expression retrieval approach from humans to multiple stylized characters in 2D. We evaluated our method on a set of retrieval tasks on our collected stylized character dataset of expressions. We showed that the ranking order predicted by the proposed features is highly correlated with the ranking order provided by a facial expression expert and Mechanical Turk experiments. In Chapter 3, we presented a multi-stage deep learning network that uses images of human facial expressions to generate perceptually valid and geometrically consistent stylized character expressions using a deep learning framework. Our approach is unique in that it uses geometry plus perceptual validity rather than pure geometry-based mathematical operations. The qualitative and quantitative evaluation of our method based on Mechanical Turk tests showed the high perceptual accuracy of our expression transfer results. In Chapter 4, we demonstrated retargeting of facial expressions on stylized animal characters with different facial geometry as compared to humans using Bayesian latent variable models. Future work will integrate the universal character into the expression retargeting pipeline and build an end-to-end system, as well as perform a thorough analysis and systematic study to verify the non-humanoid expression retargeting results. For now, our deep learning-based expression transfer approach is specifically targeted to enhance the performance-based animation for 3D characters. However, the notion of universal animation parameters could be potentially extended beyond 3D

animation (e.g., Cut-Out animation, 2D animation and Claymation).

Our real-time lip sync model described in Chapter 5 utilized a deep learning model to convert the vocal performance of an actor into a corresponding viseme sequence for a 2D layered character with less than 200ms latency. Our extensive human judgement experiments show that our results are preferred over several competing methods, including those that only support offline (non-live) processing. For future work, we will consider fine-tuning the lip sync model for the animator's style and learning a perceptually-based viseme loss that reflects the most relevant perceptual differences between lip sync sequences.

In Chapter 6, we designed an end-to-end voice-based conversational agent capable of multi-turn dialogue to study the effect of conversational style matching on participants' perception of the agent. Our user study results showed that users with high consideration conversational styles reported the agent to be more trustworthy when it matched their conversational style. Whereas, users with high involvement conversational styles were indifferent. Future work will consider the use of an end-to-end conversational agent that combines visual information with textual inputs to generate image-grounded dialogue. We also aim to deploy an agent longitudinally to measure the effects of style matching over longer interactions and in more natural contexts.

We validated our perceptually valid expression retargeting, including lip sync, by designing a virtual agent as described in Chapter 7, and enhanced its facial expression and lip sync capabilities. In Chapter 8, we presented an expressive, embodied conversational agent that can recognize human behavior during open-ended conversations and automatically align its responses to the conversational and expressive style of the user. Our user study results show that the participants rated the style matching agent as being more empathetic and believable than the control. Based on a total of almost 10 hours of interaction, participants who preferred interpersonal involvement evaluated the agent as significantly more animated as compared to the participants who valued consideration and independence. For future work, we will consider the implementation of a better turn-taking mechanism, analysis of the conversational errors, reducing the delay in the agent's response and generating agent's expressive visual response from a combination of multimodal input signals instead of mimicking the facial expressions for a better understanding of the context.

We see many exciting applications of our work:

**Enhance performance-based animation.** The result of our research can improve visual storytelling applications, including animated films, gaming, online marketing, VR/AR experiences, and robotics. In all of these applications, higher quality character expressions can enhance the participant’s story experience and build a more effective connection between the story designer(s) and their intended audience. Another relevant application for animation could be for real-time (“live”) animation settings where facial performance is a convenient input modality, and it provides novices a way to tell stories with animation via performance.

**Generate convincing expressions.** Our approach can be used to generate facial expressions with different kinds of variations in facial appearance that are both possible and likely to occur in a given context. We should be able to provide semantic descriptions, including identity and emotion description, that would provide rich information to constrain the underlying representations for generating animated characters usefully. For example, - “create a character looking surprised with an open mouth and wide eyes.”

**Feedback.** Our expression retargeting perceptual model can be used to accurately predict the clarity and intensity of a character’s facial expression interactively, thus providing feedback to an animator. This can save time by quickly creating good starting points for further refinement, or perhaps for students/novices to learn animation. This extends to lip sync tasks as well, where auto lip sync makes it quicker and more immediate for live settings.

**Machine Learning for Animation.** Our work demonstrates a way to encode artistic rules for facial animation and lip sync with deep neural networks. We believe there are many more opportunities to apply modern machine learning techniques to improve animation workflows. Thus far, one challenge for this domain has been the paucity of training data, which is expensive to collect. However, as we show in this work, there may be ways to leverage structured data and automatic editing algorithms to maximize the utility of hand-crafted animation data. Overall, applying machine learning-based methods and a better understanding of performance mapping to the output will enhance the performance-based animation experience and will also positively impact the creative authoring tools in the field of animation.

## BIBLIOGRAPHY

- [1] R Adolphs. Recognizing emotion from facial expressions: Psychological and neurological mechanisms. *Behavioral and Cognitive Neuroscience Reviews*, 1.1:21–62, March 2002.
- [2] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: Reconstruction and parameterization from range scans. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, page 587–594, New York, NY, USA, 2003. ACM.
- [3] Zara Ambadar, Jonathan W. Schooler, and Jeffrey F. Cohn. Deciphering the enigmatic face the importance of facial dynamics in interpreting subtle facial expressions. *Psychological Science*, 16(5):403–410, May 2005. PMID: 15869701.
- [4] R. Amini and C. Lisetti. HapFACS: an open source API/Software to generate FACS-Based expressions for ECAs animation and for corpus generation. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 270–275, September 2013.
- [5] Deepali Aneja, Bindita Chaudhuri, Alex Colburn, Gary Faigin, Linda Shapiro, and Barbara Mones. Learning to generate 3d stylized character expressions from humans. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 160–169. IEEE, 2018.
- [6] Deepali Aneja, Alex Colburn, Gary Faigin, Linda Shapiro, and Barbara Mones. Modeling stylized character expressions via deep learning. In *Asian conference on computer vision*, pages 136–153. Springer, 2016.
- [7] Deepali Aneja, Rens Hoegen, Daniel McDuff, and Mary Czerwinski. Designing style matching conversational agents. 2019.
- [8] Deepali Aneja, Daniel McDuff, and Shital Shah. A high-fidelity open embodied avatar with lip syncing and expression capabilities. In *2019 International Conference on Multimodal Interaction, ICMI '19*, pages 69–73, New York, NY, USA, 2019. ACM.
- [9] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, April 1988.
- [10] Anonymous. Planes of the head. <http://www.planesofthehead.com/>, 1976.

- [11] Archer at ComicCon, 2017.
- [12] Ali Arya, Steve DiPaola, and Avi Parush. Perceptually valid facial expressions for character-based applications. *International Journal of Computer Games Technology*, 2009:e462315, March 2009.
- [13] Autism and Beyond. Duke university school of medical, 2015.
- [14] Stephen W Bailey, Martin Watt, and James F O’Brien. Repurposing hand animation for interactive applications. In *Symposium on Computer Animation*, pages 97–106, 2016.
- [15] Christoph Bartneck, Dana Kulić, Elizabeth Croft, and Susana Zoghbi. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International journal of social robotics*, 1(1):71–81, 2009.
- [16] Christoph Bartneck and Juliane Reichenbach. Subtle emotional expressions of synthetic characters. *Int. J. Hum.-Comput. Stud.*, 62(2):179–192, February 2005.
- [17] Joseph Bates. The role of emotion in believable agents. *Commun. ACM*, 37(7):122–125, July 1994.
- [18] Nikolaus Bee, Bernhard Falk, and Elisabeth Andr . Simplified facial animation control utilizing novel input devices: A comparative study. In *Proceedings of the 14th International Conference on Intelligent User Interfaces, IUI ’09*, page 197–206, New York, NY, USA, 2009. ACM.
- [19] Štefan Beňuš. Conversational entrainment in the use of discourse markers. In *Recent Advances of Neural Network Models and Applications*, pages 345–352. Springer, 2014.
- [20] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.
- [21] Vinay Bettadapura. Face expression recognition and analysis: The state of the art. *arXiv:1203.6722 [cs]*, March 2012.
- [22] Timothy Bickmore and Justine Cassell. how about this weather? social dialogue with embodied conversational agents. In *Proc. AAI Fall Symposium on Socially Intelligent Agents*, 2000.
- [23] Timothy Bickmore and Justine Cassell. Relational agents: a model and implementation of building user trust. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 396–403. ACM, 2001.

- [24] Timothy Bickmore and Justine Cassell. Social dialogue with embodied conversational agents. *Advances in natural multimodal dialogue systems*, 30:23–54, 2005.
- [25] P. Blair. *How to Animate Film Cartoons*. W.T. Foster, 1980.
- [26] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. *Computer Graphics Forum*, 22(3):641–650, September 2003.
- [27] Dan Bohus, Sean Andrist, and Mihai Jalobeanu. Rapid development of multimodal interactive systems: a demonstration of platform for situated intelligence. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 493–494. ACM, 2017.
- [28] Sofien Bouaziz and Mark Pauly. Semi-supervised facial animation retargeting. Technical report, 2014.
- [29] Sofien Bouaziz, Yangang Wang, and Mark Pauly. Online modeling for realtime facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):40, 2013.
- [30] Matthew Brand. Voice puppetry. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 21–28. ACM Press/Addison-Wesley Publishing Co., 1999.
- [31] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.
- [32] Ian Buck, Adam Finkelstein, Charles Jacobs, Allison Klein, David H Salesin, Joshua Seims, Richard Szeliski, and Kentaro Toyama. Performance-driven hand-drawn animation. In *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 101–108. ACM, 2000.
- [33] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon’s mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5, February 2011.
- [34] Peter Burkert, Felix Trier, Muhammad Zeshan Afzal, Andreas Dengel, and Marcus Liwicki. Dexpression: Deep convolutional neural network for expression recognition. *arXiv preprint arXiv:1509.05371*, 2015.
- [35] Daniel C Burnett, Mark R Walker, and Andrew Hunt. Speech synthesis markup language (ssml) version 1.0. *W3C recommendation*, 7, 2004.

- [36] Nester Burtnyk and Marcelli Wein. Computer-generated key-frame animation. *Journal of the SMPTE*, 80(3):149–153, 1971.
- [37] A.J. Calder, A.W. Young, J. Keane, and M. Dean. Configural information in facial expression perception. *Journal of Experimental Psychology: Human Perception and Performance*, 26(2):527–551, April 2000.
- [38] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 3d shape regression for real-time facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):41, 2013.
- [39] Yong Cao, Wen C Tien, Petros Faloutsos, and Frédéric Pighin. Expressive speech-driven facial animation. *ACM Transactions on Graphics (TOG)*, 24(4):1283–1302, 2005.
- [40] Luca Cappelletta and Naomi Harte. Phoneme-to-viseme mapping for visual speech recognition. In *ICPRAM (2)*, pages 322–329, 2012.
- [41] Justine Cassell. Embodied conversational agents: representation and intelligence in user interfaces. *AI magazine*, 22(4):67–67, 2001.
- [42] Justine Cassell and Timothy Bickmore. Negotiated collusion: Modeling social language and its relationship effects in intelligent agents. *User modeling and user-adapted interaction*, 13(1-2):89–132, 2003.
- [43] Justine Cassell, Joseph Sullivan, Elizabeth Churchill, and Scott Prevost. *Embodied conversational agents*. MIT press, 2000.
- [44] Edwin Catmull. A system for computer generated movies. In *Proceedings of the ACM annual conference-Volume 1*, pages 422–431. ACM, 1972.
- [45] Edwin Catmull. The problems of computer-assisted animation. In *ACM SIGGRAPH Computer Graphics*, volume 12, pages 348–353. ACM, 1978.
- [46] Justine Cauell, Tim Bickmore, Lee Campbell, and Hannes Vilhjálmsón. Designing embodied conversational agents. *Embodied conversational agents*, pages 29–63, 2000.
- [47] The Late Night Show CBS, 2017.
- [48] Jin-xiang Chai, Jing Xiao, and Jessica Hodgins. Vision-based control of 3d facial animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 193–206. Eurographics Association, 2003.

- [49] Character Animator. *Adobe After Effects CC 2016*. Adobe Systems Incorporated, San Jose, CA 95110-2704, 2016.
- [50] @\_CHINOSAUR. "venue is too cold" #bingo #chi2014. Tweet, May 2014. Retrieved February 2, 2015 from [https://twitter.com/\\_CHINOSAUR/status/461864317415989248](https://twitter.com/_CHINOSAUR/status/461864317415989248).
- [51] Hyunjoo Cho and Jeeho Lee. Direct retargeting method from facial capture data to facial rig. *Journal of The Korea Computer Graphics Society*, 22(2):11–19, 2016.
- [52] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.
- [53] Erika Chuang and Chris Bregler. Performance driven facial animation using blendshape interpolation. *Computer Science Technical Report, Stanford University*, 2(2):3, 2002.
- [54] Michael M Cohen, Dominic W Massaro, et al. Modeling coarticulation in synthetic visual speech. *Models and techniques in computer animation*, 92:139–156, 1993.
- [55] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *J. Artif. Int. Res.*, 4(1):129–145, March 1996.
- [56] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.
- [57] Tim F Cootes, Mircea C Ionita, Claudia Lindner, and Patrick Sauer. Robust and accurate shape model fitting using random forest regression voting. In *European Conference on Computer Vision*, pages 278–291. Springer, 2012.
- [58] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [59] CrazyTalk, 2017.
- [60] Andreas Damianou, Carl Ek, Michalis Titsias, and Neil Lawrence. Manifold relevance determination. *arXiv preprint arXiv:1206.4610*, 2012.
- [61] Charles Darwin. *The Expression of the Emotions in Man and Animals*. 1872. The original was published 1898 by Appleton, New York. Reprinted 1965 by the University of Chicago Press, Chicago and London.

- [62] Donald M Decker et al. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.
- [63] Salil Deena and Aphrodite Galata. Speech-driven facial animation using a shared gaussian process latent variable model. In *International Symposium on Visual Computing*, pages 89–100. Springer, 2009.
- [64] Zhigang Deng and Xiaohan Ma. Perceptually guided expressive facial animation. In *SCA '08: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, July 2008.
- [65] David DeVault, Ron Artstein, Grace Benn, Teresa Dey, Ed Fast, Alesia Gainer, Kallirroi Georgila, Jon Gratch, Arno Hartholt, Margaux Lhomme, et al. Simsensei kiosk: A virtual human interviewer for healthcare decision support. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1061–1068. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
- [66] Abhinav Dhall et al. Collecting large, richly annotated facial-expression databases from movies. 2012.
- [67] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 2106–2112. IEEE, 2011.
- [68] Hamdi Dibeklioglu, Albert Salah, and Theo Gevers. Like father, like son: Facial expression dynamics for kinship verification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1497–1504, 2013.
- [69] Piotr Diddy, Tobias Ritschel, Elmar Eisemann, Karol Myszkowski, and Hans-Peter Seidel. A perceptual model for disparity. *ACM Trans. Graph.*, 30(4):96:1–96:10, July 2011.
- [70] Hui Ding, Shaohua Kevin Zhou, and Rama Chellappa. Facenet2expnet: Regularizing a deep face recognition net for expression recognition. *arXiv preprint arXiv:1609.06591*, 2016.
- [71] Guillaume-Benjamin Duchenne and R Andrew Cuthbertson. *The mechanism of human facial expression*. Cambridge university press, 1990.
- [72] Melanie Dumas. Emotional expression recognition using support vector machines. In *Proceedings of International Conference on Multimodal Interfaces*. Citeseer, 2001.
- [73] Dynamixyz. Performer suite. <http://www.dynamixyz.com/>.

- [74] Pif Edwards, Chris Landreth, Eugene Fiume, and Karan Singh. Jali: an animator-centric viseme model for expressive lip synchronization. *ACM Transactions on Graphics (TOG)*, 35(4):127, 2016.
- [75] Bradley Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. CRC Press, May 1994.
- [76] P. Ekman. Strong evidence for universals in facial expressions: A reply to russell’s mistaken critique. *Psychological Bulletin*, 115(2):268–287, March 1994.
- [77] P. Ekman. Basic emotions. *Handbook of cognition and emotion*, 1999.
- [78] P. Ekman and W.V. Friesen. *Unmasking the face: a guide to recognizing emotions from facial expressions*. 1975.
- [79] P. Ekman and W.V. Friesen. Facial action coding system: A technique for the measurement of facial movement. *Consulting Psychologists Press*, 1978.
- [80] P. Ekman, W.V. Friesen, M. O’Sullivan, and A. Chan. Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of Personality and Social Psychology*, 53(4):712–717, October 1987.
- [81] Paul Ekman. Universals and cultural differences in facial expressions of emotion. In J Cole, editor, *Nebraska Symposium on Motivation*, pages 207–282. University of Nebraska Press, Lincoln, 1972.
- [82] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.
- [83] Rosenberg Ekman. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997.
- [84] Kevin El Haddad, Hüseyin Çakmak, Emer Gilmartin, Stéphane Dupont, and Thierry Dutoit. Towards a listening agent: a system generating audiovisual laughs and smiles to show interest. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 248–255. ACM, 2016.
- [85] Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic. Discriminative shared gaussian processes for multiview and view-invariant facial expression recognition. *IEEE transactions on image processing*, 24(1):189–204, 2014.
- [86] Ahmed Elmatarawi. Cody rig. <http://www.cgmeetup.net/home/cody-dog-rig-free-maya-rig-cartoon-character-rig/>.

- [87] Greg Elofson. Developing trust with intelligent agents: An exploratory study. In *Trust and deception in virtual societies*, pages 125–138. Springer, 2001.
- [88] Antti J Eronen, Vesa T Peltonen, Juha T Tuomi, Anssi P Klapuri, Seppo Fagerlund, Timo Sorsa, Gaëtan Lorho, and Jyri Huopaniemi. Audio-based context recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):321–329, 2006.
- [89] Irfan Essa, Sumit Basu, Trevor Darrell, and Alex Pentland. Modeling, tracking and interactive animation of faces and heads//using input from video. In *Computer Animation'96. Proceedings*, pages 68–79. IEEE, 1996.
- [90] N.L. Etcoff and J.J. Magee. Categorical perception of facial expressions. *Cognition*, 44(3):227–240, September 1992.
- [91] Tony Ezzat and Tomaso Poggio. Miketalk: A talking facial display based on morphing visemes. In *Computer Animation 98. Proceedings*, pages 96–102. IEEE, 1998.
- [92] C Fabian Benitez-Quiroz, Ramprakash Srinivasan, and Aleix M Martinez. Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5562–5570, 2016.
- [93] Facebook, 2017.
- [94] Faceshift. Faceshift. <http://faceshift.com/studio/2015.2/>.
- [95] Faceware. Faceware live. <http://facewaretech.com/products/software/>.
- [96] G. Faigin. *The Artist's Complete Guide to Facial Expression*. Watson-Guption, 1991.
- [97] Ian Failes. How 'The Simpsons' used Adobe Character Animator to create a live episode. <http://bit.ly/1swlyXb>, 2016. Published: 2016-05-18.
- [98] Bo Fan, Lijuan Wang, Frank K Soong, and Lei Xie. Photo-real talking head with deep bidirectional lstm. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4884–4888. IEEE, 2015.
- [99] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. In Eugene Fiume, editor, *Proceedings of SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 15–22. ACM, ACM Press / ACM SIGGRAPH, 2001.

- [100] Ronald Aylmer Fisher. *The design of experiments*. Oliver And Boyd; Edinburgh; London, 1937.
- [101] Kathleen Kara Fitzpatrick, Alison Darcy, and Molly Vierhile. Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (woebot): a randomized controlled trial. *JMIR mental health*, 4(2), 2017.
- [102] The Simpsons Fox, 2016.
- [103] Wallace V Friesen, Paul Ekman, et al. Emfacs-7: Emotional facial action coding system. *Unpublished manuscript, University of California at San Francisco*, 2(36):1, 1983.
- [104] T. Fukuda, J. Myung-Jin, M. Nakashima, F. Arai, and Y. Hasegawa. Facial expressive robotic head system for human-robot communication and its application in home environment. *Proceedings of the IEEE*, 92(11):1851–1865, November 2004.
- [105] Shoichi Furukawa, Tsukasa Fukusato, Shugo Yamaguchi, and Shigeo Morishima. Voice animator: Automatic lip-synching in limited animation by audio. In *International Conference on Advances in Computer Entertainment*, pages 153–171. Springer, 2017.
- [106] Epic Games. Unreal engine. *Online: <https://www.unrealengine.com>*, 2007.
- [107] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, David S Pallett, Nancy L Dahlgren, and Victor Zue. Timit acoustic-phonetic continuous speech corpus. *Linguistic data consortium*, 10(5):0, 1993.
- [108] Michaël Gilbert, Samuel Demarchi, and Isabel Urdapilleta. Facshuman a software to create experimental material by modeling 3d facial expression. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pages 333–334. ACM, 2018.
- [109] B. Gooch, E. Reinhard, and A. Gooch. Human facial illustrations: Creation and psychophysical evaluation. *ACM Transactions on Graphics*, 23(1):27–44, 2004.
- [110] Jonathan Gratch, Gale M Lucas, Aisha King, and Louis-Philippe Morency. It’s only a computer: the impact of human-agent interaction in clinical interviews. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 85–92. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
- [111] Jonathan Gratch, Ning Wang, Jillian Gerten, Edward Fast, and Robin Duffy. Creating rapport with virtual agents. In *International workshop on intelligent virtual agents*, pages 125–138. Springer, 2007.

- [112] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1764–1772, 2014.
- [113] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE, 2013.
- [114] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on*, pages 6645–6649. IEEE, 2013.
- [115] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.
- [116] Kun Han, Dong Yu, and Ivan Tashev. Speech emotion recognition using deep neural network and extreme learning machine. In *Fifteenth annual conference of the international speech communication association*, 2014.
- [117] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, 52:139–183, 1988.
- [118] J.V. Haxby, E.A. Hoffman, and M.I. Gobbiniq. The distributed human neural system for face perception. *Trends in Cognitive Science*, 4(6):223–233, June 2000.
- [119] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [120] U Hess, S Blairy, and R E Kleck. The Intensity of Emotional Facial Expressions and Decoding Accuracy - Springer. *Journal of Nonverbal Behavior*, 1997.
- [121] Ursula Hess and Sylvie Blairy. Facial mimicry and emotional contagion to dynamic emotional facial expressions and their influence on decoding accuracy. *International journal of psychophysiology*, 40(2):129–141, 2001.
- [122] Julia Hirschberg. Speaking more like you: Entrainment in conversational speech. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.
- [123] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

- [124] Rens Hoegen, Deepali Aneja, Daniel McDuff, and Mary Czerwinski. An end-to-end conversational style matching agent. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents, IVA '19*, pages 111–118, New York, NY, USA, 2019. ACM.
- [125] Rens Hoegen, Deepali Aneja, Daniel McDuff, and Mary Czerwinski. An end-to-end conversational style matching agent. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents, IVA '19*, pages 111–118, New York, NY, USA, 2019. ACM.
- [126] Kate S Hone and Robert Graham. Towards a tool for the subjective assessment of speech system interfaces (sassi). *Natural Language Engineering*, 6(3-4):287–303, 2000.
- [127] Mohammed Ehsan Hoque, Matthieu Courgeon, Jean-Claude Martin, Bilge Mutlu, and Rosalind W Picard. Mach: My automated conversation coach. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 697–706. ACM, 2013.
- [128] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- [129] Shaobo Hou, Aphrodite Galata, Fabrice Caillette, Neil Thacker, and Paul Bromiley. Real-time body tracking using a gaussian process latent variable model. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.
- [130] Jing Huang and Catherine Pelachaud. Expressive body animation pipeline for virtual agent. In *International Conference on Intelligent Virtual Agents*, pages 355–362. Springer, 2012.
- [131] Bernd Huber and Daniel McDuff. Facial expression grounded conversational dialogue generation. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 365–372. IEEE, 2018.
- [132] Bernd Huber, Daniel McDuff, Chris Brockett, Michel Galley, and Bill Dolan. Emotional dialogue generation using image-grounded language models. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 277. ACM, 2018.
- [133] David Huggins-Daines, Mohit Kumar, Arthur Chan, Alan W Black, Mosur Ravishankar, and Alexander I Rudnicky. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE, 2006.
- [134] Benjamin Inden, Zofia Malisz, Petra Wagner, and Ipke Wachsmuth. Timing and entrainment of multimodal backchanneling behavior for an embodied conversational agent. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 181–188. ACM, 2013.

- [135] CINECA-Consortio Interuniversitario. Lucia: An open source 3d expressive avatar for multimodal hmi.
- [136] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [137] Horace HS Ip and CS Chan. Script-based facial gesture and speech animation using a nurbs based face model. *Computers & Graphics*, 20(6):881–891, 1996.
- [138] David Ireland, Christina Atay, Jacki Liddle, Dana Bradford, Helen Lee, Olivia Rushin, Thomas Mullins, Dan Angus, Janet Wiles, Simon McBride, et al. Hello harlie: Enabling speech monitoring through chat-bot conversations. *Studies in health technology and informatics*, 227:55–60, 2016.
- [139] Katherine Isbister and Patrick Doyle. Design and evaluation of embodied conversational agents: A proposed taxonomy. In *The first international joint conference on autonomous agents & multi-agent systems*, 2002.
- [140] C.E. Izard. Innate and universal facial expressions: evidence from developmental and cross-cultural research. *Psychological Bulletin*, 115(2):288–299, March 1994.
- [141] Rachael E. Jack, Oliver G. B. Garrod, and Philippe G. Schyns. Dynamic facial expressions of emotion transmit an evolving hierarchy of signals over time. *Current Biology*, 24(2):187–192, January 2014. PMID: 24388852.
- [142] Rachael E. Jack, Oliver G. B. Garrod, Hui Yu, Roberto Caldara, and Philippe G. Schyns. Facial expressions of emotion are not culturally universal. *Proceedings of the National Academy of Sciences*, pages 7241–7244, April 2012. PMID: 22509011.
- [143] Igor Jauk, Ipke Wachsmuth, and Petra Wagner. Dynamic perception-production oscillation model in human-machine communication. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 213–216. ACM, 2011.
- [144] László A Jeni, András Lőrincz, Zoltán Szabó, Jeffrey F Cohn, and Takeo Kanade. Spatio-temporal event classification using time-series kernel based structured sparsity. In *Computer Vision–ECCV 2014*, pages 135–150. Springer, 2014.
- [145] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

- [146] J. Jimenez, T. Scully, N. Barbosa, C. Donner, X. Alvarez, P. Matts T. Vieira, V. Orvalho, D. Gutierrez, and T. Weyrich. A practical appearance model for dynamic facial color. *SIG-GRAPH ASIA '10*, (141), 2010.
- [147] Daniel J Jobson, Zia-ur Rahman, and Glenn A Woodell. Retinex image processing: Improved fidelity to direct visual observation. In *Proceedings of the IS&T Fourth Color Imaging Conference: Color Science, Systems, and Applications*, volume 4, pages 124–125. The Society for Imaging Science and Technology, 1995.
- [148] Oliver P John and Sanjay Srivastava. The big five trait taxonomy: History, measurement, and theoretical perspectives. *Handbook of personality: Theory and research*, 2(1999):102–138, 1999.
- [149] O. Johnston and F. Thomas. *The Illusion of Life: Disney Animation*. Disney Editions, 1995.
- [150] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.
- [151] Heechul Jung, Sihaeng Lee, Junho Yim, Sunjeong Park, and Junmo Kim. Joint fine-tuning in deep neural networks for facial expression recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2983–2991, 2015.
- [152] Y. Jung, C. Weber, J. Keil, and T. Franke. Real-time rendering of skin changes caused by emotions. *IVA '09 Proceedings of the 9th International Conference on Intelligent Virtual Agents*, pages 504–505, 2009.
- [153] Samira Ebrahimi Kahou, Xavier Bouthillier, Pascal Lamblin, Caglar Gulcehre, Vincent Michalski, Kishore Konda, Sébastien Jean, Pierre Froumenty, Yann Dauphin, Nicolas Boulanger-Lewandowski, et al. Emonets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, 10(2):99–111, 2016.
- [154] M. Kanoh, S. Kato, and H. Itoh. Facial expressions using emotional space in sensitivity communication robot ”ifbot”. *Intelligent Robots and Systems, 2004*, 2:1586–1591, 2004.
- [155] Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics (TOG)*, 36(4):94, 2017.
- [156] Daniel Kartch. *Efficient Rendering and Compression for Full-Parallax Computer-Generated Holographic Stereograms*. PhD thesis, Cornell University, 2000.
- [157] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

- [158] MASE Kenji. Recognition of facial expression from optical flow. *IEICE TRANSACTIONS on Information and Systems*, 74(10):3474–3483, 1991.
- [159] Natasha Kholgade, Iain Matthews, and Yaser Sheikh. Content retargeting using parameter-parallel facial layers. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 195–204. ACM, 2011.
- [160] Bo-Kyeong Kim, Hwaran Lee, Jihyeon Roh, and Soo-Young Lee. Hierarchical committee of deep cnns with exponentially-weighted decision fusion for static facial expression recognition. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 427–434. ACM, 2015.
- [161] Taehwan Kim, Yisong Yue, Sarah Taylor, and Iain Matthews. A decision tree framework for spatiotemporal sequence prediction. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 577–586. ACM, 2015.
- [162] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.
- [163] Ross D. King, Kenneth E. Whelan, Ffion M. Jones, Philip G. K. Reiser, Christopher H. Bryant, Stephen H. Muggleton, Douglas B. Kell, and Stephen G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–252, January 2004.
- [164] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [165] Scott R. Klemmer, Michael Thomsen, Ethan Phelps-Goodman, Robert Lee, and James A. Landay. Where do web sites come from?: Capturing and interacting with design history. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 1–8, New York, NY, USA, 2002. ACM.
- [166] Hiroshi Kobayashi and Fumio Hara. Facial interaction between animated 3d face robot and human beings. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, volume 4, pages 3732–3737. IEEE, 1997.
- [167] Nishanth Koganti, Tomoya Tamei, Kazushi Ikeda, and Tomohiro Shibata. Bayesian non-parametric learning of cloth models for real-time state estimation. *IEEE Transactions on Robotics*, 33(4):916–931, 2017.

- [168] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [169] Sumedha Kshirsagar. A multilayer personality model. In *Proceedings of the 2Nd International Symposium on Smart Graphics, SMARTGRAPH '02*, page 107–115, New York, NY, USA, 2002. ACM.
- [170] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
- [171] Hayden Landis. Production-ready global illumination. ACM SIGGRAPH 2002 Course #16 Notes, July 2002.
- [172] Liliana Laranjo, Adam G Dunn, Huong Ly Tong, Ahmet Baki Kocaballi, Jessica Chen, Rabia Bashir, Didi Surian, Blanca Gallego, Farah Magrabi, Annie YS Lau, et al. Conversational agents in healthcare: a systematic review. *Journal of the American Medical Informatics Association*, 25(9):1248–1258, 2018.
- [173] J. Lasseter. Principles of traditional animation applied to 3d computer animation. *SIGGRAPH*, pages 34–44, 1987.
- [174] John Lasseter. Principles of traditional animation applied to 3d computer animation. *SIGGRAPH Comput. Graph.*, 21(4):35–44, August 1987.
- [175] John Lasseter. Principles of traditional animation applied to 3D computer animation. In *the 14th annual conference*, 1987.
- [176] Chan-Su Lee, A. Elgammal, and D. Metaxas. Synthesis and control of high resolution facial expressions for visual interactions. In *2006 IEEE International Conference on Multimedia and Expo*, pages 65–68, July 2006.
- [177] Soonkyu Lee and DongSuk Yook. Audio-to-visual conversion using hidden markov models. *PRICAI 2002: Trends in Artificial Intelligence*, pages 9–20, 2002.
- [178] Xie Lei, Jiang Dongmei, Ilse Ravyse, Wemer Verhelst, Hichem Sahli, Velina Slavova, and Zhao Rongchun. Context dependent viseme models for voice driven animation. In *Video/Image Processing and Multimedia Communications, 2003. 4th EURASIP Conference focused on*, volume 2, pages 649–654. IEEE, 2003.
- [179] Rivka Levitan, Stefan Benus, Ramiro H Gálvez, Agustín Gravano, Florencia Savoretti, Mar-ian Trnka, Andreas Weise, and Julia Hirschberg. Implementing acoustic-prosodic entrainment in a conversational avatar. In *INTERSPEECH*, volume 16, pages 1166–1170, 2016.

- [180] Marc Levoy. A color animation system: based on the multiplane technique. In *ACM SIGGRAPH Computer Graphics*, volume 11, pages 65–71. ACM, 1977.
- [181] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk. The digital michelangelo project. In Kurt Akeley, editor, *Proceedings of SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pages 131–144, New York, 2000. ACM, ACM Press / ACM SIGGRAPH.
- [182] B. Li, S. Mehta, D. Aneja, C. Foster, P. Ventola, F. Shic, and L. Shapiro. A facial affect analysis system for autism spectrum disorder. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4549–4553, Sep. 2019.
- [183] B. Li, S. Mehta, D. Aneja, C. Foster, P. Ventola, F. Shic, and L. Shapiro. A facial affect analysis system for autism spectrum disorder. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4549–4553, Sep. 2019.
- [184] B. Li, S. Mehta, D. Aneja, C. Foster, P. Ventola, F. Shic, and L. Shapiro. A facial affect analysis system for autism spectrum disorder. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4549–4553, Sep. 2019.
- [185] Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph.*, 32(4):42–1, 2013.
- [186] Toby Jia-Jun Li, Igor Labutov, Brad A Myers, Amos Azaria, Alexander I Rudnicky, and Tom M Mitchell. An end user development approach for failure handling in goal-oriented conversational agents. *Studies in Conversational UX Design*, 2018.
- [187] Jianhua Lin. Divergence measures based on the shannon entropy. *Information Theory, IEEE Transactions on*, 37(1):145–151, 1991.
- [188] Yuxu Lin, Mingli Song, Dao Thi Phuong Quynh, Ying He, and Chun Chen. Sparse coding for flexible, robust 3d facial-expression synthesis. *Computer Graphics and Applications, IEEE*, 32(2):76–88, 2012.
- [189] Chengjun Liu and Harry Wechsler. Independent component analysis of gabor features for face recognition. *Neural Networks, IEEE Transactions on*, 14(4):919–928, 2003.
- [190] Mengyi Liu, Shaoxin Li, Shiguang Shan, and Xilin Chen. Au-aware deep networks for facial expression recognition. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–6. IEEE, 2013.

- [191] Ping Liu, Shizhong Han, Zibo Meng, and Yan Tong. Facial expression recognition via a boosted deep belief network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1805–1812, 2014.
- [192] G. Loffler, G. Yourganov, F. Wilkinson, and H.R. Wilson. fmri evidence for the neural representation of faces. *Nature Neuroscience*, 8(10):1386–1391, 2005.
- [193] Gale Lucas, Giota Stratou, Shari Liebling, and Jonathan Gratch. Trust me: multimodal signals of trustworthiness. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 5–12. ACM, 2016.
- [194] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE, 2010.
- [195] Ewa Luger and Abigail Sellen. Like having a really bad pa: the gulf between user expectation and experience of conversational agents. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5286–5297. ACM, 2016.
- [196] D Lundqvist, A Flykt, and A Öhman. The karolinska directed emotional faces-kdef. cdrom from department of clinical neuroscience, psychology section, karolinska institutet, stockholm, sweden. Technical report, ISBN 91-630-7164-9, 1998.
- [197] Shuang Ma, Daniel McDuff, and Yale Song. A generative adversarial network for style modeling in a text-to-speech system. 2018.
- [198] Xiaohan Ma, Binh Huy Le, and Zhigang Deng. Perceptual analysis of talking avatar head movements: a quantitative perspective. In *CHI '11: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, May 2011.
- [199] Ahmed Maalej, Boulbaba Ben Amor, Mohamed Daoudi, Anuj Srivastava, and Stefano Berretti. Shape analysis of local facial patches for 3D facial expression recognition. *Pattern Recognition*, 44(8):1581–1589, August 2011.
- [200] Mohammad H Mahoor, Mu Zhou, Kevin L Veon, S Mohammad Mavadati, and Jeffrey F Cohn. Facial action unit recognition with sparse representation. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 336–342. IEEE, 2011.
- [201] Dominic W Massaro, Jonas Beskow, Michael M Cohen, Christopher L Fry, and Tony Rodriguez. Picture my voice: Audio to visual speech synthesis using artificial neural networks. In *AVSP'99-International Conference on Auditory-Visual Speech Processing*, 1999.

- [202] B. D. Mather. Making up titles for conference papers. In *Ext. Abstracts CHI 2000*, pages 1–2. ACM Press, 2000.
- [203] Yoichi Matsuyama, Arjun Bhardwaj, Ran Zhao, Oscar Romeo, Sushma Akoju, and Justine Cassell. Socially-aware animated intelligent personal assistant agent. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 224–227, 2016.
- [204] Iain Matthews, Natasha Kholgade, and Yaser Sheikh. Content retargeting using facial layers, January 26 2016. US Patent 9,245,176.
- [205] Iain Matthews, Jing Xiao, and Simon Baker. 2d vs. 3d deformable face models: Representational power, construction, and real-time fitting. *International journal of computer vision*, 75(1):93–113, 2007.
- [206] Wesley Mattheyses and Werner Verhelst. Audiovisual speech synthesis: An overview of the state-of-the-art. *Speech Communication*, 66:182–217, 2015.
- [207] S Mohammad Mavadati, Mohammad H Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F Cohn. Disfa: A spontaneous facial action intensity database. *Affective Computing, IEEE Transactions on*, 4(2):151–160, 2013.
- [208] Rachel McDonnell, Martin Breidt, and Heinrich H Bülthoff. Render me real?: investigating the effect of render style on the perception of animated virtual humans. *ACM Transactions on Graphics (TOG)*, 31(4):91, 2012.
- [209] Daniel McDuff, Abdelrahman Mahmoud, Mohammad Mavadati, May Amr, Jay Turcot, and Rana el Kaliouby. Affdex sdk: a cross-platform real-time multi-face expression recognition toolkit. In *Proceedings of the 2016 CHI conference extended abstracts on human factors in computing systems*, pages 3723–3726. ACM, 2016.
- [210] Robert McGregor. Apple animoji demo for iphonex. *AppleEvent September 12*, 2017.
- [211] Adam S Miner, Arnold Milstein, Stephen Schueller, Roshini Hegde, Christina Mangurian, and Eleni Linos. Smartphone-based conversational agents and responses to questions about mental health, interpersonal violence, and physical health. *JAMA internal medicine*, 176(5):619–625, 2016.
- [212] Nicole Mirnig, Gerald Stollnberger, Markus Miksch, Susanne Stadler, Manuel Giuliani, and Manfred Tscheligi. To err is robot: How humans assess and act toward an erroneous social robot. *Frontiers in Robotics and AI*, 4:21, 2017.

- [213] Mixamo. Face plus. <https://www.mixamo.com/faceplus/>.
- [214] Ali Mollahosseini, David Chan, and Mohammad H Mahoor. Going deeper in facial expression recognition using deep neural networks. *arXiv preprint arXiv:1511.04110*, 2015.
- [215] Ali Mollahosseini, David Chan, and Mohammad H Mahoor. Going deeper in facial expression recognition using deep neural networks. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–10. IEEE, 2016.
- [216] Ali Mollahosseini, Behzad Hasani, and Mohammad H Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 2017.
- [217] Ali Mollahosseini, Behzad Hasani, Michelle J Salvador, Hojjat Abdollahi, David Chan, and Mohammad H Mahoor. Facial expression recognition from world wild web. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 58–65, 2016.
- [218] Robert R Morris, Kareem Kouddous, Rohan Kshirsagar, and Stephen M Schueller. Towards an artificially empathic conversational agent for mental health applications: System design and user perceptions. *Journal of medical Internet research*, 20(6), 2018.
- [219] Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios P Spithourakis, and Lucy Vanderwende. Image-grounded conversations: Multimodal context for natural question and response generation. *arXiv preprint arXiv:1701.08251*, 2017.
- [220] Barzan Mozafari, Purnamrita Sarkar, Michael J. Franklin, Michael I. Jordan, and Samuel Madden. Active learning for crowd-sourced databases. *arXiv:1209.3686 [cs]*, September 2012.
- [221] I. Mpiparis, S. Malassiotis, and M.G. Strintzis. Bilinear models for 3-d face and facial expression recognition. *IEEE Transactions on Information Forensics and Security*, 3(3):498–511, September 2008.
- [222] Henning Müller, Stéphane Marchand-Maillet, and Thierry Pun. The truth about core-evaluation in image retrieval. In *Image and Video Retrieval*, pages 38–49. Springer, 2002.
- [223] Kate G Niederhoffer and James W Pennebaker. Linguistic style matching in social interaction. *Journal of Language and Social Psychology*, 21(4):337–360, 2002.
- [224] Nintendo R&D1 and Intelligent Systems. *Super Metroid*. Game [SNES], April 1994. Nintendo, Kyoto, Japan. Played August 2011.

- [225] Jun-yong Noh and Ulrich Neumann. Expression cloning. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 277–288. ACM, 2001.
- [226] Jun-yong Noh and Ulrich Neumann. Expression cloning. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01*, page 277–288, New York, NY, USA, 2001. ACM.
- [227] Tetsuo Ono, Michita Imai, and Hiroshi Ishiguro. A model of embodied communications with gestures between human and robots. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 23, 2001.
- [228] Open Broadcaster Software, 2017.
- [229] OptiTrack. Expression. <http://optitrack.com/products/expression/>.
- [230] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1717–1724, 2014.
- [231] J. Ostermann. Animation of synthetic faces in MPEG-4. In *Computer Animation 98. Proceedings*, pages 49–55, June 1998.
- [232] Jahna Otterbacher, Chee Siang Ang, Marina Litvak, and David Atkins. Show me you care: Trait empathy, linguistic style, and mimicry on facebook. *ACM Transactions on Internet Technology (TOIT)*, 17(1):6, 2017.
- [233] Todd Palamar. *Mastering Autodesk Maya 2016: Autodesk Official Press*. John Wiley & Sons, 2015.
- [234] Maja Pantic, Michel Valstar, Ron Rademaker, and Ludo Maat. Web-based database for facial expression analysis. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 5–pp. IEEE, 2005.
- [235] Gabriele Paolacci, Jesse Chandler, and Panagiotis G. Ipeirotis. Running experiments on amazon mechanical turk. SSRN Scholarly Paper ID 1626226, Social Science Research Network, Rochester, NY, June 2010.
- [236] Devi Parikh and Kristen Grauman. Relative attributes. In *Computer Vision, IEEE International Conference on*, volume 0, pages 503–510, Los Alamitos, CA, USA, 2011. IEEE Computer Society.

- [237] S. W. Park, L. Linsen, O. Kreylos, J. D. Owens, and B. Hamann. Discrete sibson interpolation. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):243–253, March/April 2006.
- [238] Frederic I. Parke and Keith Waters. *Computer Facial Animation*. A. K. Peters, 1996.
- [239] Stefano Pasquariello and Catherine Pelachaud. Greta: A simple facial animation engine. In *Soft computing and industry*, pages 511–525. Springer, 2002.
- [240] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- [241] S.R. Peck. Atlas of facial expression: An account of facial expression for artists, actors, and writers.
- [242] S.R. Peck. *Atlas of Human Anatomy for the Artist*. Oxford University Press, 1951.
- [243] Florian Pecune, Jingya Chen, Yoichi Matsuyama, and Justine Cassell. Field trial analysis of socially aware robot assistant. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1241–1249. International Foundation for Autonomous Agents and Multiagent Systems, 2018.
- [244] Catherine Pelachaud. Modelling multimodal expression of emotion in a virtual agent. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3539–3548, December 2009. PMID: 19884148 PMCID: PMC2781894.
- [245] Catherine Pelachaud and Isabella Poggi. Subtleties of facial expressions in embodied agents. *The Journal of Visualization and Computer Animation*, 13(5):301–312, 2002.
- [246] Fabio Pellacini, Kiril Vidimče, Aaron Lefohn, Alex Mohr, Mark Leone, and John Warren. Lpics: a hybrid hardware-accelerated relighting engine for computer cinematography. *ACM Transactions on Graphics*, 24(3):464–470, August 2005.
- [247] Fernando C. Pereira and Touradj Ebrahimi. *The MPEG-4 Book*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2002.
- [248] Pierre Philip, Jean-Arthur Micoulaud-Franchi, Patricia Sagaspe, Etienne De Sevin, Jérôme Olive, Stéphanie Bioulac, and Alain Sauteraud. Virtual human as a new diagnostic tool, a proof of concept study in the field of major depressive disorders. *Scientific Reports*, 7:42656, 2017.

- [249] Isabella Poggi, Catherine Pelachaud, Fiorella de Rosis, Valeria Carofiglio, and Berardina De Carolis. Greta. a believable embodied conversational agent. In *Multimodal intelligent information presentation*, pages 3–25. Springer, 2005.
- [250] Tom Porter. Creating lifelike characters in toy story. *ACM SIGART Bulletin*, 8(1-4):10–14, 1997.
- [251] Tom Porter and Galyn Susman. On site: creating lifelike characters in pixar movies. *Communications of the ACM*, 43(1):25, 2000.
- [252] Hyewon Pyun, Yejin Kim, Wonseok Chae, Hyung Woo Kang, and Sung Yong Shin. An example-based approach for facial expression cloning. In *ACM SIGGRAPH 2006 Courses, SIGGRAPH '06*, New York, NY, USA, 2006. ACM.
- [253] S. Ramanathan, A. Kassim, Y.V. Venkatesh, and W.S. Wah. Human facial expression recognition using a 3D morphable model. In *2006 IEEE International Conference on Image Processing*, pages 661–664, 2006.
- [254] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.
- [255] William T Reeves. Inbetweening for computer animation utilizing moving point constraints. *ACM SIGGRAPH Computer Graphics*, 15(3):263–269, 1981.
- [256] Laurel D Riek and Peter Robinson. Real-time empathy: Facial mimicry on a robot. In *Workshop on Affective Interaction in Natural Environments (AFFINE) at the International ACM Conference on Multimodal Interfaces (ICMI 08)*. ACM. Citeseer, 2008.
- [257] Noodle rig, 2019.
- [258] Etienne B. Roesch, Lucas Tamarit, Lionel Reveret, Didier Grandjean, David Sander, and Klaus R. Scherer. FACSGen: a tool to synthesize emotional facial expressions through systematic manipulation of facial action units. *Journal of Nonverbal Behavior*, 35(1):1–16, March 2011.
- [259] William T Rogers. The contribution of kinesic illustrators toward the comprehension of verbal behavior within utterances. *Human communication research*, 5(1):54–62, 1978.
- [260] Zsófia Ruttkay and Catherine Pelachaud. *From brows to trust: evaluating embodied conversational agents*, volume 7. Springer Science & Business Media, 2006.

- [261] Yusaku Sako and Kikuo Fujimura. Shape similarity by homotropic deformation. *The Visual Computer*, 16(1):47–61, 2000.
- [262] Maha Salem, Friederike Eyssel, Katharina Rohlfing, Stefan Kopp, and Frank Joublin. To err is human (-like): Effects of robot gesture on perceived anthropomorphism and likability. *International Journal of Social Robotics*, 5(3):313–323, 2013.
- [263] Georgia Sandbach, Stefanos Zafeiriou, Maja Pantic, and Lijun Yin. Static and dynamic 3D facial expression recognition: A comprehensive survey. *Image and Vision Computing*, 30(10):683–697, 2012.
- [264] Marcos Santos-Pérez, Eva González-Parada, and José Manuel Cano-García. Avatar: An open source architecture for embodied conversational agents in smart environments. In *International Workshop on Ambient Assisted Living*, pages 109–115. Springer, 2011.
- [265] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Real-time avatar animation from a single image. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 117–124. IEEE, 2011.
- [266] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.
- [267] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.
- [268] Marilyn Schwartz. *Guidelines for Bias-Free Writing*. ERIC, Bloomington, IN, USA, 1995.
- [269] Lauren E Scissors, Alastair J Gill, Kathleen Geraghty, and Darren Gergle. In cmc we trust: The role of similarity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 527–536. ACM, 2009.
- [270] Lauren E Scissors, Alastair J Gill, and Darren Gergle. Linguistic mimicry and trust in text-based cmc. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 277–280. ACM, 2008.
- [271] Adrian Secord, Jingwan Lu, Adam Finkelstein, Manish Singh, and Andrew Nealen. Perceptual models of viewpoint preference. *ACM Transactions on Graphics*, 30(5), October 2011.
- [272] Yoshihiro Sejima, Yutaka Ishii, and Tomio Watanabe. A virtual audience system for enhancing embodied interaction based on conversational activity. In *Symposium on Human Interface*, pages 180–189. Springer, 2011.

- [273] Yeongho Seol, Jaewoo Seo, Paul Hyunjin Kim, JP Lewis, and Junyong Noh. Artist friendly facial animation retargeting. In *ACM Transactions on Graphics (TOG)*, volume 30, page 162. ACM, 2011.
- [274] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [275] Burr Settles. Active learning literature survey. Technical report, 2010.
- [276] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer, 2018.
- [277] Ameneh Shamekhi, Mary Czerwinski, Gloria Mark, Margeigh Novotny, and Gregory A Bennett. An exploratory study toward the preferred conversational style for compatible virtual agents. In *International Conference on Intelligent Virtual Agents*, pages 40–50. Springer, 2016.
- [278] Caifeng Shan, Shaogang Gong, and Peter W McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009.
- [279] Aaron Shon, Keith Grochow, Aaron Hertzmann, and Rajesh P Rao. Learning shared latent structure for image synthesis and robotic imitation. In *Advances in neural information processing systems*, pages 1233–1240, 2006.
- [280] Chang Shu, Xiaoqing Ding, and Chi Fang. Histogram of the oriented gradient for face recognition. *Tsinghua Science & Technology*, 16(2):216–224, 2011.
- [281] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron J Weiss, Rob Clark, and Rif A Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. *arXiv preprint arXiv:1803.09047*, 2018.
- [282] Leon Li-Aun Sooi and Xiong Lin. Mathilda rig. <http://www.cgmeetup.net/home/mathilda-rig-free-maya-rig-female-character-rig/>.
- [283] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. In *Proc. of NAACL-HLT*, 2015.

- [284] Hamit Soyel and Hasan Demirel. Facial expression recognition using 3D facial feature distances. In Mohamed Kamel and Aurelio Campilho, editors, *Image Analysis and Recognition*, number 4633 in Lecture Notes in Computer Science, pages 831–838. Springer Berlin Heidelberg, January 2007.
- [285] C Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 1904.
- [286] Jesse Spencer-smith, Heather Wild, Å. . . se H. Innes-ker, James Townsend, Christy Duffy, Chad Edwards, Kristina Ervin, Nicole Merritt, and Jae Won Paik. Making faces: Creating three-dimensional parameterized models of facial expression. *Behavior Research Methods, Instruments & Computers*, 33:115–123, 2001.
- [287] W. Stanchfield. *Drawn to Life: 20 Golden Years of Disney Master Classes: Volume 1 the Walt Stanchfield Lectures*. Focal Press, 2009.
- [288] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):95, 2017.
- [289] Noriko Suzuki and Christoph Bartneck. Subtle expressivity for characters and robots. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, page 1064–1065, New York, NY, USA, 2003. ACM.
- [290] Xiaoyang Tan and Bill Triggs. Fusing gabor and lbp feature sets for kernel-based face recognition. In *Analysis and Modeling of Faces and Gestures*, pages 235–249. Springer, 2007.
- [291] Hiroki Tanaka, Hiroyoshi Adachi, Norimichi Ukita, Manabu Ikeda, Hiroaki Kazui, Takashi Kudo, and Satoshi Nakamura. Detecting dementia through interactive computer avatars. *IEEE journal of translational engineering in health and medicine*, 5:1–11, 2017.
- [292] Hiroki Tanaka, Hideki Negoro, Hidemi Iwasaka, and Satoshi Nakamura. Embodied conversational agents for multimodal automated social skills training in people with autism spectrum disorders. *PloS one*, 12(8):e0182151, 2017.
- [293] Hao Tang and T.S. Huang. 3D facial expression recognition based on properties of line segments connecting facial feature points. In *8th IEEE International Conference on Automatic Face Gesture Recognition, 2008. FG '08*, pages 1–6, September 2008.
- [294] Deborah Tannen. Conversational style. In Hans W Dechert and Manfred Raupach, editors, *Psycholinguistic models of production*. Ablex, Norwood, NJ, 1987.

- [295] Deborah Tannen. *Conversational style: Analyzing talk among friends*. Oxford University Press, New York, new edition, 2005.
- [296] Sarah Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. A deep learning approach for generalized speech animation. *ACM Transactions on Graphics (TOG)*, 36(4):93, 2017.
- [297] Sarah L Taylor, Moshe Mahler, Barry-John Theobald, and Iain Matthews. Dynamic units of visual speech. In *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation*, pages 275–284. Eurographics Association, 2012.
- [298] D. Terzopoulos, B. Mones-Hattal, B. Hofer, F. Parke, D. Sweetland, and K. Waters. Facial animation (panel): Past, present and future. *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, August 1997.
- [299] F. Thomas and O. Johnson. *Disney animation the illusion of life*. 1984.
- [300] Frank Thomas. *Disney Animation. The Illusion of Life*. August 1987.
- [301] Frank Thomas, Ollie Johnston, and Frank. Thomas. *The illusion of life: Disney animation*. Hyperion New York, 1995.
- [302] Paul Thomas, Mary Czerwinski, Daniel McDuff, Nick Craswell, and Gloria Mark. Style and alignment in information-seeking conversation. In *Proceedings of the 2018 Conference on Human Information Interaction&Retrieval*, pages 42–51. ACM, 2018.
- [303] Paul Thomas, Daniel McDuff, Mary Czerwinski, and Nick Craswell. Misc: A data set of information-seeking conversations. In *SIGIR 1st International Workshop on Conversational Approaches to Information Retrieval (CAIR'17)*, volume 5, 2017.
- [304] Ying-li Tian, Takeo Kanade, and Jeffrey F Cohn. Evaluation of gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 229–234. IEEE, 2002.
- [305] A. Tinwell, M. Grimshaw, D.A. Nabi, and A. Williams. Facial expression of emotion and perception of the uncanny valley in virtual characters. *Computers in Human Behavior*, 27(2):741–749, March 2011.
- [306] Michalis Titsias and Neil D Lawrence. Bayesian gaussian process latent variable model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 844–851, 2010.

- [307] ToonBoom, 2017.
- [308] David Traum and Jeff Rickel. Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, pages 766–773. ACM, 2002.
- [309] Stef van der Struijk, Hung-Hsuan Huang, Maryam Sadat Mirzaei, and Toyoaki Nishida. Facsvatar: An open source modular framework for real-time face based facial animation. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pages 159–164. ACM, 2018.
- [310] Anu Venkatesh, Chandra Khatri, Ashwin Ram, Fenfei Guo, Raefer Gabriel, Ashish Nagar, Rohit Prasad, Ming Cheng, Behnam Hedayatnia, Angeliki Metallinou, et al. On evaluating and comparing conversational agents. *arXiv preprint arXiv:1801.03625*, 4:60–68, 2018.
- [311] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 426–433. ACM, 2005.
- [312] P. Vuilleumier and G. Pourtois. Distributed and interactive brain mechanisms during emotion face perception: evidence from functional neuroimaging. *Neuropsychologia*, 45(1):174–194, January 2007.
- [313] Bjorn Vuylsteker. Speech recognition — A comparison of popular services in en and nl. <http://bit.ly/2BmZ3sn>, 2017. Published: 2017-02-12.
- [314] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. Sphinx-4: A flexible open source framework for speech recognition. 2004.
- [315] C. Wallraven, M. Breidt, D.W. Cunningham, and H.H. Bühlhoff. Evaluating the perceptual realism of animated facial expressions. *ACM Transactions on Applied Perception*, 4(4), 2008.
- [316] C. Wallraven, J. Fischer, D.W. Cunningham, D. Bartz, and H.H. Bühlhoff. The evaluation of stylized facial expressions. *APGV '06 Proceedings of the 3rd symposium on Applied perception in graphics and visualization*, pages 85–92, 2006.
- [317] Jun Wang, Lijun Yin, Xiaozhou Wei, and Yi Sun. 3D facial expression recognition based on primitive surface feature distribution. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1399–1406, 2006.

- [318] Zhikun Wang, Katharina Mülling, Marc Peter Deisenroth, Heni Ben Amor, David Vogt, Bernhard Schölkopf, and Jan Peters. Probabilistic movement modeling for intention inference in human–robot interaction. *The International Journal of Robotics Research*, 32(7):841–858, 2013.
- [319] David Watson, Lee A Clark, and Auke Tellegen. Development and validation of brief measures of positive and negative affect: the panas scales. *Journal of personality and social psychology*, 54(6):1063, 1988.
- [320] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. In *ACM Transactions on Graphics (TOG)*, volume 30, page 77. ACM, 2011.
- [321] Jacob Whitehill, Gwen Littlewort, Ian Fasel, Marian Bartlett, and Javier Movellan. Toward practical smile detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(11):2106–2111, 2009.
- [322] R. Williams. *The Animator’s Survival Kit*. Faber & Faber, 2012.
- [323] Wayne Xiong, Lingfeng Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke. The microsoft 2017 conversational speech recognition system. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5934–5938. IEEE, 2018.
- [324] Xuehan Xiong and Fernando Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 532–539, 2013.
- [325] Yuyu Xu, Andrew W Feng, Stacy Marsella, and Ari Shapiro. A practical and configurable lip sync method for games. In *Proceedings of Motion on Games*, pages 131–140. ACM, 2013.
- [326] Katsu Yamane, Yuka Ariki, and Jessica Hodgins. Animating non-humanoid characters with human motion data. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 169–178. Eurographics Association, 2010.
- [327] Yang Li Hector Yee. Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. Master’s thesis, Cornell University, 2000.
- [328] Ugur Ulvi Yetiskin. Tuna rig. <https://www.behance.net/gallery/31141085/Tuna-Rig-for-FREE>.

- [329] Zi-Lu Ying, Zhe-Wei Wang, and Ming-Wei Huang. Facial expression recognition based on fusion of sparse representation. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, pages 457–464. Springer, 2010.
- [330] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.
- [331] YouTube, 2017.
- [332] Dong Yu and Li Deng. *Automatic speech recognition: A deep learning approach*. Springer, 2014.
- [333] Xiang Yu, Jianchao Yang, Linjie Luo, Wilmot Li, Jonathan Brandt, and Dimitris Metaxas. Customized expression recognition for performance-driven cutout character animation. In *Winter Conference on Computer Vision*, 2016.
- [334] Zhiding Yu and Cha Zhang. Image based static facial expression recognition with multiple deep network learning. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 435–442. ACM, 2015.
- [335] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2015.
- [336] Thiago HH Zavaschi, Alceu S Britto, Luiz ES Oliveira, and Alessandro L Koerich. Fusion of feature sets and classifiers for facial expression recognition. *Expert Systems with Applications*, 40(2):646–655, 2013.
- [337] Polle T. Zellweger, Niels Olof Bouvin, Henning Jehøj, and Jock D. Mackinlay. Fluid annotations in an open world. In *Proc. Hypertext 2001*, pages 9–18. ACM Press, 2001.
- [338] Robert Zemeckis. *The polar express*. 2005.
- [339] Zhengyou Zhang, Michael Lyons, Michael Schuster, and Shigeru Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 454–459. IEEE, 1998.
- [340] Guoying Zhao and Matti Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):915–928, 2007.

- [341] Lin Zhong, Qingshan Liu, Peng Yang, Bo Liu, Junzhou Huang, and Dimitris N Metaxas. Learning active facial patches for expression analysis. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2562–2569. IEEE, 2012.
- [342] Yang Zhou, Shan Xu, Chris Landreth, Evangelos Kalogerakis, Subhransu Maji, and Karan Singh. Visemenet: Audio-driven animator-centric speech animation. *arXiv preprint arXiv:1805.09488*, 2018.