

Facial Animation Pipeline Optimisation with Blendshape Importance Ordering and Perceptual Metrics

by

Emma Violet Carrigan, B.A. Mod.

Dissertation

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

for the Degree of

Doctor of Philosophy

University of Dublin, Trinity College

April 2020

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Emma Violet Carrigan

October 21, 2020

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Emma Violet Carrigan

October 21, 2020

Acknowledgments

I would like to thank my supervisor, Rachel McDonnell, for the guidance throughout this PhD, the long hours committed, and for the regular coffee meetings which made the last few years much more manageable and enjoyable.

I would also like to give a special thanks to Katja Žibrek who was my confidante both in work and out, and with whom I was able to work through any problem. (And there were many.)

To my friends in the office (there are too many to mention, apologies), thank you for being available to chat, vent, drink, and distract when I needed. Hopefully I provided some of the same support!

I must also mention my best friends of almost 9 years at this point. Anne, Austin, Darragh, Glenn, Kevin, Killian, and Sam (listed alphabetically, feel free to argue over how they would be ordered by importance), I cannot begin to express how important you all are to me and how thankful I am that we somehow ended up together. Sorry for how much you have had to listen to me complain. There will definitely be more in the future. Look forward to it.

Last but not least, I want to thank my family. To my mother and brother who have given me immeasurable support and guidance throughout my life, and continue to do so, and to my father, who unfortunately could not see me submit this thesis, but whose memory reminds me that the most important thing I can do is live happily.

EMMA VIOLET CARRIGAN

University of Dublin, Trinity College

April 2020

Facial Animation Pipeline Optimisation with Blendshape Importance Ordering and Perceptual Metrics

Publication No. _____

Emma Violet Carrigan, Ph.D.
University of Dublin, Trinity College, 2020

Supervisor: Rachel McDonnell

The quality of realistic virtual human faces has increased in recent years with the availability of new technologies such as photogrammetry systems to scan actors and reproduce their shape and texture digitally with minute detail, improved motion capture to allow for the recording of intricate facial movements, and new game engines which allow for the real-time rendering of complex structures and materials such as the human eye. All of these combined have led to an intensely high level of quality in virtual characters, but the added complexity of these technologies has increased the amount of time, money, and effort needed to create them, with a significant increase

in the computational power and memory required due to the high level of detail.

The goal of this thesis is to review the facial animation pipeline, identify areas to optimise, and create new methods and algorithms to assist in the creation of high-quality, realistic facial animation. The current industry standard for the high quality facial animation is blendshape animation. This method requires a large amount of geometry to describe the different possible poses of the face, known as blendshapes.

We first optimise the rig creation stage, the stage in which blendshapes are created either by scanning an actor, or through a process called blendshape transfer, which transfers existing blendshape expressions from a template character to create the required shapes. The blendshapes created through blendshape transfer can be further improved by providing training example expressions of the target character. We conduct a perceptual experiment on the impact of this training on blendshape transfer, and then propose a novel algorithm to suggest training examples using blendshape importance ordering, i.e. the ordering of blendshapes such that the most and least important blendshapes can be easily identified for a given task, as an integral part of the example creation process, and an unexplored area that could be applied to various stages of the pipeline and benefit from further investigation.

We go on to apply this idea of blendshape ordering to the animation stage. We implement a GPU animation method with level of detail optimisation through blendshape reduction, using the idea of blendshape importance to identify which blendshapes are least important and can be removed.

We conclude with a perceptual experiment to further investigate the numeric metrics used throughout this thesis and their relation to human perception of faces. These results provide a perceptual basis for blendshape ordering which can be used to optimise the facial animation pipeline in methods such as blendshape reduction for level of detail optimisation and training example creation using blendshape importance ordering, such as those which we explore in this thesis.

Contents

| | |
|------------------------------------|-------------|
| Acknowledgments | iv |
| Abstract | v |
| List of Tables | x |
| List of Figures | xiii |
| Chapter 1 Introduction | 1 |
| 1.1 Methodology | 5 |
| 1.1.1 Geometric Metrics | 5 |
| 1.1.2 Image Metrics | 7 |
| 1.1.3 Subjective Metrics | 7 |
| 1.1.4 Other | 8 |
| 1.2 Motivation | 8 |
| 1.3 Scope | 9 |
| 1.4 Limitations | 10 |
| 1.5 Contributions | 11 |
| 1.6 Summary of Chapters | 12 |

| | | |
|------------------|--|-----------|
| Chapter 2 | Related Work | 14 |
| 2.1 | Facial Animation Pipeline | 14 |
| 2.1.1 | Character Creation | 16 |
| 2.1.2 | Rigging | 18 |
| 2.1.3 | Animation | 23 |
| 2.2 | The Human Face | 30 |
| 2.2.1 | Recognition | 32 |
| 2.2.2 | Attention | 33 |
| 2.2.3 | Emotion | 34 |
| 2.2.4 | Perception | 37 |
| 2.2.5 | Perception in Graphics | 38 |
| 2.3 | Conclusion | 40 |
| Chapter 3 | Geometry Optimisation | 42 |
| 3.1 | Perception of Example-Based Facial Blendshape Creation | 44 |
| 3.1.1 | Method | 45 |
| 3.1.2 | Results | 49 |
| 3.1.3 | Discussion | 53 |
| 3.2 | Expression Packing | 55 |
| 3.2.1 | Blendshape Transfer | 57 |
| 3.2.2 | Optimal Reference Expressions | 58 |
| 3.2.3 | Results and Evaluation | 73 |
| 3.2.4 | Discussion | 79 |
| Chapter 4 | Real-Time Optimisation | 90 |
| 4.1 | GPU Blendshape Animation | 92 |

| | | |
|--|--|------------|
| 4.1.1 | Method | 94 |
| 4.1.2 | Level of detail | 98 |
| 4.1.3 | Test Data | 101 |
| 4.1.4 | Results | 102 |
| 4.1.5 | Discussion | 104 |
| Chapter 5 Perceptual Optimisation | | 107 |
| 5.1 | Perceptual Importance of Blendshapes | 109 |
| 5.1.1 | Preliminary Experiment | 112 |
| 5.1.2 | Main Experiment | 115 |
| 5.1.3 | Perceptual Results | 118 |
| 5.1.4 | Error Results | 124 |
| 5.1.5 | Model Fit | 129 |
| 5.1.6 | Discussion | 140 |
| Chapter 6 Discussion | | 146 |
| 6.1 | Research Contributions | 147 |
| 6.2 | Methodology Contributions | 149 |
| 6.3 | Material Contributions | 150 |
| 6.4 | Guidelines for Character Design | 150 |
| Chapter 7 Future Work | | 152 |
| Bibliography | | 157 |

List of Tables

| | | |
|-----|--|-----|
| 3.1 | The Action Units we used in our experiment with their FACS names. The third column shows the numbers we used for them in our experiment. Expressions 0-15 are lower face expressions, 16-18 are upper face. | 47 |
| 3.2 | Results grouped by ratio of trained to untrained responses. | 52 |
| 3.3 | Notation overview of the most relevant variables. | 59 |
| 3.4 | Timings for computing the first reference expression. Computation time depends on the vertex number N and the number of blendshapes K in the blendshape rig. | 71 |
| 3.5 | Number of blendshapes selected by greedy and optimal algorithms and their overlap in the first 5, 10 and 20 training expressions. Format: Greedy/Optimal (Overlap). | 71 |
| 5.1 | Action Units shown in the experiment. Note: AU 27 (shown in italics alongside AU 26) was shown in the preliminary experiment, but was changed to AU 26 for the main experiment to better match the magnitude of the other AUs. | 113 |

| | | |
|-----|--|-----|
| 5.2 | ANOVA interactions with dependent variable “Perceptual Difference” from the perceptual results. (BS = Blendshape, * represents significant p-values, F* stand for Greenhouse-Geisser correction for violations of sphericity). Effects sizes are reported in the last column (η_p^2). | 119 |
| 5.3 | Our blendshapes ordered from most to least important by each of the different error methods and the perceptual results, averaged across all characters and activation levels. | 130 |
| 5.4 | Pearson correlations between the error metrics we calculated and the perceptual results. Includes Neutrals, average Similarity result across participants. | 130 |
| 5.5 | Pearson correlations between our error metrics and the perceptual results broken down by Blendshape, Sex, and Race; $p = 0.00$ for all RMS and Triangle Difference (Tri. Diff.), $p < 0.0003$ for Hausdorff, $p < 0.02$ for MSE, $p < 0.001$ for SSIM. * denotes insignificant results. Similarity results averaged across participants. | 131 |
| 5.6 | ANOVA test Response Time of participants with relation to Perceptual Difference and blendshape. Only Perceived Difference is a significant regressor for Response Time. (Difference = PerceptualDifference, BS = Blendshapes) | 133 |
| 5.7 | A linear model is fitted to predict Response Time with relation to Perceptual Difference and it shows a negative correlation between Perceptual Difference and Response Time of participants. (Difference = PerceptualDifference, BS = Blendshapes) | 133 |

| | | |
|------|---|-----|
| 5.8 | Model comparison with AIC_{\downarrow} to explain the perceived difference. Best link function reported for Poisson between Identity(Id), log and sqrt. Only Id link function has been tested with Gaussian distribution. . . . | 134 |
| 5.9 | Perceptual difference with relation to RMS and interaction with character Sex, character Race and Blendshapes. The red rows mark insignificant results. | 135 |
| 5.10 | Perceptual difference with relation to SSIM and interaction with character Sex, character Race and Blendshapes. | 136 |
| 5.11 | Coefficients for model in Figure 5.14 (b) using the Neutral expression as reference. Red values indicate models with a bad fit. $\Pr(> z)$ indicates the p-value. | 139 |
| 5.12 | The blendshapes ordered by average perceptual difference. | 143 |
| 5.13 | Sum of the absolute values of each component of each vertex difference of each blendshape from the neutral. | 145 |
| 5.14 | Correlations between the X, Y and Z components defined in Table 5.13 and the similarity results. | 145 |

List of Figures

| | | |
|-----|---|----|
| 1.1 | <i>Hellblade: Senua's Sacrifice</i> (2017) | 2 |
| 1.2 | <i>Detroit: Become Human</i> (2018) | 3 |
| 2.1 | The facial animation pipeline. The stages discussed in detail in this thesis are highlighted in green. | 15 |
| 2.2 | An actor being scanned in a light stage. | 17 |
| 2.3 | A simple bipedal skeleton. | 19 |
| 2.4 | A virtual human with the bones of his facial rig shown on the surface of his face (left) and the complex internal structure (right). | 19 |
| 2.5 | The virtual human Louise created by Eisko with an open mouth blendshape at 0.6 activation. | 20 |
| 2.6 | The musculature of the face, from <i>Gray's Anatomy E-Book: The Anatomical Basis of Clinical Practice</i> [Sta15] | 31 |
| 2.7 | The basic emotions as defined by Ekman [Ekm92b], including the later added <i>contempt</i> emotion, from <i>paulekman.com</i> [Ekm20] | 35 |
| 3.1 | The facial animation pipeline with the rig creation stage highlighted. | 42 |
| 3.2 | An example of stimuli shown to participants comparing the facial rigs created without training, and those with training. | 45 |

| | | |
|-----|---|----|
| 3.3 | Examples of scanned poses from the Bosphorus database. | 46 |
| 3.4 | Main effect of Expression on preference of EBFR over DT. | 50 |
| 3.5 | (a) The texture artifact which affected expressions 16 and 18. (b) The artifact which affected expression 13. | 51 |
| 3.6 | Deformation Transfer [SP04] showing the output with (A) only the non-translational transformations, (B) all transformations, and (C) Deformation Transfer, which solves the optimisation problem balancing the non-translational transformations with new translations such that vertex connectivity is maintained. | 81 |
| 3.7 | Illustration of our expression packing algorithm as part of the digital double pipeline. We run our expression packing algorithm to create as-few-as-possible reference expressions, which help to guide an actor during a scanning session. Using non-rigid registration, e.g. [IBP15, FNH ⁺ 17, LBB ⁺ 17], the mesh topology of the template rig is transferred to the 3D scans. The resulting training expressions are optimal for the blendshape transfer algorithm, since they lead to best results with as-few-as-possible expressions. Finally, the blendshapes are computed for the personalised rig. | 81 |
| 3.8 | Illustration of the different types of vertices for one blendshape: static (blue), smoothing (green) and active (red). The blendshape on the left is self-symmetric, the two blendshapes on the right are symmetric, but have overlapping active vertices. | 82 |
| 3.9 | First five expressions computed with ($\beta = 0.5$) and without ($\beta = 0.0$) the optional blendshape uniqueness metric. | 82 |

| | | |
|------|---|----|
| 3.10 | Visualisation of Displacement Strength (left) and Triangle Distortion (right). | 82 |
| 3.11 | Symmetric blendshape activation. Shape symmetry remains while lifting both eyebrows (left) and motion symmetry is created due to perpendicular movement of the lips (right). | 83 |
| 3.12 | Comparison of individual blendshapes after applying blendshape transfer without training expressions (Deformation Transfer) and with only four training expressions based on our reference expressions. Each row shows two blendshapes that have been corrected by one training expression. | 84 |
| 3.13 | Output from expression packing algorithm for our Female test-case: the first four computed reference expressions using different thresholds for the definition of binary blendshapes. A threshold of up to $\mu = 0.3$ creates plausible results in most cases even for model with many blendshapes ($K = 265$). | 85 |
| 3.14 | Two volunteers demonstrating the pose-ability of the DigiDouble expressions. The number of blendshapes in each expression is shown on the left and right columns. | 86 |
| 3.15 | Average RMS-error per vertex between ground truth and the output of example-based blendshape transfer for 3 different test-cases. Number of training expressions ranges from 2 to 15, (shown on x-axis). Importance weighting that considers both, vertex displacement (λ_{Dis}) and triangle distortions (λ_{TD}), creates most accurate results across different characters and number of training expressions. | 87 |

| | | |
|------|--|----|
| 3.16 | First packed reference expression (Figure 3.18) together with the nine individual blendshapes. | 87 |
| 3.17 | RMS-error for our DigiDouble test case ($k = 161$) with different numbers of training expressions as input (ranging from 2 to 25, shown on x-axis). Training expressions were computed with the optimal and greedy algorithms. | 87 |
| 3.18 | Output from our expression packing algorithm showing the first 3 computed reference expressions. The number below each is the number of blendshapes that have been packed into that reference expression. Expressions were computed (left) with no importance weighting or symmetry constraints, (middle) with importance weights only, and (right) with importance- weights and symmetry constraints. | 88 |
| 3.19 | Selection of individual blendshapes after applying blendshape transfer in combination with a full set of training expressions, where every blendshape is present in one of the training expressions. | 89 |
| 4.1 | The facial animation pipeline with the real-time animation stage highlighted. | 90 |
| 4.2 | A real-time rendering of 40 blendshape animated heads with 72 blendshapes each using our GPU technique. | 93 |
| 4.3 | The blendshapes are serialised and stacked together in the 1-dimensional Texture Buffer Object. | 94 |

| | | |
|-----|---|-----|
| 4.4 | The blendshape animation encoded in a 32 bit floating-point texture. Each row contains the animated weight of a specific blendshape over time. We can see a group of blendshapes activate in the middle (bright white). | 95 |
| 4.5 | Combining expression blendshapes to make a final, composite expression. | 101 |
| 4.6 | Plots compare the basic CPU method, the vertex shader method, and the compute shader method. The top row gives GPU times for rasterisation-disabled drawing for 1 and 100 heads, each head with unique blendshape weights. The bottom row gives corresponding times on the CPU. | 102 |
| 4.7 | The special case of instanced rendering for crowds (sharing weights) on the GPU (left), and round-trip frame time on the CPU (right). We observe performance advantage in leveraging generic computing in shaders here. | 104 |
| 5.1 | The set of blendshapes used in our main experiment, shown on the Asian Female character at full activation (1.0). | 111 |
| 5.2 | The Mouth Open blendshape shown at the 5 activation levels used in our experiments, shown on the White Male character. | 111 |
| 5.3 | The perceptual difference per activation level for our preliminary experiment. | 115 |
| 5.4 | The experiment question screen showing the character we used for training, which is also the character used in the preliminary experiment. | 116 |
| 5.5 | Neutral faces of the characters used in our experiment. Left: white, middle: black, right: asian faces. Top row shows the female faces, while the bottom row shows the male faces. | 118 |

| | | |
|------|--|-----|
| 5.6 | Main effect of Blendshape in the Main Experiment. | 119 |
| 5.7 | Left: The Frown AU shown on the Asian Female and Black Male characters, Right: The Cheeks Puffed AU shown on the Asian Female and Black Male characters. | 120 |
| 5.8 | Perceived differences of activations between characters. | 121 |
| 5.9 | Perceived differences of activations between blendshapes. | 122 |
| 5.10 | Perceived differences of blendshapes between characters. | 124 |
| 5.11 | Geometry errors averaged across characters. | 127 |
| 5.12 | Image errors averaged across characters. Note: we use 1-SSIM for our SSIM graphs, as SSIM is a measure of Similarity between 0 and 1, and we are interested in the perceived difference. | 128 |
| 5.13 | A heatmap visualisation of RMS error on the set of blendshapes used in our main experiment, shown on the template character at full activation (1.0). The blendshapes are shown in order of Perceptual Difference, from highest to lowest. No displacement is dark blue, while the largest displacement is shown in dark red in (as seen in (a) Mouth Open). The colours in order of displacement are: dark blue, blue, cyan, yellow, red, dark red. | 131 |
| 5.14 | Model fits for Perceived difference using geometry metric RMS ((a) and (b)) and image metric SSIM ((c) and (d)) as per models listed Table 5.8. The 6 virtual characters behave in a similar fashion when using RMS (a) and are well captured with the simpler model (b) corresponding to the average model fit across the 6 virtual characters for each blendshape. | 138 |

5.15 Graph showing the increase in perceived difference as RMS error increases for each blendshape. The five points in each line indicate the five activation levels of each blendshape. 141

5.16 Graphs showing the increase in perceived difference as each of our error metrics increases for each blendshape. The five points in each line indicate the five activation levels of each blendshape. 144

Publications

Relevant Publications

1. Carrigan, Emma, Eduard Zell, Cédric Guiard, and Rachel McDonnell “Expression Packing: As-Few-As-Possible Training Expressions for Blendshape Transfer.” Proceedings of the 41st Annual European Association for Computer Graphics Conference. Eurographics Association, 2020. (in press)
2. Costigan, Timothy, Anton Gerdelan, Emma Carrigan, and Rachel McDonnell “Improving blendshape performance for crowds with GPU and GPGPU techniques.” Proceedings of the 9th International Conference on Motion in Games. ACM, pp. 73-78. 2016.
3. Carrigan, Emma, Ludovic Hoyet, Rachel McDonnell, and Quentin Avril “A preliminary investigation into the impact of training for example-based facial blendshape creation.” Proceedings of the 39th Annual European Association for Computer Graphics Conference: Short Papers. Eurographics Association, pp. 49-52, 2018.

Publications Under Review

1. Carrigan, Emma, Katja Zibrek, Rozenn Dayhot, and Rachel McDonnell “Investigating metrics to predict perceptual importance of facial expressions for virtual humans.” Submitted to ACM Symposium on Applied Perception 2020.

Other Publications

1. Carrigan, Emma, Elena Kokkinara, Florian Gheorghe, Maxime Houlier, Stephane Donikian, and Rachel McDonnell “Crowd appearance affects player performance in game combat scenarios.” Proceedings of the 9th International Conference on Motion in Games. ACM, pp. 187-192. 2016.
2. Torre, Ilaria, Emma Carrigan, Killian McCabe, Rachel McDonnell, and Naomi Harte “Survival at the Museum: A Cooperation Experiment with Emotionally Expressive Virtual Characters.” Proceedings of the 2018 on International Conference on Multimodal Interaction. ACM, pp. 423-427. 2018.
3. Torre, Ilaria, Emma Carrigan, Rachel McDonnell, Katarina Domijan, Killian McCabe, and Naomi Harte “The Effect of Multimodal Emotional Expression and Agent Appearance on Trust in Human-Agent Interaction.” In MIG '19: Proceedings of the ACM SIGGRAPH Conference on Motion, Interaction and Games, pp. 1-6. 2019.

Posters

1. Carrigan, Emma, Katja Zibrek, and Rachel McDonnell “Perception of blendshape importance for virtual faces.” ACM Symposium on Applied Perception 2019. Posters.

Chapter 1

Introduction

In recent years, improvements to technology have made it possible to create a highly personalised virtual human that is almost indistinguishable from reality [unr18, 3la18, Sey19]. This desire for highly realistic graphics has, in turn, considerably increased the time, effort, and cost necessary for creating these characters. Additionally, real-time character animation is constrained by the computational power available, limiting the quality of virtual characters even further. With such a strain on the creation process, there is high demand for optimisation of the virtual character animation pipeline.

The recent game release *Hellblade: Senua's Sacrifice* (2017) [Sey16] presents one of the most realistic examples of facial animation in games to date (see Figure 1.1). We can see the attention to detail and the various state-of-the-art technologies used to recreate those minute details which are required to reach what is becoming a standard for games. The protagonist was created using a vast array of cutting edge technologies. She has a complex facial rig, created by scanning an actress, which accurately replicates her facial movements and skin rendering, including blood flow. Her eyes are rendered by combining complicated geometry and lighting to create one of the most accurate



Figure 1.1: *Hellblade: Senua's Sacrifice* (2017)

portrayals of the human eye in gaming. She is animated using state-of-the-art face and eye tracking, recording over 200 feature points at 90fps which are then fed into an animation solver to recreate the actress's performance. This involved the collaboration of multiple companies including 3Lateral, CubicMotion, and Epic Games. Despite all of this, *Hellblade's* budget of \$10 million was considered relatively small [McC18]. This gives us a good idea of just how much work is required to create realistic facial animation.

The creation of realistic human faces is not just a question of rendering, however. Perception of emotion and intent on human faces is now a mechanic that can be used to facilitate gameplay and other interactive applications. *Heavy Rain* (2010) and *L.A. Noire* (2011) are games based on trust and deception of virtual characters, as portrayed by their behaviour. The player must decide whether a character is telling the truth, or attempting to deceive them, in order to solve a mystery. *Detroit: Become Human* (2018) is a critically-acclaimed game that had multiple digital-double characters created



Figure 1.2: *Detroit: Become Human* (2018)

from actor scans and performances (see Figure 1.2). It explores a world where extremely realistic androids exist, following some of these androids through complicated story trees guided by player choices, and exploring a number of ethical questions. Despite a development budget of €30 million [Aud18] and a cast of famous actors, the game received some negative reviews regarding “wooden acting”, which is detrimental to a game that is trying to invoke an emotional response from the player. Realistic graphics does not always equal perceived realism, which is why investigating the impact of these technological advances in facial animation is important. Similarly, there are various games that allow the player to have a relationship with non-player characters, such as the *Dragon Age* series (2009-2014), the *Mass Effect* series (2007-2017), and *Skyrim* (2011). In these games, the relationships are not integral to the main story or gameplay, but provide additional content for players to enjoy. However the characters must be able to elicit some sort of emotion from the player, and a character’s facial animation can hugely affect their ability to do this.

A more recent application for virtual human facial animation is within virtual spaces. Virtual Reality (VR) has recently achieved mainstream availability, causing a huge surge in products and research in the field. One application that is becoming more popular is that of virtual hangouts or meetings in VR. This involves avatars representing users interacting in VR in place of their real selves. While it may be acceptable for casual meetups to have avatars which do not represent the user in any way, for more formal situations like meetings or conferences there needs to be a way to accurately represent human faces, including all necessary emotional and communicative expressions, in VR. This field is becoming popular, with methods already existing to replicate a person's face while reconstructing the upper half of their face which is occluded by the headset [TZS⁺18, LTO⁺15].

As we can see, optimising the creation of realistic facial animation is desirable, however it must be done with the final application in mind. Increasing the realism of the graphics does not equate to increased realism of performance, so it is important to also investigate the perception of the output of any optimisation.

This work focusses on creating virtual characters to be used for real-time applications such as video games, however it is worth mentioning the contrast in workflows between creating characters used for real-time applications using the pipeline discussed in this thesis, or those created for movies using a VFX (virtual effects) pipeline [Sey18]. Notably, the VFX pipeline is used to create animations at the highest possible quality to be used, for example, in a few seconds of footage for a movie, while the real-time pipeline creates characters that can be rendered and animated interactively depending on the application. This greatly affects the focus of certain aspects of the pipelines. While the real-time pipeline aims to automate and optimise the creation of animations such that a large amount of animation at a reasonably high quality can be achieved, the

VFX pipeline heavily relies on manual tweaking, for example frame-by-frame keyframing of an animation, in order to get the highest possible quality output of a much smaller amount of animation.

The goal of this thesis is to optimise the facial animation pipeline, with perceptual analysis of our technological choices. We investigate both the animation and rig creation stages, identify a common factor of blendshape ordering in the optimisation of these stages, and develop perceptual metrics to guide the technical contributions through perceptual experiments.

1.1 Methodology

As we investigate multiple stages of the facial animation pipeline, including analysis and perception of completed facial rigs, i.e. the completed character setup to perform the required facial animation, we require a wide range of methods and metrics to appropriately investigate each stage. Optimisation can be defined in a number of ways, whether it be faster computation times, reduced memory usage, or automising labour-intensive manual tasks. Here we describe the specific techniques that we used for the experiments in this thesis.

1.1.1 Geometric Metrics

In this thesis, we use geometric error for importance ordering, for evaluation of results, and for investigating the impact of said error on perceived difference for various expressions and characters. Root-Mean-Square Error and Hausdorff Distance are commonly used metrics in the field of geometry processing [Lav11, BDBP12], while Triangle Distortion is a metric more commonly used for “as-rigid-as-possible” geometry mapping

methods [LZX⁺08, Oh19]. For our purposes, we use a simplified version of Triangle Distortion in order to measure the “stretch” of triangles after deformation. Here, we give an overview of these metrics and a brief description of how they are used.

Mean-Squared-Error (MSE) and Root-Mean-Squared (RMS) Error of the vertices of a character mesh are used to give the absolute value for the average movement of a vertex between two meshes that share topology. We use these for measuring error in results compared to ground truth (Section 3.2.3), and for analysing and ordering blendshapes based on their variance from the neutral expression (Section 4.1.2 and Chapter 5). In terms of our work with example-based blendshape transfer, identifying large geometric displacements is important when trying to find suitable examples to assist the transfer, as Deformation Transfer [SP04] can produce artifacts when transferring large displacements, and providing training examples can reduce artifacts in these cases. We also use Displacement Strength as a metric for blendshape ordering in Section 3.2, but as we define it as the average displacement of all vertices in a mesh rather than, say, the maximum displacement of a single vertex, it is simply a multiple of MSE or RMS.

Triangle Distortion is a metric designed to measure the stretch of a triangle between meshes with the same topology. Similar to the other geometric metrics, this is useful for example-based blendshape transfer methods as large differences in triangles could cause artifacts in transfer.

Hausdorff Distance is a measure of the distance between two sets of points. In terms of 3D meshes, it is the greatest distance from any point on either mesh to its closest neighbour on the opposite mesh. Unlike the other metrics, Hausdorff distance does not require correlation between the meshes, however all data measured using this metric does share topology.

1.1.2 Image Metrics

While geometric metrics are calculated based on the 3D vertex positions of a mesh, image metrics are calculated on pixel values of a rendered image. In general, image-based metrics are less appropriate to our needs as they are highly affected by application-specific variables such as texture and lighting, and even the scene in which the model is displayed. In the case of our animation optimisation work (Chapter 4), we render images of a character in the neutral pose, as well as each blendshape activated on its own, and use these to calculate image metrics between each blendshape and the neutral face, essentially creating a mock scenario for how this character will be displayed. However, in the case of our perceptual optimisation work (Chapter 5), it can be much more precise, as the images we use to calculate the metrics in this case are exactly those which are seen by the participants.

We use image Mean-Squared-Error (MSE) to give us a sum of all per-pixel differences. While MSE does not take into account human perception, it still describes the difference between two images in a simple and meaningful way.

Structural Similarity Index Metric [WBSS04] and Perceptual Metric [Yee04] are image-based error metrics that attempt to replicate the human vision system. Given the highly perceptual nature of our research, having metrics that attempts to mimic perception is of great interest.

1.1.3 Subjective Metrics

Subjective metrics are commonly used to analyse perception, such as asking participants to submit questionnaires, to make a choice, or to rate something. In this thesis, we use the methods of two-alternative forced choice (2AFC) (Section 3.1) and rating

using a Likert scale (Sections 3.1 and 5.1) [Bla52, Lik32].

2AFC forces a user to make a choice between two stimuli, recording choice and response time as the measurements, also measuring reliability if stimuli are repeated [Bla52].

Likert is a numeric scale used for magnitude estimation, with some or all of the points on the scale labelled. The number of points on the scale is advised to be no less than 5 and no more than 11 [WC11]. Likert provides a numeric rating for each stimulus on an interval scale.

1.1.4 Other

Alongside the metrics discussed above, we use computation time to measure optimisation of a task (Section 4.1) as well as practicality of implementation (Section 3.2). We conduct a user study of our proposed method in Section 3.2 as the pipeline would require an actor to replicate our results, so the feasibility of this is paramount.

1.2 Motivation

Our initial motivation for this research was technical. State-of-the-art virtual characters are computationally intensive both in memory, with new scanning technologies producing models with hundreds of thousands of vertices [FNH⁺17, GZC⁺16] and high-quality animation techniques such as blendshape animation require the storage of hundreds of copies of this model with varying expressions [LAR⁺14], and computationally, as the position of these vertices must be computed for real-time animation in the case of blendshape animation. These blendshapes are either created manually by an artist, or through automatic methods which often require manual cleaning. The cost in terms of time, effort, and money required to create these state-of-the-art characters

is immense, so we considered it of great interest, both for research and industry, to provide solutions to reduce that cost.

We identified bottlenecks in the facial animation pipeline at the animation and rig creation stages and found a common need for blendshape ordering based on an importance metric, which we needed to define in each case. In terms of our GPU animation optimisation work in Chapter 4, we needed to identify the least important blendshapes so we could remove them from the animation. In our character rig creation work in Chapter 3, however, we required a way to identify the most important blendshapes to include in training examples to best improve blendshape transfer results.

This leads us to the perceptual motivation. Our technical contributions required a way to determine blendshape importance, which we defined numerically using information about the geometry of the blendshapes. However, perception of faces is a complex topic, and while there has been much previous work on faces in perception, none have investigated perception of displacement importance on different parts of the face through different Action Units (AUs) [EF78]. For this reason, we decided that a perceptual investigation was necessary, as well as analysis of the metrics used throughout the technical chapter of this thesis with reference to our perceptual results.

1.3 Scope

We investigate the character rig creation and animation stages of the pipeline, however we do not address the actor scanning or rendering stages. Within the animation stage, we focus on optimising computation time and memory usage of real-time animations, we do not investigate offline animation methods or animation creation or retargeting.

We focus on realistic characters as opposed to stylised or cartoon characters who

often have simplified face structures and exaggerated movements.

We suggest our method would be useful for level of detail techniques, however we never conduct experiments to test the effect on perception of different distances. We present our results as estimates for level of detail techniques, but we would require further experimentation for a definitive answer, which was out of the scope of this thesis.

1.4 Limitations

Our work regarding blendshape transfer between characters, including example expressions of the target character to improve transfer (Chapter 3) was limited by the available datasets.

Section 3.1 required a common set of scanned expressions performed by a number of actors, all brought into correspondence with a template character rig in order to perform and evaluate Example-Based Facial Rigging [LWP10]. This required a lot of pre-processing of the meshes, and limited the expressions we could use to those we could find examples of across many characters in the databases we had access to. The pre-processing time also caused us to limit the number of stimuli in our project due to time constraints, as we initially wanted to test a number of subsets of training expressions, but in the end only performed a perceptual evaluation of one set.

We managed to obtain two highly realistic virtual human rigs for our work in Section 3.2 through our collaboration with Eisko, who also brought the rigs into correspondence to allow us to perform blendshape transfer on a realistic test case and compare our results against ground truth. However, as we relied entirely on our collaborators for this data, we were limited to just these two models and could not obtain

more for further test cases.

In our perceptual experiment in Chapter 5, we only used static stimuli and did not investigate the effect of animated stimuli on perception.

1.5 Contributions

We formulate, to the best of our knowledge, the first training example suggestion algorithm for example-based blendshape transfer. Blendshape transfer is a method to assist in the creation of blendshape rigs by transferring a set of blendshapes from one character to another target character, which can be further improved and personalised by including example expressions of the target character. Our novel algorithm identifies important blendshapes and combines them such that the maximum amount of information can be input to the algorithm, thus allowing complex personalised rigs to be created with the minimum number of examples.

Additionally, we suggest a novel GPGPU real-time animation approach with animations passed to the GPU as texture buffer objects. We also investigate the possibility of blendshape ordering metrics for blendshape reduction in the context of level of detail, i.e. removing less important blendshapes for lower fidelity representations.

Finally, we define guidelines on blendshape importance based on their perceptual importance, as well as its relation to various geometry-based and image-based distance metrics, and we present linear models for predicting facial blendshape perceptual importance from geometry and image displacement metrics calculated between the face in neutral and face under expression.

1.6 Summary of Chapters

We begin in Chapter 2 with a discussion on some of the literature in the fields relating to the topic of this thesis, namely creation and animation of virtual humans, and human face perception. We then follow with the technical contributions made during this PhD. In Chapter 3, we introduce the largest contribution of this thesis, which is a method for suggesting optimal training expressions to be used with example-based blendshape transfer algorithms. We begin in Section 3.1 by investigating the perception of facial rigs created using the Example-Based Facial Rigging algorithm with a certain example set, looking for particular blendshapes or face areas that are most or least affected by the inclusion of training examples with this algorithm. We then form, to the best of our knowledge, the first training expression suggestion algorithm for example-based blendshape transfer in Section 3.2. The training expressions are created by combining the most important blendshapes, with some symmetry constraints to make the expressions posable by a human. The importance of blendshapes are defined by a number of importance factors that we discuss in detail. In Chapter 4 we discuss a technique for optimising blendshape animation in the case of multiple users in a shared VR simulation, as well as crowd animations using GPGPU methods. We also suggest level of detail optimisations through reduction of the number of blendshapes, which would lessen the amount of information needed to be sent to the GPU. We discuss different methods for ordering blendshapes in order to choose the least important blendshapes to remove.

In Chapter 5 we perform a perceptual investigation into the idea of blendshape importance, keeping in mind the technical applications of blendshape ordering from the previous chapter. We begin with a preliminary experiment on the perception of

various blendshapes of a highly realistic virtual human rig at various levels of activation, referencing geometric mesh difference metrics to provide a deeper insight into the results. We continue this work with a larger variety of characters and metrics to test the generalisability of our results across different face types. We then present models for predicting the perceptual importance of blendshapes given the geometry or image displacement metrics of a blendshape when compared to the neutral.

In Chapter 6 we summarise the contributions of this thesis and discuss the potential conclusions that can be drawn from our results, and in Chapter 7 we suggest future research on these topics.

Chapter 2

Related Work

The creation of realistic virtual character faces spans many disciplines. Character creation requires 3D scanning capabilities to generate the geometry, as well as high-resolution image capture to recreate the texture. Rigging requires artists to create intuitive controls so that the face can be animated with natural movement. Rendering involves estimating the properties of the physical materials being displayed, as well as the effect of the environment on these materials. On the topic of realistic faces, we cannot ignore the psychology of face perception and how this impacts both the creation of realistic characters and their application. In this chapter, we discuss the background of each of these areas and provide a baseline for which to review the contributions of this thesis.

2.1 Facial Animation Pipeline

The motivation for this thesis is to optimise the facial animation pipeline, focussing on high-quality characters such as main characters in AAA games, so we begin by

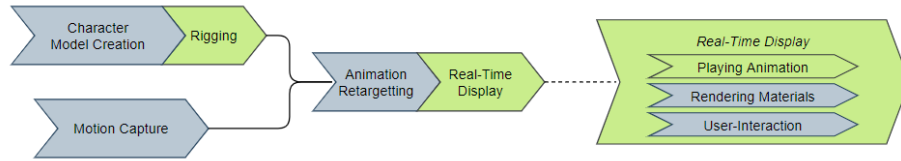


Figure 2.1: The facial animation pipeline. The stages discussed in detail in this thesis are highlighted in green.

providing an overview of the stages of the pipeline. We focus on realistic character creation, as this thesis is interested in the creation and perception of realistic virtual humans. An overview of the pipeline can be found in Figure 2.1, with the particular stages that we focus on in this thesis highlighted in green.

As mentioned in Chapter 1, the pipeline we focus on here is the real-time facial animation pipeline for use with high-quality realistic characters in applications such as AAA video games. When creating animations for use in a pre-rendered video, such as CGI in a movie, the pipeline changes. While it is still necessary to create a character and animate it, these stages no longer rely on optimised algorithms to create characters and animations more quickly to be used in a vast array of scenarios. Rather, they rely heavily on manual work so that each frame of each animation can be perfected for the specific scene in which they are used.

A character will be created with a rig specifically geared towards the sentences and emotions portrayed in each scene, and sometimes this animation is applied to directly to the vertices of the model with no initial rig, with a rig only created as an aid for animators to manually edit the already keyframed animation. This differs greatly from the real-time pipeline which creates characters which have very comprehensive rigs in order to anticipate the wide variety of facial movements they will need to portray throughout a game.

We will continue to focus on the real-time pipeline for the remainder of this thesis.

2.1.1 Character Creation

The first stage of the pipeline is the character creation stage. This can be done either manually by artists or automatically using 3D scanning technologies. Manual character creation is achieved using modelling software such as Maya or Blender [Aut16, Com18, Mar03]. While this allows complete control over the creation, it is also the most intensive in terms of work involved, and can be quite expensive as it requires a skilled artist to create a high-quality model. Automatic methods involve scanning an actor using photographs [PHL⁺06, HSW⁺17] or 3D scans [ZSCS08, GZC⁺16, FNH⁺17]. While these methods are much faster and cheaper than manual character creation, they often still require a manual step to clean and edit the resulting model.

There are a number of fast and refined avatar creation products that only require a single photo and create an avatar within seconds on tablet or phone [Loo20, Pin19, itS20, Did19, ZTG⁺18]. The geometry of the face can be reconstructed using a PCA model with pixel-wise optimisation [HSW⁺17, TZS⁺16, YSN⁺18, SWH⁺17] or a face can be imitated by using dynamic textures [NSX⁺18]. These methods currently produce lower quality or stylised characters, which are not suitable for the use case of high quality game characters that we focus on, however this is another avenue for character creation optimisation that is likely to improve in quality in the future.

There are methods for fast human scanning and rigging, such as fitting a template model to a point mesh scanned using multi-view stereo scanning, which are suitable for many applications [AWLB17]. While the results are realistic, these methods do not produce the high level of quality needed for AAA games or films. A more appropriate



Figure 2.2: An actor being scanned in a light stage.

example for reaching the desired level of realism and detail would be the light stage work of Debevec and colleagues (e.g. [Deb12]). An example of such a photogrammetry system with surrounding light stage used for scanning can be seen in Figure 2.2. The capture described here can yield sub-millimetre measurements of facial geometry in seconds, however it requires a light stage setup which is not readily available, and does not produce animation-ready characters as it outputs only a single scanned expression. To complete the character creation, it is necessary to scan multiple expressions; hundreds, if each blendshape is scanned individually. The geometry of these then need to be simplified in order to be usable in real-time, and the topology needs to be put into correspondence in order for blendshape animation to be possible.

In order to maintain a consistent topology between scans, it is possible to use a template model or shape prior and fit it either to scans or a video sequence using non-rigid registration [ZSCS08, GVWT13, GZC⁺16, FHCP19, IBP15, FNH⁺17]. While this is a suitable solution in many cases, it can reduce the quality of the scan as the fitted mesh is only an approximation of the shape. As well, there is the possibility

of texture drift when transferring the recorded texture to the template mesh. One solution for this is to identify corresponding feature points on the texture rather than the geometry, allowing for correspondence to be created using UV space [CFA⁺16]. While these methods propose rapid creation of blendshape rigs, it is worthwhile to note that the quality of the finalised rig depends on the scanning methods used, and as well does not take into account the necessary reconstruction of obstructed areas of the face such as the internals of the mouth, which is better handled by methods such as deformation transfer [SP04, LWP10].

2.1.2 Rigging

Once a character has been created, it needs to be rigged in order to be animated. Without a rig, a character can only be animated by directly manipulating the vertex of the character mesh and meticulously keyframing each pose. While possible, this method is not only time intensive but also very difficult, and is not feasible in practice.

There are multiple methods for rigging, the most common of which are bone-based rigging [OBP⁺12] (including automatic bone rigging methods [DMOB10, MAF10]) and blendshape rigging [LAR⁺14]. Skeletal animation is conveniently also used to rig bodies, as shown in a basic example in Figure 2.3, and is therefore a natural choice for character artists. It is achieved by creating a skeletal structure beneath the mesh consisting of a number of joints. Each joint affects a number of vertices, with multiple joints able to affect the same vertex. The character can then be animated by simply moving the joints. While this method is intuitive, it can be difficult to create detailed realistic facial movements, even with a complicated skeleton, such as the one shown on the character in Figure 2.4.

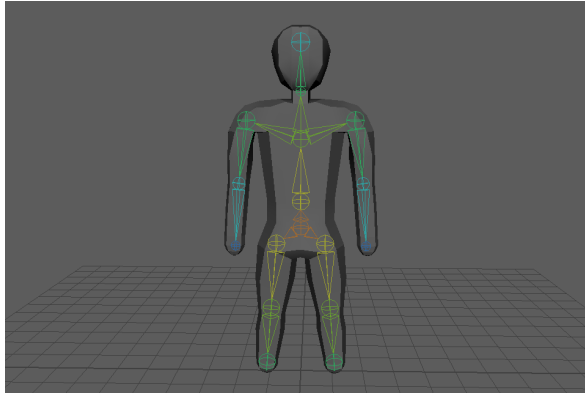


Figure 2.3: A simple bipedal skeleton.

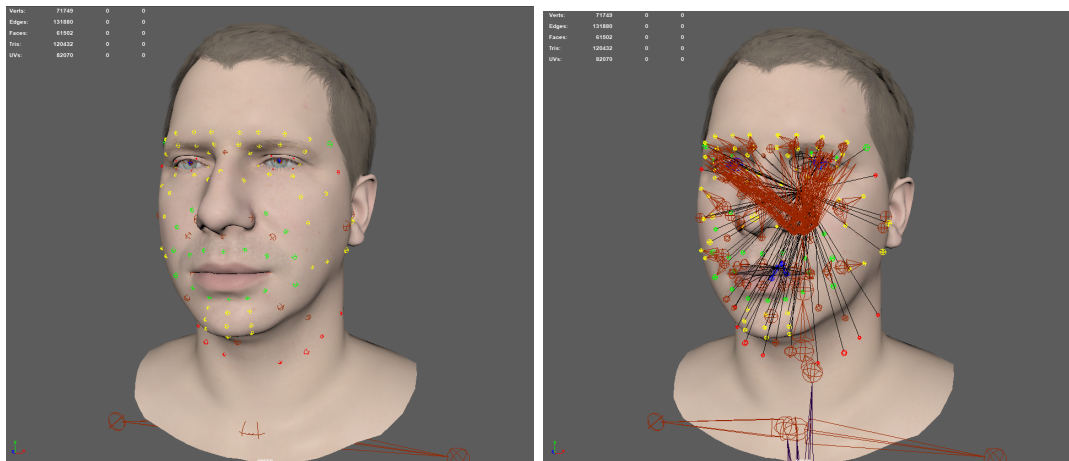


Figure 2.4: A virtual human with the bones of his facial rig shown on the surface of his face (left) and the complex internal structure (right).

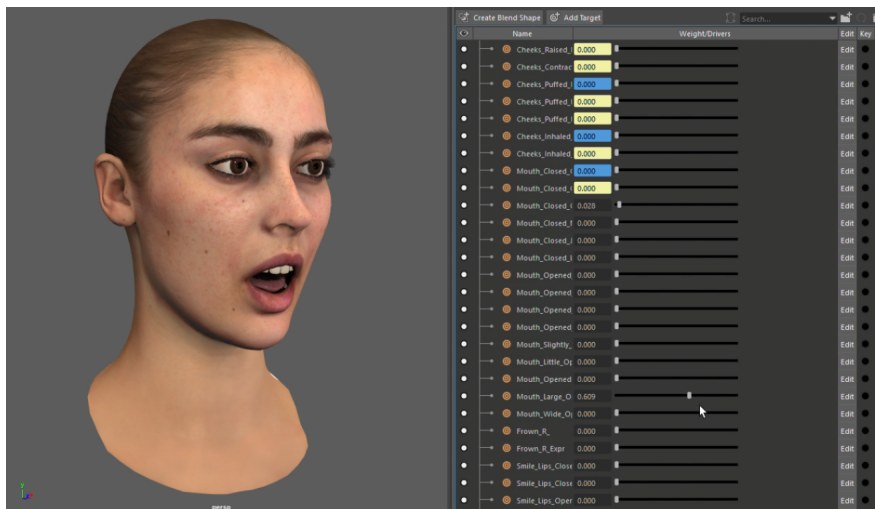


Figure 2.5: The virtual human Louise created by Eisko with an open mouth blendshape at 0.6 activation.

For this reason, blendshape rigs are often favoured for high-quality animation. A “blendshape” is a deformation of the character face representing a certain shape, typically an atomic movement like an eye blink or mouth open shape. A blendshape rig consists of many of these blendshapes, usually hundreds for production level rigs, and a blendshape animation is created by linearly interpolating the vertex positions between the character neutral and a combination of these blendshapes, an example of which can be seen in Figure 2.5. As blendshape rigs are of particular interest to this thesis, we will go into detail in this section.

Blendshape Rigs

The current state-of-the-art for high quality facial animation is blendshape animation [LA10, SSK⁺11, SLS⁺12, LAR⁺14, ZTG⁺18]. This method has seen use in many films such as *The Lord of the Rings* and *Star Wars* [DN08] and games such as *Ryse: Son of Rome* [EMH14]. Personalised blendshape models, i.e. models which are created

to imitate the characteristics and dynamics of a certain person’s face, are also used in related research, such as markerless facial capturing [WBLP11, BWP13, LYYB13, CHZ14, CBZB15, TZN⁺15].

One issue with blendshape animation is deciding what exactly these expression blendshapes will be. There is currently no consensus on what blendshapes a rig should contain, with the decision being left entirely to the artist. However, one solution is to base these expressions on the Facial Action Coding System (FACS) of Ekman et al. [EF78]. FACS defines a set of atomic facial expressions, or more specifically fundamental muscle activations, known as Action Units (AUs). These AUs can be combined to create almost any facial expression, which makes them an ideal fit for a blendshape model. We discuss FACS in more detail in Section 2.2.

Blendshape rigs are non-orthogonal [LAR⁺14], meaning blendshapes may interfere with each other when activated together. To compensate, additional shapes may be added to a rig known as corrective blendshapes. Corrective blendshapes are blendshapes with specific purposes of “fixing” undesirable results of combining blendshapes within a rig caused by non-orthogonality of blendshapes. For example, activating jaw open and smile blendshapes together may cause some unnatural movements as they both affect the mouth area, so a corrective blendshape could be used to fix this unnaturalness and add realism. The detail added by a corrective blendshape is meaningless on its own, but very useful in the context of a combination of specific shapes. Corrective blendshapes may also be used to add realism to a rig, for example adding wrinkles.

Blendshapes are popular for realistic animation as they allow a large degree of artistic control [SILN11]. If additional expressivity is required, one can simply add more blendshapes. Other systems such as linear blend skinning are more constrained and might require a reworking of the animation rig to facilitate new expressions. Blend-

shapes are not without disadvantages though. In particular, they can take a very long time to produce, as each expression must be hand crafted and requires considerably more vertices than other methods. Movie standard blendshape models can often have hundreds of blendshapes, covering not only the basic expressions, but also ones correcting geometric anomalies when certain expressions are active.

As the manual creation of blendshapes is a time-consuming task [Osi07], several methods have been developed to facilitate the transfer of a set of blendshapes from an existing rig to new characters. These methods are an application of surface deformation, in this case deforming one face to match another while both maintaining the personality of the face being deformed and the semantics of the face it is matching.

There are a number of types of surface deformation. Space deformation deforms the space around an arbitrary object, usually defined by a lattice or a cage, and then infers the deformation onto the object [JSW05, LKCOL07, JMD⁺07]. These methods usually require careful manual creation of the bounding area, and can often lose local details [Bec94].

Surface-based deformation methods operate directly on the surface of the mesh, as opposed to its surrounding space. This allows for much more control over the topology and therefore these methods are better suited for maintaining local features. Within this group, there are physically-based methods [TPBF87], as well as methods based on differential surface representations, such as Laplacian-based [Ale01, SCOL⁺04, LSC⁺04] and gradient-based methods [BS07, SP04].

We focus on Sumner and Popović’s gradient-based method of deformation transfer for triangle meshes. This method allows for the transfer of blendshapes from a template rig containing the desired shapes to a target character rig [SP04]. The transfer is achieved through solving a constrained optimisation for the target mesh topology that

matches the source deformations as closely as possible, while maintaining consistency constraints. This was expanded upon by allowing deformation transfer to be applied to more than just single component manifold triangle meshes [BCWG09], or by adding semantic constraints [Sai13] or physical constraints like collision detection [IKNDP16].

The quality and personalisation of these blendshapes, i.e. how well they represent the details of the captured actor’s facial structure and unique expressions, can be improved by providing examples of the target character face [LWP10]. Li et al.’s method performs blendshape optimisation in gradient space, balancing the original blendshape geometry with those of the provided examples. Extensions on Li et al.’s method include actor-specific rig improvements using captured performance as input [MLD⁺16, SML16, BiRZL⁺17]. Similar to the question of which blendshapes should be included in a rig, there is no consensus on which examples should be provided to best improve a rig. This is one of the areas of focus in this thesis.

2.1.3 Animation

Animations are created by posing a character using its rig and changing that pose over time. With no rig you change the individual vertex positions, with a skeleton rig you change the joint rotations, and with a blendshape rig you change the blendshape weights, which are usually constrained in the range 0 to 1. For example, setting a blendshape weight to 0.5 would mean linearly interpolating each vertex position halfway from the neutral position to the blendshape position.

Posing each frame manually, like every other manual stage so far, can be quite time-consuming. For offline animation, i.e. animation created before the running of an application, such as in a movie or cutscene in games, keyframe animation is most

common. Instead of posing each frame, an animator would only pose the “key” frames and allow the animation software to interpolate between the key frames to complete the animation [RP06].

There are automatic methods for creating animation using performance capture [ZTG⁺18]. These methods either use traditional motion capture markers to track facial movements [RDK⁺16] or computer vision methods to track facial features [LKA⁺17] as equivalent to markers, and feed these through a neural network which has been trained to produce an animation on a provided character rig. Performance capture methods can be used to drive online real-time facial animation [BWP13, BGVS19].

Ultra-high quality animation can be achieved using 4D scanning, i.e. scanning an actor over time in order to reconstruct a full performance [DI419]. This method is expensive in both computational time and memory, so it is only applicable as an offline process at the moment. There are methods which do partial 4D animation reconstruction online, however, such as applying wrinkles in real time to a low resolution face mask [CBZB15].

As before, we will focus in particular on blendshape animation. While blendshapes provide the current best solution for realistic facial animation, they come with the disadvantage that they require a significant amount of memory, as information about each of the blendshapes in the rig need to be stored in order for the correct vertex positions to be calculated [SILN11]. As we mentioned before, production-level rigs can have hundreds of blendshapes, so this disadvantage can be quite restrictive. High-quality rigs usually also have a large number of vertices, adding to complexity and memory issues. In this section, we discuss a few methods for optimising blendshape animation.

Retargetting

If the desired motion already exists on a character, another character can be animated using the same motion through motion retargetting, initially used for body motion [Gle98], which is the process of transferring an animation from one character to another. More detailed animations, such as wrinkles, can be retargetted by separating the process into two parts, base mesh retargetting and detail mesh retargetting, allowing for the transfer of highly detailed motion data [NJ04]. As well, characteristics of the target model may be preserved by an additional warping step after the initial retargetting [SCSN11].

Procedural and Behavioural

Another method for animation creation is procedural animation, i.e. animation which is created on the fly at run-time without captured motion. This allows for a wide variety of motion to be created with low memory costs, however the animation quality is lower than that of captured motion. Examples of procedural motion include motion graph animation given an input expression label [SCR⁺18], neural network based methods based on speech [KAL⁺17, SSKS17, TKY⁺17], as well as methods to generate motion based on audio input [LMD12].

Behavioural animation is a type of scripted animation that allows for a character to react appropriately to outside stimuli. Animation is created by defining a set of rules by which a character can animate itself [MHK99, DRPP⁺03, PP02]. This is closely related to procedural animation, in that animation is generated at run-time, however it differs in that there is a strict definition of rules for how a behaviourally animated character can respond. There are also methods which mix behavioural and procedural

animation in order to add more variety and realism to a character [SOC16].

GPU Optimisation

In order to discuss the optimisation of the real-time animation of blendshape rigs, we first review the blendshape equation in Equation 2.1. We define \mathbf{B} as the matrix of unrolled delta blendshapes, i.e. each row is a single blendshape with the (x, y, z) coordinates of the difference between its vertices and the neutral shape flattened into a single vector. \mathbf{b}^0 is the neutral mesh, \mathbf{w} is the vector of blendshape weights, and \mathbf{x} is the resulting expression. To create an expression, each delta blendshape is multiplied by its weight and added to the neutral mesh.

$$\mathbf{x} = \mathbf{b}^0 + \mathbf{w}\mathbf{B} \quad (2.1)$$

One avenue for optimising blendshape animation is by implementing the animation on the GPU. Early attempts to perform blendshape animation on the GPU were limited. Methods based on vertex attributes were restricted to a fairly small number of blendshapes (typically 4). The number of blendshapes could be increased by accumulating the final blendshape over multiple passes but this involved using the CPU to drive the blendshapes. However, over the last 10 years, it has been possible to implement a buffer-template method using more modern GPU features. This improves performance, with greater performance as the number of blendshapes in a model increases. There is a 1.34 times improvement using a small rig of only six blendshapes, which improves to a 2.4 times for a model with 50 blendshapes [Lor07]. One more recent method of GPU animation has displayed results with high enough fps such that it allows for real-time GPU animation, although it is only tested with four blendshapes [YWZ13].

On the GPU, the number of blendshapes we can render is still limited and due to their size. Because of this, blendshapes may benefit from some form of level of detail control to improve performance at the expense of visual quality. Geometry reducing methods could be applied such as the edge collapse method of Garland and Heckbert [GH97] but maintaining correspondence between blendshapes is difficult. While Mohr and Gleicher [MG03] adapted Garland and Heckbert’s method to work with blendshapes, it still remains an expensive process that is only really suitable as an offline pre-process step. While geometry reduction may in theory maintain expressiveness by retaining blendshapes, it may also produce unappealing artifacts and finer expressions may be lost at lower resolutions.

Blendshape Reduction

A better idea may be to retain the geometric complexity of the mesh but reduce the blendshape set. This should theoretically lower the computational and memory burden of the technique but retain the resolution necessary for fine, nuanced expressions. Principal component analysis (PCA) compression is one possible method for compressing the dataset [LYYB13, ZTG⁺18]. The fact that PCA shares many theoretical similarities to blendshapes makes it an obvious candidate but according to Lewis et al. [LAR⁺14] PCA gives poor compression ratios. It also changes the very nature of the model and animations, trading a dense set of intuitive expressions for a sparser but less intuitive set. By unintuitive, we mean that PCA blendshapes do not necessarily represent any recognisable expressions. Work has been done to overcome this by segmenting the face and applying PCA regionally [TDITM11]. Animations also change from large values on a small set of expressions to small values across the entire range of expressions. Overall, these changes make it very difficult for an animator to understand or control

the animation.

Another possible method, according to Lewis et al., is to apply hierarchically semi-separable algorithms to the matrices. This replaces the denser blendshape model with a coarser sparse matrix model but such a system is quite complex to integrate into existing game engines. A potentially better idea is to remove extraneous blendshapes entirely, in reverse order of their perceptual importance. This retains the sensible animation weights and is much easier to integrate into any game engine. Such a method can be applied easily in real-time unlike the geometry reduction discussed earlier. The buffer-template method of Lorach [Lor07] does something similar by only passing active blendshapes to the shader.

For small blendshape reductions, there may be no change in the final animation, as removed expressions might never have been used. Depending on the severity of the reduction required though, more important expressions may eventually be affected. This can be offset through careful consideration of where and when to use lower quality models. Many level of detail methods position lower quality models in the background [ESV99]. This takes advantage of reduced screen resolution to hide any missing details. Lower quality meshes can also be placed in the foreground by exploiting inattentive blindness [LMS10] to direct the user's focus.

Ideally, we would like to retain critical facial features while simplifying less important regions similar to what Duarte et al. [DERC⁺11] try to achieve. The body of research into facial feature importance is quite small, however we find that according to Sadr et al. [SJS03b] it is generally believed that for facial recognition at least, the most important features are the eyes (especially the eyebrows), followed by the mouth and then the nose region. However it has been shown that the recognition of different emotions rely on different areas of the face, with the mouth being important for the

detection of happiness and fear, and the eyes being important for anger, fear, and sadness [BSSM⁺13, WVK⁺17]. Of course, if we consider the case of speech, the mouth would naturally have a higher importance over other areas of the face. It is necessary to consider the context when talking about areas of importance of the face. Removing minor expressions could have unintended side effects on the emotional content of the animation so some manual intervention will likely be required to determine the correct ordering.

Error metrics to assess perceptual dissimilarity for meshes have typically been used for watermarking, simplification, or lossy compression, with the goal being to edit a mesh without introducing a perceptible difference. The types of metrics involved are *view-dependent* and *view-independent*, or image-based and model-based. An overview of these can be seen in the review paper by Corsini et al. [CLL⁺13]. One novelty of our work is to apply error metrics to perceived saliency of facial expressions. Our goal is not to edit a mesh with minimal perceptual difference, but to estimate perceptual difference of expressions with error metrics.

Motion Capture Optimisation

Marker-based motion capture, while trivial for large-scale body motion, is much more complicated when considering small areas where markers need to be more densely placed, such as finger motion, and areas where motion is non-rigid, such as the face. For these cases, marker placement optimisation is crucial, as too many markers will cause confusion (markers cannot be accurately differentiated) and too few will result in loss of detail. In order to address these issues, reduced and optimally placed marker sets are necessary, and can be calculated using PCA [SMB15, WJZ13], k-means [RGL15], linear optimisations [LZD13], or through perceptual evaluation [HRMO12]. Alternatively,

FACS estimation of facial expressions can be estimated by using additional make-up markings on the face to assist in optical facial motion capture [RDK⁺16]. For memory optimisation of motion capture, motion can be segmented and stored hierarchically, and an animation can then be represented by motion pattern indices [GPD09]. Alternatively, lossy compression through Bezier curves and PCA can be used [Ari06]. We expect that these techniques would not be directly applicable to facial animation data due to the difference in scale between subtle facial movements and larger body movements.

2.2 The Human Face

On the topic of realistic human facial animation, we must pay due diligence to research on the human face. In this section, we discuss the anatomy of the face, methods for defining and measuring facial expressions, as well as the purpose of the face with regard to our research.

The face is defined as the front of the head, or more specifically “the various structures between the superciliary arches superiorly, the lower edge of the mandible inferiorly, and as far back as the ears on either side” [DVM09]. It consists of a number of notable features such as the forehead, eyes, nose, cheeks, and mouth. The human face consists of many muscles, most of which can be grouped into the following categories: orbital, nasal, and oral. These groups control the muscles around the eyes, nose, and mouth respectively. The remaining muscles are either auricular, controlling ear movements, or the occipitofrontalis muscles which control the scalp and the skin of the eyebrows. Together, these muscles facilitate speech and perform the important social function of conveying emotion.

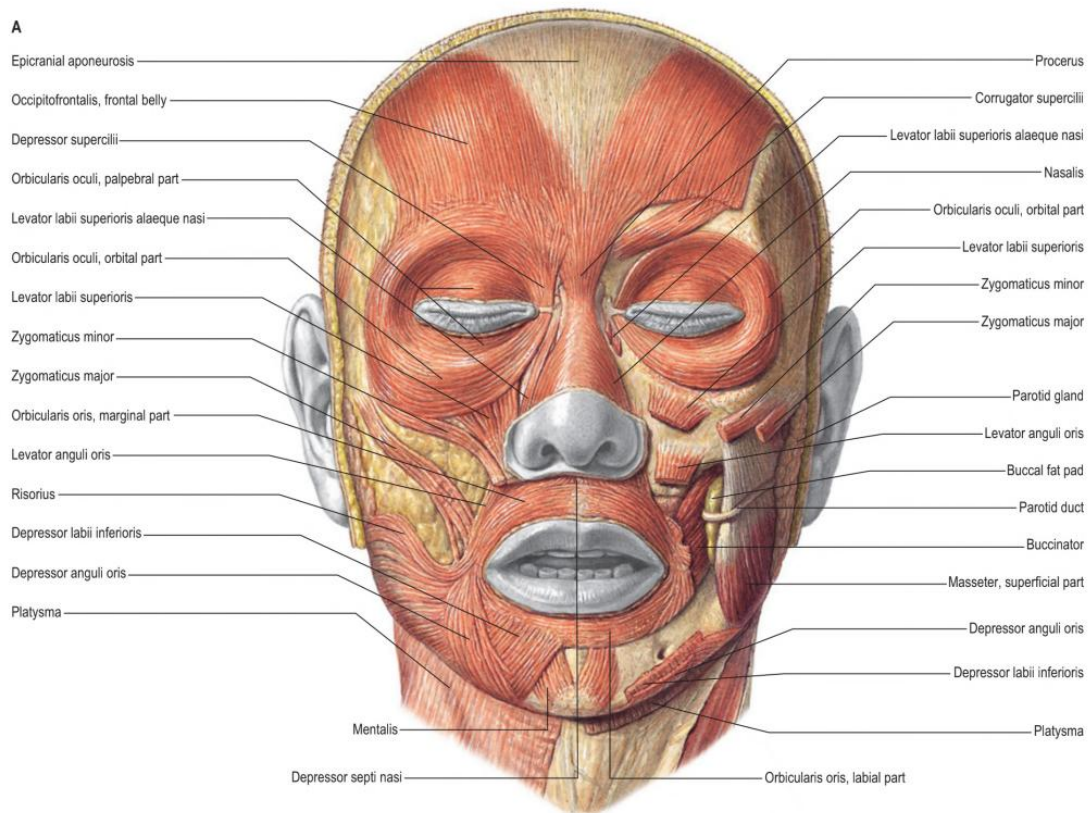


Figure 2.6: The musculature of the face, from *Gray's Anatomy E-Book: The Anatomical Basis of Clinical Practice* [Sta15]

While we can rely on anatomy to define the face in terms of discrete muscles and bones, it is more useful to face perception to define the face in terms of the effect that these muscle movements have. For this purpose, the Facial Action Coding System (FACS) was defined [EF78]. FACS classifies facial expressions in terms of Action Units (AUs), which it considers the fundamental movements of the face, and can be described by which muscles are involved in the movement. This allows us to more easily define facial expressions and conduct facial perception research.

The face has a number of communicative purposes, namely recognition, emotion, and speech. The recognition of faces is what allows us to distinguish people, as well as

determine identifying information from them, while emotion communicates information non-verbally. Speech, while involving internal structures as well, is also a communicative function of the face, as it requires movement of the lips, tongue, and jaw to form specific sounds, and to some degree, even without the auditory accompaniment provided by the vocal cords, speech can be understood through lip-reading.

2.2.1 Recognition

Face recognition is the mechanism through which we identify people by their faces. It tells us information about a person such as their identity, sex, mood, age, race, and direction of attention [TL08]. A person's ability to recognise faces is affected by both the time spent viewing that face, and also the amount of previous experience that person has with the variety of faces they have viewed before [MIGC15].

In the case of the facial structure being unclear, such as in degraded images, it has been shown that faces are better recognised when moving than when static [LCB99], and are better recognised again when animated using their natural motion than an artificial morphed motion [LCW06]. While these results were not significantly reproduced for non-degraded stimuli, it is still worthwhile considering the recognition advantage of animation, specifically natural animation of the person, in facial recognition.

For computer facial recognition, the ability to recognise faces has improved substantially in recent years, with current methods not only able to find faces in images in videos, but recognise face identity with high levels of accuracy [WZLQ16]. One method for face recognition is using Active Appearance Models [CET98, CET01], which learn a statistical model of the shape and grey-level appearance of an object. Other methods often rely on deep features defined internally by a neural network [SYCW17]. Neu-

ral network approaches for facial recognition require large amounts of data. While databases are available that contain real, in-the-wild data, it has been shown that generated databases can be used to train networks with a high degree of recognition accuracy [XGS⁺18]. When discussing face tracking for facial animation, however, more meaningful feature points are tracked, such as the corners of the eyes and mouth, which are then passed to an animation solver to recreate the tracked motion on a virtual character [CHZ14, CWLZ13].

The recognition of Action Units from FACS using computer vision has been explored using facial component models, with AUs being recognised with greater than 95% accuracy [TKC01]. Computer recognition of AUs is interesting to our work as it allows us to assess the similarities between human perception and computer vision. Most AUs were recognised correctly, with incorrect recognition attributing to either an additional similar AU being recognised (e.g. both Inner and Outer Brow Raiser being recognised when only one was present), or a similar AU being incorrectly recognised (e.g. Jaw Drop being recognised instead of Lips Part). It is noted that one of the pairs of AUs that were confused, Cheek Raiser and Lid Tightener, are confused by humans as well [CZLK99]. Recognition of AUs, as well as automatic recognition of intensity of AUs, has also been accomplished using pure deep learning methods, although with lower accuracy [SLC⁺19].

2.2.2 Attention

Attention is interesting both in terms of what draws attention, and what can direct attention. For facial animation, and specifically for optimisation of facial animation, knowing what areas of the face and what type of movements draw the most attention

from the observer is useful. Similarly, we are interested in the ways in which an observer's attention is affected, for example it has been shown that the gaze of a face stimulus can affect the observer [FBT07] which makes it a very relevant topic for facial animation.

It has been shown that selective attention of emotional faces differs depending on the psychological state of the observer, such as a bias towards sad faces for people suffering from or recovered from depression [JG07], or a bias towards angry faces for people with social phobias [MPB04].

Our attention to faces differ depending on many things. For example, there is a difference in gaze between sexes [BPT05], we pay more attention to faces of babies than faces of adults [BSS07] (particularly mothers [TBVM⁺14]), and surprisingly emotion has a negative impact on attention, with studies showing a preference for the neutral face [KVB⁺07].

2.2.3 Emotion

Similar to how FACS was defined to better conduct facial expression research, it is useful to define emotions to further research them. The definition of emotion itself is difficult, and the research into this is ongoing [LHJB10]. At a basic level, it can be defined as the feeling of either pleasure or pain [Fri88]. One definition is that there exists basic emotions. There are a number different definitions of basic emotions, but there is a lot of commonality between them, with a general agreement on the existence of happiness, sadness, fear and anger [EC11, Iza11, Lev11, PW11, PK80]. For our purposes, we follow Ekman's definition that the basic emotions are those of anger, fear, enjoyment, sadness, surprise and disgust [Ekm92a], displayed in Figure 2.7. This

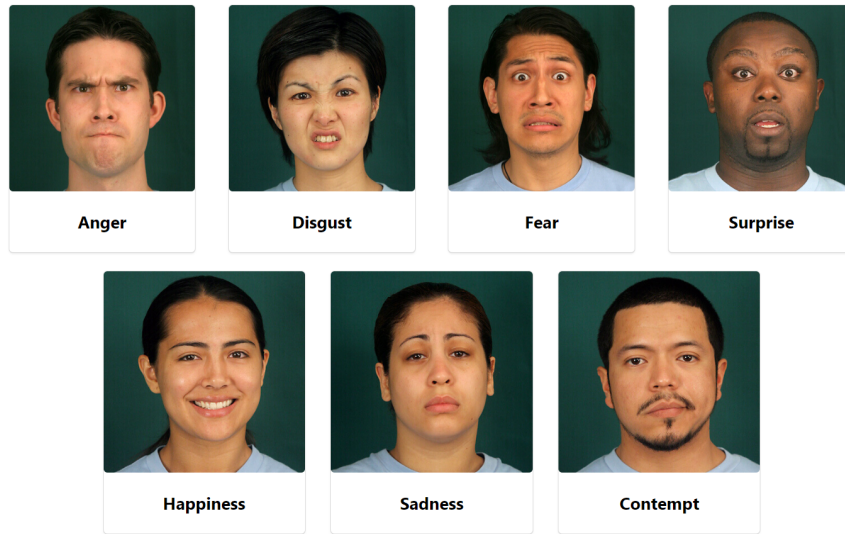


Figure 2.7: The basic emotions as defined by Ekman [Ekm92b], including the later added *contempt* emotion, from *paulekman.com* [Ekm20]

model can also include contempt as a seventh basic emotion, however this was included later and most studies limit themselves to the initial six [TR11].

The definition of basic emotions has sparked some debate [OT90, Ekm92a], and has been challenged using the example that emotions cannot be distinguished at extreme intensities, or even classified as positive or negative [ATT12], such as the happiness of an athlete winning a competition appearing as anger without context, however it is one of the most widely used definitions we have at the moment. A number of studies have investigated emotional faces and found that body cues assist in the distinguishing of emotions. For example, a fearful face can be confused as sad without accompanying body motion, with the inverse being true for happy or angry bodies, which require an accompanying face for distinction [EHEM13, EI15]. Similarly, audio cues have been found to improve emotion recognition and increase the intensity rating of an emotion [HCKH14].

The purpose of emotional expressions is both physiological, such as the widening of the eyes in a fearful expression to increase visual field, and communicative, such as using anger to threaten [ST11]. This purpose may also explain why some emotions are perceived more easily than others. Happiness, the only positive emotion of the basic emotions, is most quickly recognised and least often confused with other emotions [CN08, PC04, LH04], while angry faces, which are important for discerning the danger of a situation, are more easily detected within a crowd [CC13].

For each emotional expression, specific parts of the expression appear to be more important for the classification of an emotion [SCGS05]. Since particular areas of the face are important for the recognition of emotion, different action units could potentially be more salient than others. The evidence supporting this suggests that areas, specialised for the perception of action units, exist in the brain (region pSTS). This could indicate that action units are a necessary precursor to categorization of emotion [SGM16]. In addition, particular action units are responsible for a correct recognition of an emotion [WVK⁺17]: for happiness, this is the lip corner puller and parting of lips; for disgust, the most important are the raising and plucking of the lip. For fear, surprise, anger and sadness the regions around the eyes have the highest weights, with the lid raiser (exposing the sclera of the eyes) important for fear, and the lid tightener significantly most important for anger. Brows are important for sadness and both eyes and mouth contribute significantly to the recognition of surprise.

There were also studies which used the information about individual action units to generate synthetic expressions. A gradual activation of specific action units resulted in detection of an expression [YGS12]. Reverse engineering expressions based on perceptual relevance helped with improved facial recognition in artificial faces [CGZ⁺18]. There is enough evidence to suggest that action units alone have a perceptually signif-

icant impact on emotion categorisation.

While the mouth is understandably a significantly attended to area due to its importance for emotional expression and communication [NCWB08, EA11] and its size relative to other facial features, the eyes and eyebrows can also be considered highly important despite their considerably smaller size. Eyebrows are integral for emotional and conversational signals [Ekm79], and can alter the perception of the eyes [MMMY15], however they are important in their own right for face recognition [SJS03a] and not just in relation to how they change the perception of eyes.

2.2.4 Perception

Face perception is an interesting area of study, as humans have been shown to perceive faces in a different way to regular perception [BY13, FWDT98, KMC97]. As well, the different areas of the face have been shown to be important in terms of speech and emotion perception [BY86, Ado06]. A great deal of research is ongoing in the areas of face recognition, detection, memory, the other-race effect and the effect of experience on face perception, critical features for recognition, and social evaluation of faces. For an overview of recent work, see the review paper by Oruc et al. [OBL19].

Experiments have shown that viewing inverted faces (i.e. upside-down faces) affects our perception of faces much more than of other objects, such as houses. This simple fact opens up an interesting way of investigating face perception. By comparing participant responses when viewing upright faces compared to inverted faces, we can separate perceptual results into those that are face-specific and those that are not [KMC97].

This face inversion effect is linked to the fact that we view faces holistically, rather than as a number of different features combined. This is further supported from results

of other experiments such as the composite experiment, where the participant tries to identify the different people’s faces which are combined to make a new face under different conditions, and the parts-whole task, which tests memory and recognition of isolated facial features, as well as faces with certain features removed [TG11].

It has been shown that people perceive faces of their own race differently to faces of other races, with studies showing an own-race recognition memory advantage [LJC91], as well as an own-race encoding advantage [WT03]. One explanation for this phenomenon is that people have more exposure to people of their own race, and there is evidence that experience can mitigate these other-group effects even if the experience is acquired during adulthood [CPKC09]. For an overview of work in own-race/other-race perception, please see the review paper by Meissner and Brigham [MB01]. There is also a neurological basis for perceptual differences of faces based on both shape and pigment [BN10]. Similarly, a person’s sex has been shown to affect their recognition of faces, with opposite-sex faces being recognised faster [HSL06], as well as their gaze strategy when examining other faces [CBH⁺16].

The area of perception of human faces is extensive, and we focus only on some relevant papers to our work in this thesis. For further information on face perception, recognition, emotion, attention, and the neurological processes involved please see these state-of-the-art papers [Bet12, BI12, JS15, PR07, CN16, NO11].

2.2.5 Perception in Graphics

Notably, face perception research often uses 2D photographs as stimuli. Many perceptual studies have been conducted with artificial data, i.e., static, usually black and white, and often posed photos [Ado06]. These stimuli are usually masked such that

only the inner face is visible, i.e. not including the hairline, ears, or neck. For dynamic stimuli, videos can be used. Both photograph and video stimuli have the same limitation - the need for a human to pose the expression. One solution is image or video editing, however this can cause unnatural results [BY12, DBS18]. This can be considered a major limitation, as unnaturalness of these types of stimuli, while necessary for the purpose of conducting a controlled experiment, may impact the results. With advanced game engines and highly realistic facial rigs, we can conduct perceptual experiments with control over the minute details of how stimuli are displayed. Using virtual characters as stimuli gives us a lot more control over the exact expressions being shown, which areas of the face are activated or not, the texture and lighting of the scene, and the replication of motion across characters. This allows us to do interesting experiments involving changing the level of activation of face or body motion [HJO⁺10] or creating artificial animations with varying face area onset orders [TGRJ19], or investigating linear and non-linear motion [CKH10]. With VR, we can even begin to investigate the psychology surrounding social presence in virtual spaces [ZM19].

Visual saliency of meshes is an interesting topic that leads to applications such as optimal viewpoint selection and mesh simplification. [LVJ05, SF07]. The relationship between 3D geometry and face perception is of particular interest, due to the special way in which we perceive faces as opposed to other objects. Identifying the salient areas of the face, as well as optimal viewpoints, using mesh saliency methods and investigating those through perceptual experiments may give further insight into the special properties of face perception.

While improvements in computation power and capture methods; as well as algorithms to display complex geometry, textures, and animations; allow for more realistic virtual humans to be displayed, it is important that the perceptual impact of these

improvements is appropriately assessed. Perceptually guided character and animation creation can lead to more believable, recognisable, intense, and sincere expressions [DM08, WBCB08].

These perceptual guidelines are of particular importance in the case of automatically generated animations such as those found in human-like agents [CPB⁺94] In this case, a plausible animation must be generated given the context the agent finds themselves in, such as a dialogue. Appropriate head rotations, eye gaze, and facial expression are integral to the perceived naturalness of the agent, and are necessary in order to avoid discomfort in the user interacting with the agent. Similarly, a broad understanding of the social signals involved in human interaction is necessary to accurately understand a situation in order for an agent to appropriately respond [VPH⁺11].

2.3 Conclusion

In this chapter we provided an overview of the fields of realistic facial animation, focussing on the areas of character creation and real-time animation, and the psychology of perception of the human face, with a focus on the effect of structure and dynamics of the face on the perception of emotion, and highlighting the importance of having a deep understanding of this field in order to create the highest possible quality of virtual human.

We discussed the technical implications of current state-of-the-art character creation, noting the high levels of computation power and large amount of memory necessary to store and animate such characters. We introduced the creation and real-time animation of blendshapes as areas which could most benefit from optimisation, and suggested GPU animation, blendshape reduction, and optimised input for example-based

blendshape transfer as methods through which these optimisations could be achieved.

We gave a brief overview of the physiognomy of the face; as well as the areas of face recognition, attention, emotion, and perception; and noted its importance in human interaction, particularly focussing on the fact that the visual saliency of different areas of the face vary drastically and that this idea can drive decisions based on importance weighting of blendshapes for facial animation and rig creation algorithms. We explained that, while the optimisations discussed above may be achievable through purely technical means, it is imperative to assess their perceptual impact through appropriate experimentation.

With this in mind, we present in this thesis optimisations to the facial animation pipeline stages of rig creation and real-time animation with the idea of blendshape importance as a key factor in these optimisations. We conduct perceptual studies of our choices throughout to evaluate and further guide future optimisations in these areas.

Chapter 3

Geometry Optimisation

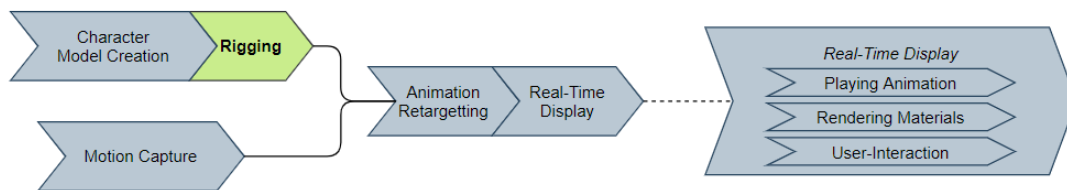


Figure 3.1: The facial animation pipeline with the rig creation stage highlighted.

Our first pipeline investigation is at the rig creation stage (highlighted in Figure 3.1). In the related work in Chapter 2, we discuss the methods of Deformation Transfer [SP04] and Example-Based Facial Rigging (EBFR) [LWP10] for blendshape creation. We identified that there was no method for deciding which examples to provide for EBFR, and that providing a method for calculating which examples to create could be beneficial.

To briefly recap, Deformation Transfer transforms two meshes similarly given a correspondence map between the vertices of the meshes. It achieves this by solving an optimisation problem to transform each triangle of the target mesh to match that of the closest corresponding triangle on the source mesh. It also maintains a consistent

topology by adding the constraint that shared vertices must be transformed to the same place.

Example-Based Facial Rigging further personalises the results of Deformation Transfer by allowing for input training examples which are included in the optimisation problem. The final output is a balance between the exact shape of the source mesh (pure Deformation Transfer) and the nuances of the topology of the target mesh.

We begin by investigating the perceptual impact of training examples provided to EBFR in Section 3.1 by analysing the output of EBFR given a set of training poses to see how well the results reproduced our ground truth actor scans compared to a Deformation Transfer approach. We then go on to formulate a training example suggestion algorithm in Section 3.2, keeping in mind the deficits of Deformation Transfer, which is the method used for transfer in EBFR when no examples are present, such that we prioritise blendshapes to be included in the training examples appropriately. This is achieved by ordering blendshapes based on various importance metrics. Notably, these metrics are purely numeric, and are not perceptually evaluated until Chapter 5.

Finally, we conduct a thorough evaluation of the method, including default parameter suggestions based on some tests and a quick feasibility study, showing that the training examples we suggest are appropriate for digital double creation pipelines that require an actor to pose the example faces.

Section 3.1 was done as part of a collaboration between Trinity College Dublin, and Inria and Technicolor in Rennes, France. I contributed to all parts of this work. Section 3.2 was done in collaboration with a post-doctoral researcher from Trinity. I was the main contributor to everything except for the implementation of the example-based blendshape transfer technique which we used to create blendshape rigs with results from our algorithm.

3.1 Perception of Example-Based Facial Blendshape Creation

The process of creating blendshapes is still very reliant on artists. Although there is a lot of research done on scanning and rigging faces, any facial blendshapes or rigs that are created using these methods require extensive editing for use in games and movies. Additionally, some methods require a large amount of facial scans or motion capture, which can be costly depending on the technology required or the amount of time for which an actor must be hired. Currently, these are necessary costs for any AAA game or blockbuster.

One method which is used in production is Example-Based Facial Rigging [LWP10], which is an extension of Deformation Transfer [SP04]. This method uses a generic blendshape rig template, i.e. a neutral face with a number of blendshape target faces, as well as the neutral face of the character for which you want to create the rig, and a number of facial poses of this character. The algorithm recreates each of the blendshapes of the generic rig for the desired target face, while also incorporating facial details which it learns from the supplied facial poses.

While this method is used in professional pipelines, companies still need to hire an actor to create numerous poses and 3D modelling artists to clean the final blendshapes. However, we believe that we can improve the blendshape creation process to cut down the dependency on actors by finding the optimal types of poses to supply to the system.

Our contribution is a preliminary perceptual study into the effect of a set of input scans on the EBFR system. From this, we can see what areas of the face were most improved by the algorithm and which areas were least improved. This gives us an idea of the impact of our supplied training poses and is a basis for further research into

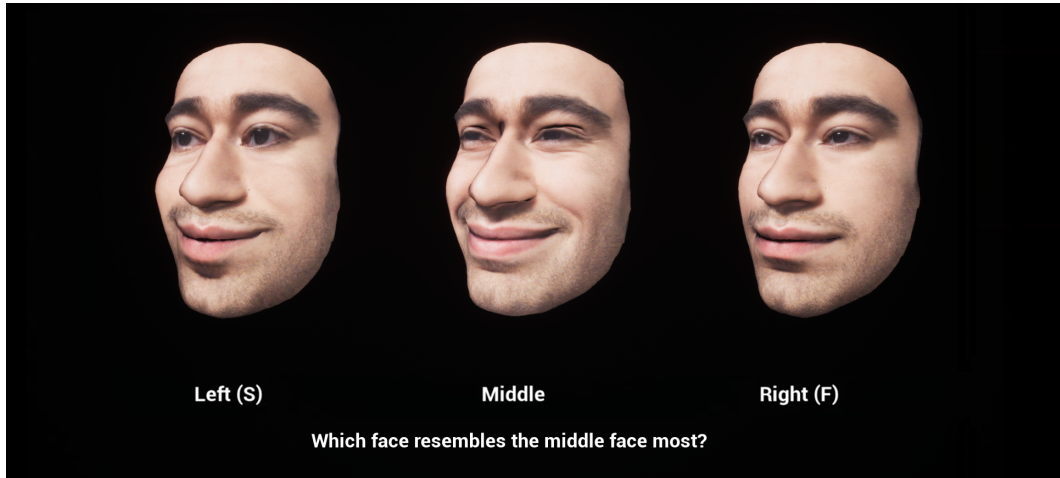


Figure 3.2: An example of stimuli shown to participants comparing the facial rigs created without training, and those with training.

reducing the number of scans needed to attain suitable blendshape rigs using EBFR.

3.1.1 Method

Our idea was to use all the common training poses available across a number of actors, which would theoretically create the best possible trained rigs for our data using EBFR. We then ran an experiment comparing these trained rigs, as well as untrained rigs created using the Deformation Transfer method, to a ground truth facial scan. The participants chose which of DT or EBFR faces best resembled the ground truth, then described how close their chosen pose was to the ground truth on a 5-point Likert scale. This showed us which parts of the face were most improved or disimproved by EBFR. An example of the stimuli can be seen in Figure 3.2.

Stimuli

We used the Bosphorus Database to get data for our experiment [SAD⁺08]. The data is provided as point clouds with textures. We meshed, cleaned, normalised and registered

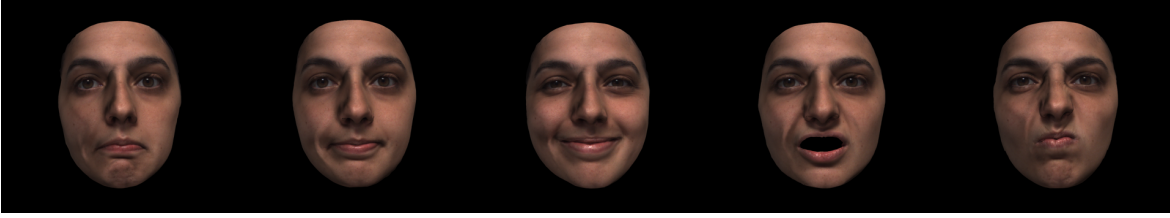


Figure 3.3: Examples of scanned poses from the Bosphorus database.

this data to create meshes with consistent topology for ease of use for our experiment. After this preprocessing step, we had a large number of facial expressions from different actors which we could use both as ground truth and as input to EBFR. An example of the cleaned data we used can be seen in Figure 3.3.

The Bosphorus database is a database of facial scans of over a hundred actors attempting to recreate the Action Units (AUs) as described in Ekman et al.’s Facial Action Coding System (FACS) [EF78], however due to the difficulty of activating certain facial muscles in isolation, most scans are actually a combination of AUs. Fortunately, each scan in the database has been annotated by a FACS expert. Using this information, we can attempt to recreate the scans using our facial rigs created using DT and EBFR, as the blendshapes of the rigs we used were based on FACS.

We selected 4 female and 4 male actors from this database, and created trained and untrained rigs for each selected actor. The untrained rig was created using Deformation Transfer using a generic facial animation rig whose blendshapes were based on FACS AUs. The trained rig was created using Example-Based Facial Rigging, using the same generic model and neutral pose as the untrained rig, but including 19 additional facial scans of the actor with different expressions as training poses.

The expressions in the scans used as training were the same across all actors. These expressions consisted of the 19 Action Units as detailed in Table 3.1. These expressions

| AU No. | FACS Name | Expression No. |
|--------|----------------------|----------------|
| 9 | Nose Wrinkler | 0 |
| 10 | Upper Lip Raiser | 1 |
| 12 | Lip Corner Puller | 2 |
| 14 | Dimpler | 3 |
| 15 | Lip Corner Depressor | 4 |
| 16 | Lower Lip Depressor | 5 |
| 17 | Chin Raiser | 6 |
| 18 | Lip Pucker | 7 |
| 22 | Lip Funneler | 8 |
| 23 | Lip Tightener | 9 |
| 24 | Lip Pressor | 10 |
| 25 | Lips Part | 11 |
| 26 | Jaw Drop | 12 |
| 27 | Mouth Stretch | 13 |
| 28 | Lip Suck | 14 |
| 34 | Cheek Puff | 15 |
| 2 | Outer Brow Raiser | 16 |
| 4 | Brow Lowerer | 17 |
| 43 | Eyes Closed | 18 |

Table 3.1: The Action Units we used in our experiment with their FACS names. The third column shows the numbers we used for them in our experiment. Expressions 0-15 are lower face expressions, 16-18 are upper face.

were chosen because they were the most commonly represented expressions in the database and we required a common set of expressions across all actors. They were also chosen to convey information from all the different areas of the face, e.g., mouth, nose, eyes.

Participants and Procedure

Participants were presented with 152 trials: 2 Actor Sex (Female, Male) \times 4 Actors \times 19 Expressions. In each trial, participants were presented with the ground truth face (scan) in the middle of the screen, and both the trained and untrained faces randomly presented on the left or right side of the screen (Figure 3.2). Participants could rotate

the faces simultaneously using the arrow keys on the keyboard, to a maximum of 30 degrees in each direction. For each stimulus they were asked “Which face resembles the middle face most?”, and answered using the S and F keyboard keys. They saw the faces for a maximum of 10s, after which they were forced to provide an answer. Then they were asked to rate how close the face they selected was to the middle face on a scale from 1 (Not at all) to 5 (Identical) using the keyboard. The trials in the experiment were presented in blocks: each actor of one gender was presented in a random order, then the actors of the other gender. The genders were presented in a randomised order. All the expressions for one actor were presented in a random order before moving to the next actor.

We included training stimuli at the beginning of the experiment, identical across participants and using an actor who did not appear in the experiment. The participants used these stimuli to become familiar with the experiment and the buttons needed to answer our questions. Responses for these stimuli were not recorded. A screen was shown between the training and real experiment to warn the participants that their responses would begin to be recorded.

Twenty-three participants took part in our experiment (5 female, 17 male and 1 other, aged 23-61 years). They viewed the experiment on a 24” display of resolution 1920x1200. Each participant was given an information sheet and consent form to sign. The information was repeated on the screen at the beginning of the experiment. The participant was then asked to input age and sex before they began the experiment.

3.1.2 Results

To assess whether trained (EBFR) faces were preferred to untrained (DT) faces, as well as whether differences appear for different parts of the faces, we performed a one-way repeated measures Analysis of Variance (ANOVA) with within-subject factors *Expression* on the percentage of times EBFR was preferred over DT. To analyse these results, each participant's results were averaged across all the actors for each condition. All effects are reported at $p < 0.05$. When we found main or interaction effects, we further explored the cause of these effects using Newman-Keuls ($p < 0.05$) post hoc tests for pairwise comparisons.

First, we found a main effect of *Expression* ($F_{18,396} = 41.65$, $p \approx 0$), where post hoc analysis showed that EBFR was clearly preferred for some expressions, and less for others (Figure 3.4). To further explore these effects, we conducted single t-tests against 50% to evaluate if preference was above chance level ($p < 0.05$). Results showed 3 categories of expression, which are listed below:

Improved by EBFR: 0, 1, 2, 3, 5, 6, 7, 8, 9, 14 and 15

No preference between EBFR and DT: 4, 10, 12, 17 and 18

EBFR worsened the results: 11, 13, 16

Excluded Results

We found that, for some of the expressions, participants preferred the untrained faces across all actors, which was unusual as we expected the trained faces to be equal or better in every case. In order to understand why, we manually examined the stimuli and found some artifacts across almost all actors for certain expressions.

There were texture artifacts for expressions 16 and 18, with FACS names Outer

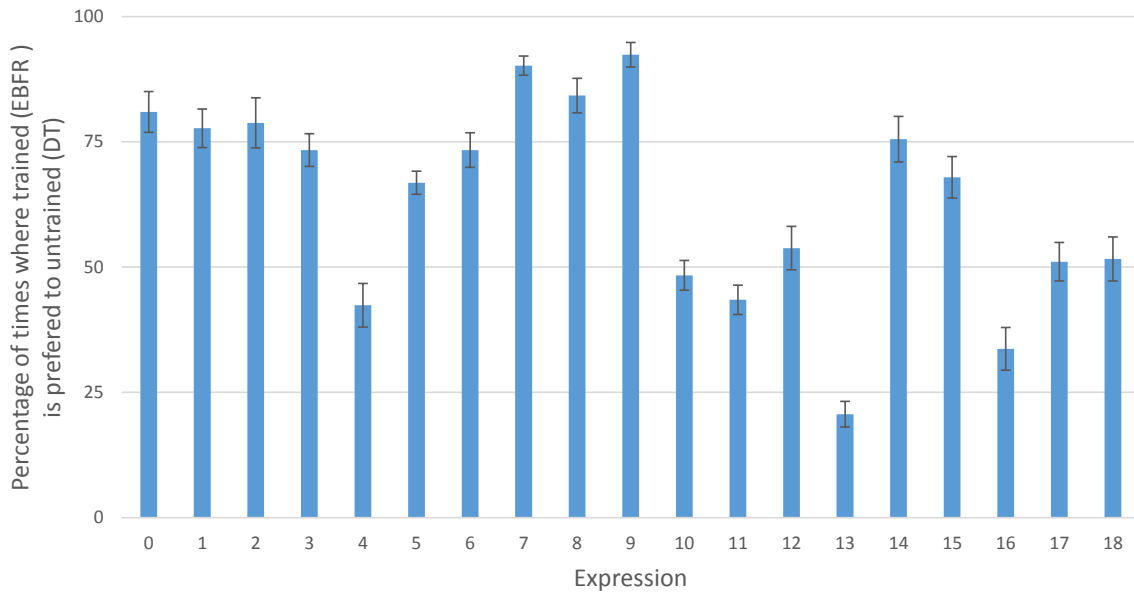


Figure 3.4: Main effect of Expression on preference of EBFR over DT.

Brow Raiser and Eyes Closed, as can be seen on the left in Figure 3.5 (a). Although our interest was purely morphological and we asked participants to ignore texture artifacts to the best of their ability, we found these artifacts to be too noticeable to ignore. For this reason, we chose to exclude expressions 16 and 18 from our analysis.

While we could have avoided these issues by removing the textures on every stimulus, we found that the meshes with no texture were unnatural and might have affected the perception of participants, as they were too unlike real faces. As we are interested in human facial perception, we decided to include the textures to ensure the faces looked as human as possible.

We also found that the trained stimulus for expression 13 (Mouth Stretch) was often unnatural looking, which we found to be caused by an error in scaling the scan from the database. In our data cleaning process, we scaled the faces to be of unit length. This had a strong negative effect on expression 13, as the actor opens their mouth as wide as possible, which causes the face to be a lot longer than when at rest. In the



(a) Texture artifact example



(b) Left: Neutral scan, Centre: Expression 13, Right: Trained rig recreation of expression 13

Figure 3.5: (a) The texture artifact which affected expressions 16 and 18. (b) The artifact which affected expression 13.

training process, we were essentially telling our algorithm to make the neutral actor scan (Figure 3.5 (b), left) shrink to match the scanned expression 13 (Figure 3.5 (b), centre). This resulted in an unnatural face (Figure 3.5 (b), right). For this reason, we excluded expression 13 from our results.

Analysis

After removing the results that were caused by artifacts, we can separate the results into groups as shown in Table 3.2.

We excluded the Mouth Stretch expression, as our data cleaning algorithm scaled the faces so the meshes would be unit-length from top to bottom. This made Mouth Stretch smaller than it should have been. However, we did not exclude Jaw Drop as there were no obvious artifacts, although it appears a similar issue may have happened. Jaw Drop is a slightly longer than normal face, so the scaling should have affected this expression unfavourably as well.

Brow Lowerer was the only upper face expression that remained after we excluded results. We had a noticeable lack of upper face expressions to choose from, and two of

| AU | FACS Name | Exp. |
|-----------|---------------------|-------------|
| 9 | Nose Wrinkler | 0 |
| 10 | Upper Lip Raiser | 1 |
| 12 | Lip Corner Puller | 2 |
| 14 | Dimpler | 3 |
| 16 | Lower Lip Depressor | 5 |
| 17 | Chin Raiser | 6 |
| 18 | Lip Pucker | 7 |
| 22 | Lip Funneler | 8 |
| 23 | Lip Tightener | 9 |
| 28 | Lip Suck | 14 |
| 34 | Cheek Puff | 15 |

(a) The expressions where EBFR was significantly preferred.

| AU | FACS Name | Exp. |
|-----------|----------------------|-------------|
| 15 | Lip Corner Depressor | 4 |
| 24 | Lip Pressor | 10 |
| 26 | Jaw Drop | 12 |
| 4 | Brow Lowerer | 17 |

(b) The expressions where there was no significant difference between EBFR and DT.

| AU | FACS Name | Exp. |
|-----------|------------------|-------------|
| 25 | Lips Part | 11 |

(c) The expressions where DT was significantly preferred.

Table 3.2: Results grouped by ratio of trained to untrained responses.

the three expressions we had were excluded due to artifacts. Brow Lowerer’s neutral result may be caused by not having enough upper face training poses.

Interestingly, for Lip Corner Depressor and Lip Pressor, expressions which cause the lips to be pushed together and stretched, it seems that the algorithm simply has a hard time recreating these. For Lip Pressor, the lips become quite thin, which confused our training algorithm and seemed to accentuate some sharp edges around the lip contour, and sometimes caused overlapping faces. For Lip Corner Depressor, the downward movement seemed to make the mouth open slightly in some cases, and stretch the bottom lip to make it look slightly larger in other cases. Lips Part had a similar issue in that it often made the contour of the lips slightly sharper. These small errors seem to be enough to affect the perception of these expressions.

3.1.3 Discussion

We created a number of facial rigs using EBFR and showed that EBFR produces perceptually better facial rigs than Deformation Transfer. We found that artifacts caused by the algorithm that affected the contour of the lips were more noticeable than artifacts that affected the other areas of the face. Our results indicate that the lip area is important when creating facial rigs. However, it is possible this was caused by the lack of an internal mouth structure. This caused any opening of the mouth to be very apparent. Future work will investigate the importance of the internal structure.

More interesting is the fact that the Lips Part expression was noticeably affected. This expression had the mouth slightly open, so we see that it is not the difference between an open and closed mouth that is noticeable, but the actual shape of the lips, specifically the edge between the lip and the inside of the mouth.

We cannot exclude the possibility that the database we used impacted our perceptual results. While we saw an overall improvement from using EBFR compared to Deformation Transfer, the specific expressions impacted could be due to the low quality of the meshes we used, or due to the lack of an internal mouth structure, or due to the geometry or texture artifacts. For our future investigations, we prioritize developing our own pipelines to acquire high quality models for testing, rather than relying on available open-source databases.

We continue our research into finding optimal training expressions for example-based blendshape transfer techniques. We implement a novel training expression suggestion algorithm, which combines blendshapes based on importance to create training expressions that efficiently provide the most salient information to the blendshape transfer method.

3.2 Expression Packing

Creating high-quality, production-ready animation rigs for individual characters is a time-consuming task which is a major bottleneck in current facial animation pipelines, where blendshape interpolation is still the method of choice for real-time [Sey16] and offline animation [Sey19]. Aiming for high quality, it became common to acquire 3D scans for digital doubles, while high-resolution models of fictional characters are sculpted in 3D. The resulting 3D models and expressions define the shape down to the pore level, but vary with regard to the vertex count and are therefore lacking the required one-to-one vertex correspondence between blendshapes. In order to ensure equal numbers of vertices and consistent connectivity across expressions, the 3D meshes are retopologised, in general at lower resolution. Academic work largely automated this process by registering a template blendshape model non-rigidly towards the 3D scan or sculpted model (e.g., [IBP15, FNH⁺17, LBB⁺17]). Within these frameworks, the template model is consistently deformed until it accurately approximates the shape of the 3D scan or sculpted model. To simplify the creation of blendshapes further, several algorithms have been proposed that transfer existing generic blendshapes from a template model to new characters [SP04] or personalise the generic blendshapes based on training expressions [LWP10, SML16]. So far, the semantics of these training expressions, whether it should be a smile, frown or any other expression, have been defined manually.

In general, blendshape transfer methods offer a trade-off between fast generation of plausible (but not necessarily artefact-free) blendshapes and creation of accurately-personalised blendshapes, at the cost of requiring many training expressions. While the two extremes are well defined (either no training expressions or one training ex-

pression for each blendshape) no obvious solution exists for an optimal set of training expressions. By optimal, we mean that we aim to satisfy the following four goals:

1. The number of training expressions should be as-few-as-possible to reduce the number of expressions to scan or sculpt.
2. To ensure accurate extraction of personalised blendshapes, no overlap of blendshapes should exist in training expressions. (e.g., instead of combining two mouth blendshapes, a combination of an eye and a mouth blendshape should be preferred).
3. If the number of training expressions is limited, blendshapes that would be problematic to transfer without examples, should be prioritised.
4. Training expressions should be poseable, meaning that a human face should be able to reproduce them.

After reviewing recent blendshape creation pipelines [FNH⁺17], we observe that the template blendshape model used in blendshape transfer methods is actually available before individual expressions are scanned or sculpted. Because the blendshapes of the template and the final, personalised rig will be similar, we can combine several non-interfering blendshapes to create complex expressions and use them during scanning as semantic references for the actual training expressions (Figure 3.7). To our knowledge, we present the first numerical optimization method that automatically combines blendshapes to form optimal expressions. In addition, results of our optimization largely satisfy the above mentioned criteria. We pre-compute a minimal set of reference expressions, ordered by their power to improve the overall results of the blendshape transfer operation. Considering only the first reference expressions of our

minimal set will improve blendshape transfer methods as-much-as-possible for an incomplete set of training expressions. This is an often encountered practical case, for example if capturing-time is short due to the availability of celebrities, or the budget limits the number of training expressions that can be post-processed. An entire run of our optimization returns the smallest set of reference expressions for each blendshape and this set is significantly smaller than the number of blendshapes of the template rig. Finally, our method is generic as it is suitable for any blendshape basis. A reference implementation is published on GitHub to facilitate replication and comparison¹.

3.2.1 Blendshape Transfer

An example-based blendshape transfer method requires a template blendshape rig \mathcal{A} , consisting of triangle meshes posing a neutral expression and K blendshapes. All meshes are of identical connectivity and vertex number (after non-rigid mesh registration [FNH⁺17]). We denote the neutral expression and all blendshapes as the set $\mathcal{A} = \{\mathbf{v}_0^A, \mathbf{v}_1^A, \dots, \mathbf{v}_K^A\}$, where the vertex positions \mathbf{v}_k^n are stacked in a single vector $\mathbf{v}_k = (\mathbf{v}_k^1, \dots, \mathbf{v}_k^N)^T$ of size $3N$ due to the coupling of xyz -coordinates. Delta-blendshapes are defined as: $\delta\mathbf{v}_k = \mathbf{v}_k - \mathbf{v}_0$. In addition to the template blendshape model, a target model exists of different identity \mathbf{v}_0^B . For the target model \mathcal{B} , J training expressions \mathbf{x}_j^B are provided ($\mathcal{B} = \{\mathbf{x}_1^B, \dots, \mathbf{x}_J^B\}$) with the same number of vertices and connectivity as \mathcal{A} . For each training expression, the (approximate) blendshape weights w_k^j are known and the goal of the blendshape transfer function is to compute the missing personalised blendshapes $\{\mathbf{v}_1^B, \dots, \mathbf{v}_K^B\}$ of \mathcal{B} , satisfying the following equation which

¹<https://github.com/Fiquem/Expression-Packing>

describes the training expression \mathbf{x}_j^B as a function of the inputs $\{\mathbf{v}_1^B, \dots, \mathbf{v}_K^B\}$

$$\mathbf{x}_j^B(\mathbf{v}_1^B, \dots, \mathbf{v}_K^B) = \mathbf{v}_0^B + \sum_{k=1}^K w_k^j (\mathbf{v}_k^B - \mathbf{v}_0^B) \quad (3.1)$$

We assume that the two blendshapes \mathbf{v}_k^A and \mathbf{v}_k^B are semantically equivalent expressions of \mathcal{A} and \mathcal{B} and of identical connectivity. In our work, we focus on finding the semantically equivalent expressions \mathbf{x}_j^A that serve as a reference for creating the training expressions \mathbf{x}_j^B , either by modelling the expressions manually or by posing the expression for a 3D scan. Our main intention is to obtain the semantics for as-few-as-possible training expressions J , and obtain the most accurate unknown blendshapes $\{\mathbf{v}_1^B, \dots, \mathbf{v}_K^B\}$ as a result of the example-based blendshape transfer method. A high-level overview of this method can be seen in Figure 3.7. For greater clarity and convenience we summarise our notation in Table 3.3.

3.2.2 Optimal Reference Expressions

In the following, we will derive step-by-step our optimization method for pre-computing the reference expressions \mathbf{x}_j^A for any blendshape basis. To simplify notation, we largely omit the indices A and B within this section. Variables without an index refer to the blendshape model \mathcal{A} , e.g. $\mathbf{v}_k \equiv \mathbf{v}_k^A$. We first introduce the concept of binary blendshapes, which is a fundamental conceptual basis for our optimization framework. Afterwards, we discuss important modifications to obtain numerically optimal as well as plausible expressions. We close this section with a discussion on how to solve the optimization function and the benefits of different solutions.

| Variable | Description |
|---------------------------------------|--|
| j, J | index / absolute number of examples |
| n, N | index / absolute number of vertices |
| t, T | index / absolute number of triangles |
| $k, K / m, M$ | index / absolute number of blendshapes |
| \mathcal{A}, \mathcal{B} | blendshape rigs |
| \mathbf{v}_k^n | position of vertex n of blendshape k |
| $\mathbf{v}_0, \mathbf{v}_k$ | neutral expressions and blendshape k |
| $\delta \mathbf{v}_k$ | delta-blendshape k , with: $\delta \mathbf{v}_k = \mathbf{v}_k - \mathbf{v}_0$ |
| $\delta \mathbf{v}_k^n$ | displacement of vertex n and blendshape k |
| w_n | weight of blendshape n |
| \mathbf{x}_j | example expression |
| \mathbf{b}_k | binary blendshape n |
| b_k^n | value for vertex n in \mathbf{b}_k , $b_k^n \in \{0, 1\}$ |
| λ | blendshape importance weight |
| s | distortion metric between two triangles |
| d | Euclidean distance between two vertices |
| $\mathbf{u}, \mathbf{v}, \mathbf{w}$ | vertex positions of a triangle \mathbf{uvw} |
| $(w^\triangleleft, w^\triangleright)$ | blendshape weights of a symmetric pair |

Table 3.3: Notation overview of the most relevant variables.

Binary Blendshapes

Example-based methods estimate personalised individual blendshapes from a training expression. This problem is ill-posed when an expression consists of overlapping blendshapes (e.g., if an expression is composed of a smile and a mouth-open blendshape). In contrast, computing individual blendshapes is easy from a training expression consisting of, for example, an eye-closing and a mouth opening blendshape as there is no overlap. To reduce ambiguity within training expressions, we search for a metric that allows a balance between combining as many blendshapes as possible in one training expression and preventing too strong an overlap between blendshapes within the training expressions (Goal 2).

For this purpose, we introduce a new concept of binary blendshapes (Equation 3.2). Delta-blendshapes $\delta\mathbf{v}_k$ define vertex displacement with respect to the neutral expression \mathbf{v}_0 , and binary blendshapes \mathbf{b}_k will contain the information of the location of relevant deformation within the blendshape. In the following, vertices of blendshape k with zero displacement ($\delta\mathbf{v}_k^n = 0$) will be called *static*. In addition, vertices with a displacement bigger than zero will be divided into two stages: *active* and *smoothing* vertices, depending whether their displacement is strong and important for recognizing the semantic meaning or whether the deformation mainly exists to maintain a smooth deformation. See Figure 3.8 for an illustration. It is worth mentioning that the differentiation between static and non-static vertices is a common part of example-based blendshape transfer methods, either in the form of soft constraints [LWP10] - remain similar to the original blendshape, or hard constraints [SML16] - static vertices remain

static.

$$b_k^n = \begin{cases} 0 & \text{if } \|\delta\mathbf{v}_k^n\| < \mu \max_{n \in N} \|\delta\mathbf{v}_k^n\|, \\ 1 & \text{otherwise} \end{cases}, \quad 0.25 \leq \mu \leq 0.4 \quad (3.2)$$

The information of whether a vertex is active or non-active (static or smoothing) is saved as a binary value $b_k^n \in \{0, 1\}$ within a vector that we define as the binary blendshape $\mathbf{b}_k = \{b_k^1, \dots, b_k^N\}$. Based on our experiments in Section 3.2.3, we recommend a relative threshold of 25 – 40% of the maximum displacement within the blendshape to decide if a vertex is active or non-active. We preferred a relative metric over an absolute one, as this scales better across subtle and strong expressions.

Optimization Problem

Creating a minimal set of reference expressions \mathbf{x}_j for a blendshape transfer method can be defined as variation of a weighted set packing algorithm [Kar72]. Given a finite set that defines all unique elements (dictionary) together with a collection of subsets, each consisting only of a fraction of the unique elements, the set packing algorithm identifies a group of subsets that are pairwise disjoint (no unique element appears twice) and that have the maximum number of unique elements.

Applied to our problem, the list of vertex indices $\{1, \dots, N\}$ is the finite set (dictionary), with every vertex index n defined as a unique element. Each binary blendshape \mathbf{b}_k defines a subset of active vertices (vertices with strong displacements). The optimization problem (Equation 3.3) is to pack as many blendshapes as possible in one expression (Goal 1), which is equivalent to maximizing the number of blendshapes with $w_k = 1$ (Equation 3.3a). At the same time, combinations of blendshapes should be prevented if their individual active vertices overlap (Goal 2). This can be expressed

as a linear constraint (Equation 3.3b) for every vertex n . In the end only two states are relevant for a blendshape, either it is part of the expression or not. We observe in practice that it is easier to model and pose extreme expressions, rather than accurately extrapolating from in-betweens. While a closed-eye expression is well defined, small inaccuracies in a half-closed eye might negatively affect the closed-eye expression. For this reason w_k is always 0 or 1 (Equation 3.3c), making the problem an integer optimization problem. If only a limited set of example expressions can be provided, we would like to control which blendshapes should be chosen first (Goal 3). For this reason, we introduce the importance weight λ_k in Equation 3.3a. Details on computing λ_k are described in Section 3.2.2 and evaluated in Section 3.2.3.

$$\max_{w_k} \sum_{k=1}^K \lambda_k w_k, \quad (3.3a)$$

$$\text{s.t.} \sum_{k=1}^K b_k^n w_k \leq 1, \quad \text{for all } n \in N \quad (3.3b)$$

$$w_k \in \{0, 1\}, \quad \text{for all } k \in K \quad (3.3c)$$

$$w^{\leftarrow} - w^{\rightarrow} = 0, \quad \text{for all symmetry pairs } (w^{\leftarrow}, w^{\rightarrow}) \quad (3.3d)$$

To increase the plausibility of the automatically computed reference expressions (Goal 4), we include symmetry constraints (Equation 3.3d). We observe from facial performance acting that it is easier to pose symmetric facial expressions (e.g., both eyes closed, both eyebrows frowning) than asymmetric facial expressions (e.g., left mouth smiling, right mouth sad). We enforce simultaneous activation of symmetric blendshapes, where $(w^{\leftarrow}, w^{\rightarrow})$ are the blendshape weights of two symmetric blendshapes $(v^{\leftarrow}, v^{\rightarrow})$. Section 3.2.2 provides more details on defining symmetric blendshape pairs.

Symmetry pairs and most importance weights are pre-computed once at the beginning. In contrast, Equation 3.3 is solved for every single reference expression and returns the blendshape weights w_k^j for computing the reference expression \mathbf{x}_j . All blendshapes with $w_k = 1$ are removed from the set of available blendshapes \mathcal{K} and the total number of blendshapes K is updated. The process is repeated until either a user-defined maximum number of expressions is reached or no blendshapes remain. The order of the computed reference expressions reflects their power to improve the overall result (Goal 3).

Importance Weighting of Blendshapes

The binary blendshape provides information about the activated vertices, but all blendshapes are considered as equally important if $\lambda_k = 1$. However, personalizing certain blendshapes is more important than others, which we model by computing an importance weight λ_k for every blendshape k in Equation 3.3. Let us analyze the origin of errors using example-based blendshape transfer techniques. First, semantically equivalent expressions can have individual variations, e.g., strong dynamic wrinkles in faces of old people versus barely visible wrinkles in young faces. While it is difficult to forecast exactly the individual differences of expressions, we can assume that subtle expressions remain subtle and expressions with strong movements have a higher chance to introduce noticeable individual differences due to strong deformations. We model this observation numerically by the displacement strength λ_{Dis} .

Second, for blendshapes that are missing training examples, deformation transfer [SP04] might introduce errors. Deformation transfer is based on the assumption that the shape of the template rig \mathcal{A} is similar to the personalised rig \mathcal{B} ($\mathbf{v}_0^A \approx \mathbf{v}_0^B$). As soon this assumption is violated, artifacts start to appear. Closer examination reveals that if

triangles are non-uniformly scaled, results become inaccurate. Unfortunately, deformation gradients are only scale invariant in the tangential plane of a triangle [KG08] but not in the normal direction. Consequently, transferring the deformation between two shapes that are non-uniformly scaled, or where facial parts differ in their size relative to the head (e.g., big eyes vs. small eyes), will lead to visible artifacts. Notice as well that the non-similarity of meshes from the algorithm’s perspective is different to the perceptual difference of characters. Simple non-uniform scaling creates highly distinctive characters for the deformation transfer algorithm, but not for a human. At the same time modifying the amount of fat in a human face leads to visually distinct faces, but not numerically, because facial parts most affected by blendshapes (e.g., eyes, mouth) remain the same. We aim to prioritise blendshapes which would be problematic to transfer by calculating the triangle distortion λ_{TD} between two characters.

Finally, we suggest an optional weighting term for blendshape uniqueness to enforce distinctiveness of consecutive reference expressions. Although a large variation of expressions may appear advantageous in the first place, we observe that example based facial rigging [LWP10] tends to remove subtle blendshape differences, which we deemed important.

We observe that high-end character models with high numbers of blendshapes intentionally have subtle differences in their expressions (e.g., several versions of smiles), which is important to achieve high levels of realism. Blendshape transfer should maintain these subtle differences for high quality results. If, for example, we had 2 similar blendshapes in the model (e.g., evil and happy smile) with only one training example (e.g., happy smile) we lose nuances between the two smiles after blendshape transfer using the original alternating optimization (expression fitting and weight estimation) [LWP10]. Due to regularization, the weight estimation changes correct blendshape

weights 0.0,1.0 to an incorrect e.g. 0.3,0.7. Enforcing correct blendshape weights 0.0,1.0 creates a generic evil smile, but more importantly an accurate fit of the happy smile. Since neither situation is sufficient for the transfer of a high-end model, we consider the uniqueness term as optional (Figure 3.9).

Each of these factors will be normalised and combined into a single factor, subject to user-defined variables (α, β) . For our purposes we define the weights as $\alpha = 0.5, \beta = 0.0$ unless otherwise defined.

$$\lambda_k(\alpha, \beta) = \alpha \lambda_{Dis_k} + (1 - \alpha) \lambda_{TD_k} + \beta \lambda_{Dist_k} \quad (3.4)$$

Displacement Strength We consider the displacement weight of each blendshape, defined as the mean displacement of all vertices n between the blendshape \mathbf{v}_k and the neutral expression \mathbf{v}_0 , visualised in Figure 3.10. We consider only active vertices, as defined for binary blendshape \mathbf{b}_k in Section 3.2.2. Notice that the dot product of $\mathbf{b}_k \cdot \mathbf{b}_k$ returns the number of active vertices. All mean displacements \bar{d}_k are normalised to guarantee that λ_{Dis_k} is of range $[0, 1]$.

$$\begin{aligned} \bar{d}_k &= \frac{1}{\mathbf{b}_k \cdot \mathbf{b}_k} \sum_{n=1}^N b_k^n \|\mathbf{v}_k^n - \mathbf{v}_0^n\| \\ \lambda_{Dis_k} &= \frac{\bar{d}_k}{\max\{\bar{d}_1, \dots, \bar{d}_K\}} \end{aligned} \quad (3.5)$$

Triangle Distortion We consider the triangle distortion (as visualised in Figure 3.10) between the neutral expressions of the template rig \mathcal{A} and target character \mathcal{B} , since the relationship between the triangles of these two meshes is key for accuracy of deformation transfer. Given the equal connectivity between two blendshape models,

we first compute the vectors \mathbf{vu} and \mathbf{vw} , for each pair of triangles \mathbf{uvw} of the meshes \mathcal{A} and \mathcal{B} . The final distortion metric s between triangles t of the meshes \mathcal{A} and \mathcal{B} is then:

$$s^t = \max \left\{ \frac{\mathbf{vu}_A}{\mathbf{vu}_B}, \frac{\mathbf{vu}_B}{\mathbf{vu}_A} \right\} + \max \left\{ \frac{\mathbf{vw}_A}{\mathbf{vw}_B}, \frac{\mathbf{vw}_B}{\mathbf{vw}_A} \right\} \quad (3.6)$$

The variable s for triangles t is of range $[1, \infty]$ and is normalised in Equation 3.7 to ensure that λ_{TD} and λ_{Dis} are both of range $[0, 1]$. Triangle distortion is only relevant for triangles affected by deformation within a blendshape. We multiply s^t , which is only based on the two neutral shapes, with the blendshape dependent b_k^t , which is 1 if at least one vertex of the triangle t is active and 0 otherwise.

$$s_k = \sum_{t=1}^T b_k^t s^t, \quad b_k^t = b_k^u \cup b_k^v \cup b_k^w \quad (3.7)$$

$$\lambda_{TD_k} = \frac{s_k}{\max \{s_1, \dots, s_K\}}$$

Blendshape Uniqueness The Pearson coefficient quantifies similarity between two blendshapes [LAR⁺14, BiRZL⁺17]. The codomain of the Pearson coefficient is in the range $[-1, 1]$, with 1 indicating two blendshapes are the same, 0 indicating they are different, and -1 indicating they are the same but in opposing directions (e.g. opening and closing of the mouth). As we wish to consider opposite movements as different, we change all negative coefficients to 0. The first reference expression is unique by definition such that $\lambda_{U_k} = 0$. All blendshapes that are part of a reference expression are saved in the set \mathcal{R} , with r being the index number and R the total number of blendshapes in \mathcal{R} . All other blendshapes remain in the set \mathcal{K} , with K being now the available blendshapes for packing and not the overall number of blendshapes anymore. To compute the uniqueness weight, we first sum the Pearson coefficient between a

blendshape k and all blendshapes in \mathcal{R} . The result is normalised and inverted. The intuition behind this metric is, if no similar blendshape is part of the set of reference expressions the uniqueness weight is high, otherwise it is low.

$$\begin{aligned}
 U_k &= \frac{1}{R} \sum_{r=1}^R \max \left\{ \frac{\delta \mathbf{v}_k \cdot \delta \mathbf{v}_r}{\|\delta \mathbf{v}_k\| \|\delta \mathbf{v}_r\|}, 0 \right\}, \text{ if } R > 0 \\
 \lambda_{U_k} &= 1 - \frac{U_k}{\max \{U_1, \dots, U_K\}}
 \end{aligned} \tag{3.8}$$

Symmetry Constraints

Our overall goal is to propose reference expressions that are meaningful and have good numerical properties. To our knowledge, no method exists that can determine whether an expression can be posed by a human. Interestingly, even well trained actors can have difficulty in posing all defined Action Units individually [CKH11]. We discarded the idea of learning plausible expressions from animation data, because this would require the creation of an expressive animation sequence for every template rig, a time-consuming and difficult task.

Furthermore, plausible expressions that are not part of the animation would be considered as infeasible, unnecessarily limiting the combination space of blendshapes.

We preferred a data-free solution that largely ensures plausible expressions in combination with an optional artist friendly refinement feature (Section 3.2.2). This enables the user to more quickly accomplish the task of creating poseable expressions. Even with manual refinement, the workflow remains largely automated.

We observe that symmetric facial expressions are easier to perform and appear more frequently, e.g. closing eyes or lifting eyebrows simultaneously, a smile on both sides.

We want to prefer such combinations because they strongly improve the likelihood to create visually plausible expressions (Goal 4). We identify two different types of facial symmetry: shape symmetry and motion symmetry (Figure 3.11). Expressions like closing both eyes, a smile etc., create shapes that are symmetric across the left and right side of the face. Face shapes are only symmetric when mirrored on the left and right sides. Nevertheless, the upper and lower lip move in coordination and certain symmetry exists between the main motion directions. The same applies for the lower eyelid and the upper eyelid together with the eyebrows. In the following, we aim to automatically identify blendshape pairs that contain symmetric activation of the face (e.g. a left eyebrow raise blendshape and a right eyebrow raise blendshape).

Shape Symmetry For shape symmetry, we require a one-to-one mapping between the vertices on the left and right-hand side of the face. This mapping can be computed automatically, either based on the mesh topology [Aut16], where a user defines a single triangle edge as the symmetry axis, or independent of the mesh topology [MGP07] by comparing different samples of the mesh surface. For all our tested blendshape models, using symmetry detection based on topology was sufficient. Once the symmetry mapping is computed on the neutral mesh, our algorithm first automatically detects all self-symmetric blendshapes, (e.g., a smile that affects both sides of the face) and excludes them from the symmetry search. Intuitively, one might also consider excluding all blendshapes that have active vertices on both sides of the symmetry axis. However, this exclusion is too broad. Many blendshapes for the mouth and eyebrows have active vertices on both sides of the symmetry axis in combination with a symmetric opponent (see Figure 3.8).

For the remaining non-self-symmetric blendshapes, we project the displacements

of blendshape m on the symmetry plane using the projection function $p(\delta\mathbf{v}_m^n)$ and compute the final difference between the potentially symmetric blendshape k and the projected blendshape $p(m)$ using the following equation:

$$\epsilon_{km} = \frac{1}{\sum_{n=1}^N (\mathbf{b}_k^n \cap p(\mathbf{b}_m^n))} \sum_{n=1}^N \|\mathbf{b}_k^n \delta\mathbf{v}_k^n - \mathbf{b}_m^n p(\delta\mathbf{v}_m^n)\| \quad (3.9)$$

Notice that the intersection of \mathbf{b}_k^n and $p(\mathbf{b}_m^n)$ is smaller than the union. While normalizing by the union would compute the average vertex displacement between \mathbf{b}_k^n and $p(\mathbf{b}_m^n)$, normalizing by the intersection amplifies ϵ_{km} for non-symmetric blendshapes. If ϵ_{km} is below the following relative threshold, the blendshapes k and m are considered as symmetric.

$$\epsilon_{km} < 0.4 \min_{n \in N} \{\max \|\delta\mathbf{v}_k^n\|, \max \|\delta\mathbf{v}_m^n\|\} \quad (3.10)$$

When multiple symmetric blendshapes are found, we take the pair with the lowest error and check for possible overlapping of active vertices. We allow small overlap between symmetric pairs, as we have found that pairs of blendshapes that should be considered symmetric often have an overlap at the axis of symmetry (e.g., Figure 3.8, left). Blendshapes that spread along both facial halves will be not combined by the symmetry constraint. As our optimization algorithm strictly prevents any overlap between blendshapes in an expression, active vertices must be corrected for symmetric pairs of blendshapes in advance. Symmetric blendshapes will be forced to be included together, therefore the total active area remains the same.

Motion Symmetry Finding a reliable mapping between vertices of the upper mouth and nose area, and the lower mouth and chin area is difficult because facial parts differ significantly in terms of shape for the neutral pose and in terms of displacement for

the blendshapes.

Lacking a promising automatic method, we use a sketch-based method, where the user selects the vertices of the two pairs of symmetric areas. As an approximate selection is sufficient, we rely on Maya’s built-in mesh painting interface in our implementation. This allows the entire task to be accomplished within a minute. Blendshapes that either have no selected vertices or selected vertices from more than one sketch, are removed from the set of possible candidates for motion symmetry. The remaining blendshapes have very local deformations, a small number of active vertices and a dominant displacement direction. Lacking a reliable one-to-one mapping approach between individual vertices, we cannot compare per-vertex displacements as we did for shape symmetry. Instead, for every blendshape we compute an average displacement direction $\delta\bar{\mathbf{v}}_k$.

$$\delta\bar{\mathbf{v}}_k = \frac{1}{\left\| \sum_{n=1}^N \mathbf{b}_k^n \delta\mathbf{v}_k^n \right\|} \sum_{n=1}^N \mathbf{b}_k^n \delta\mathbf{v}_k^n \quad (3.11)$$

We then compare the average blendshape displacements between the potential symmetry pair candidates k and m ,

$$\cos(\phi_{km}) = \delta\bar{\mathbf{v}}_k \cdot \delta\bar{\mathbf{v}}_m. \quad (3.12)$$

We consider all blendshapes with $\cos(\phi_{km})$ below a threshold of -0.985 as symmetric. The minimal threshold is equivalent to 10° difference in the opposite direction (asymmetric activation) and symmetry pairs are built based on the smallest angle. As previously, active vertices are modified to prevent possible overlapping.

| Test-case | N | K | Optimal | Greedy |
|------------|-------|-----|---------|--------|
| Male | 7366 | 73 | 4.97s | 0.29s |
| Toon | 11680 | 65 | 7.14s | 0.39s |
| Female | 5034 | 265 | 13.14s | 0.39s |
| DigiDouble | 5131 | 161 | 13.14s | 0.39s |

Table 3.4: Timings for computing the first reference expression. Computation time depends on the vertex number N and the number of blendshapes K in the blendshape rig.

| Test-case | 5 ex. | 10 ex. | 20 ex. |
|------------|------------|------------|---------------|
| Male | 36/38 (31) | 49/49 (42) | 63/63 (56) |
| Toon | 34/33 (22) | 48/49 (43) | 62/62 (61) |
| Female | 58/54 (34) | 89/85 (57) | 124/116 (93) |
| DigiDouble | 61/54 (39) | 90/79 (55) | 126/121 (104) |

Table 3.5: Number of blendshapes selected by greedy and optimal algorithms and their overlap in the first 5, 10 and 20 training expressions. Format: Greedy/Optimal (Overlap).

Solving the Optimization Problem

Solving a set packing problem (Equation 3.3) is NP-hard. However, by formulating the problem as an integer linear problem and using existing numerical libraries, the optimal solution can be computed within a reasonable time (Table 3.4). Alternatively, greedy approximation methods facilitate interactive framerates ($< 1s$) and sufficient results.

In the following, we discuss the details together with the advantages and disadvantages.

Optimal Solution Our optimization problem (Equation 3.3) is a derivation of the weighted set packing problem, which is a classic example of an NP-complete problem [Kar72]. We formulate it as a boolean linear programming problem, which is a special case of mixed integer linear programming (MILP) due to the linear objective function

together with linear (in-)equality constraints and $w_k \in \{0, 1\}$. We compute the optimal solution in Python using the PuLP library in combination with the numerical library CPLEX. The linear objective function consists of K unknowns, one for each blendshape. The number of linear constraints is $N + P$ which is the number of vertices and symmetry pairs ($w^\leftarrow, w^\rightarrow$). Various timings for solving the linear programming problem are listed in Table 3.4.

The advantage of the optimal solution is that it computes the global optimum, e.g. by using a branch and cut algorithm. Computation time depends on the number of vertices, the number of blendshapes and the distribution of activated vertices across different blendshapes. For example, if blendshapes either activate all vertices in the upper or lower face halves, the number of combinations to test is less than if different blendshapes activate vertices at different locations. This blendshape-specific component makes it difficult to predict timings exactly for different blendshape rigs. Overall, we notice that the optimal solution is slower compared to the greedy solution.

Greedy Approximation We implement Kordalewski’s greedy set packing algorithm [Kor13] with some alterations to suit our method as shown in Algorithm 1. First, we select blendshapes based on the smallest weighted sum of active vertices, where the importance weight λ_n is inverted. If the weight is constant across all blendshapes and vertices, this fits the largest number of blendshapes into one expression. Second, we include an option to allow a small percentage of overlap between blendshapes, which is not possible in case of the optimal solution.

Artist Friendly Refinement After incorporating the symmetry constraint and fine-tuning the importance weights, the suggested expressions by our algorithm are

Algorithm 1 Greedy Set Packing

```
 $\mathbf{x}_j = \mathbf{v}_0$  ▷ example expression  
 $\mathcal{K} = \{1, \dots, K\}$  ▷ non-covered blendshape indices  
 $\lambda_{max} = 1$  ▷ maximum possible weight  
for  $k$  in  $\mathcal{K}$  do  
     $n_k = \mathbf{b}_k \cdot \mathbf{b}_k$  ▷ sum active vertices  
     $\lambda_k n_k = (\lambda_{max} - \lambda_k) n_k$  ▷ precompute importance  
while  $\mathcal{K} \neq \emptyset$  do ▷ expression packing  
     $\delta \mathbf{v}^\perp = \text{smallest}(\lambda_k n_k)$   
     $\delta \mathbf{v}^\parallel = \text{find\_symmetric}(\delta \mathbf{v}^\perp)$   
     $\mathbf{x}_j += \delta \mathbf{v}^\perp + \delta \mathbf{v}^\parallel$  ▷ add blendshapes  
     $\mathcal{K} = \mathcal{K} \setminus \{k^\perp, k^\parallel\}$  ▷ remove covered blendshapes  
    for  $m$  in  $\mathcal{K}$  do ▷ remove overlapping blendshapes  
        if  $\text{overlap}(\mathbf{x}_j, \mathbf{b}_m)$  then  
             $\mathcal{K} = \mathcal{K} \setminus \{m\}$   
return  $\mathbf{x}_j$ 
```

plausible in about 85% of cases (see Section 3.2.3). To correct remaining implausible expressions but still maintain the optimal selection of blendshapes, we allow for artist intervention in our greedy algorithm.

As each blendshape is chosen, the user is presented with the options to accept or reject it. If blendshapes are rejected, the algorithm proposes alternative nearly optimal blendshapes and the process is repeated. All rejected blendshapes are no longer considered for the current expression, but will be reconsidered when computing the next expression.

3.2.3 Results and Evaluation

In this section, we evaluate the influence of different parameters and solvers on the final output and the pose-ability of the expressions. For evaluation, we use a derivation of the original example-based blendshape transfer method [LWP10]. The method required

a number of training expressions in combination with estimates of the blendshape weights. By design, the blendshape weights are 0 or 1 and are known for our training expressions. We therefore remove the weight estimation step within the alternating optimization (fitting towards examples vs. blendshape weight estimation).

Dataset

One intrinsic challenge of evaluating example-based blendshape transfer methods are suitable character assets. As we discussed in Section 3.1, using lower quality meshes from a convenient database led to difficulty in drawing conclusions from our results. In order to avoid that, we focus on obtaining high quality test cases that mimic real industry use cases for blendshape transfer.

For ground truth comparisons, two blendshape models \mathcal{A} and \mathcal{B} are required of equal vertex connectivity and semantically equivalent blendshapes. In practice, either the total number of blendshapes is small [CWZ⁺14] or a small number of blendshapes is identical between two characters, while other blendshapes can range from similar to completely different. This is even the case in recent datasets, e.g., 3D FACS and FLAME [CKH11, LBB⁺17]. Furthermore, evaluation should cover difficult cases, meaning that the two blendshape models should have significantly different proportions in areas that are deformed most by blendshapes (Section 3.2.1).

As a ground-truth test for the digital double use-case, we acquired two high-end photogrammetry-scanned characters, created by Eisko, a leading Digital Double company (Figure 3.12). We refer to these characters as Female 1 and Female 2. Semantically equivalent blendshapes between both models were selected manually. Non-equivalent blendshapes were removed. This resulted in a total of 161 in-correspondence blendshapes between Female 1 and Female 2. To establish equal connectivity between

the two high-quality facial rigs we followed the default procedure as described previously (Figure 3.7). We refer to this test-case as DigiDouble. Because the two rigs are based on real people and 3D scans, this is the closest possible approximation to a real use-case that offers ground truth comparisons. In such a dataset, the blendshape weights of the reference expressions can be copy-pasted to obtain training expressions. After running the example-based blendshape transfer algorithm with only four training expressions, we compare the individual blendshapes for character \mathcal{B} with the original (see Figure 3.12, right columns). Despite a compact packing of over 30 blendshapes in a few expressions visible artifacts that appeared when using deformation transfer only, are removed and only small numerical errors appear where blendshapes overlapped. A larger set of results can be found in Figure 3.19.

To provide more test-cases, especially of very challenging scenarios, we first gathered 3 production-quality rigs: a Male, Female (Female 1 from before), and a Toon (see Table 3.4 and Figure 3.18). The characters ranged in vertex count (5034 to 11,680), number of blendshapes (from 65 to 265), and whether they were sculpted by hand (Toon and Male) or high-end photogrammetry-scanned (Female). Since it is difficult to obtain rigs with semantically identical blendshapes, we generated altered versions of our rigs, by applying a set of deformations to the entire blendshape model. This includes non-uniform scaling as well as local enlarging and shrinking of facial parts like eyes and mouth using free-form deformation. The original and altered characters can be seen in Figure 3.18, first column. For example, the third row shows the original Toon character, as well as the Toon character with non-uniform scaling applied in order to create significantly different geometry that would be a suitable test case. These alterations were specifically designed to generate proportionally non-similar character pairs, which are difficult cases for deformation transfer. To simplify naming, test-cases

using the original character and warped version are named as the original character (Toon, Male, Female).

Generic Optimization Parameters

Our proposed optimization has a set of user-parameters to refine the computation of reference expressions. To avoid unnecessary parameter tweaking, we investigate suitable default parameters that create good results for all test cases.

In general, we aimed in previous sections for relative metrics in order to be scale independent and added normalization to be more stable across character rigs with different numbers of vertices or blendshapes.

Binary Blendshapes Threshold A relative threshold (Equation 3.2) turned out to be the best compromise for defining active vertices for both subtle and expressive blendshapes. For evaluation, we tested different values of this threshold μ on characters with few and many blendshapes. During the evaluation we set $\lambda_k = 1.0$ in Equation 3.3. Increasing μ increases the number of blendshapes within one reference expression and decreases the total number of expressions. At the same time, a high value for $\mu > 0.4$ created frequently unfeasible expressions for blendshape models with $K \approx 50$ and could drop to $\mu > 0.3$ for blendshape models with $K \approx 265$ (Figure 3.13). During parameter testing of the maximum overlap between two blendshapes, we also noticed that a higher overlap creates a bigger error during the reconstruction of individual blendshapes from training expressions. We therefore recommend $0.25 \leq \mu \leq 0.4$ as a good trade-off between the smallest number of examples and plausible examples. In the remaining evaluation, we use $\mu = 0.3$. An example of a combined expression with its constituent blendshapes can be found in Figure 3.16.

Importance Weighting For evaluation of the importance weights, we linearly interpolate between different importance weights: $\lambda = \alpha\lambda_{Dis} + (1.0 - \alpha)\lambda_{TD}$. Symmetry constraints are removed during numerical comparisons to avoid any bias. Remember that a small number of vertices were allowed to overlap for symmetric blendshapes. As a measure of improvement, we compute the root mean squared distance between the ground truth blendshapes of the target model and the output of an example-based blendshape transfer algorithm (with training expressions as input). As intended, adding importance weights decreases the error (Figure 3.15). Weighting both importance values equally ($\alpha = 0.5$) was found to be the best trade-off across different characters and numbers of training expressions. This confirms that both importance metrics are relevant.

Greedy vs Optimal Finally, we compare the reference expressions computed by the optimal and the greedy solver (Section 3.2.2). Between 67% and 98% of blendshapes selected by the optimal method have also been selected by the greedy solver (Table 3.5), meaning that both solvers compute similar, but not identical results. The greedy algorithm creates reference expressions with slightly more blendshapes due to the permission of little overlap between active vertices and is nearly 20 times faster (Table 3.4). In contrast, blendshape transfer in combination with training expressions of the optimal solver creates blendshapes that are closer to ground-truth in terms of RMS-error (See Figure 3.17). With only 20 training expressions, both methods covered approximately 67% blendshapes for Female, 86% for Male and 95% for Toon, which is a remarkable reduction of training expressions. Interestingly, adding the symmetry constraint to the optimal solution had basically no influence on the measured RMS-error, while the RMS-error increased for the greedy algorithm with active symmetry

constraints.

Expression Pose-ability

Good reference expressions should not only show good numerical properties, but should be poseable by an actor, to facilitate scanning. The general unweighted set packing algorithm creates reference expressions that might be difficult or even impossible to pose (Figure 3.18, left). Our results showed that adding importance weighting had a positive side-effect on how poseable the expressions are (Figure 3.18, middle). Blendshapes with strong displacements, which coincidentally tend to have more active vertices, started to be part of the first reference expressions. Rather than packing several subtle expression in the first reference expressions, subtle and expressive blendshapes were distributed more evenly. Some reference expressions even appeared symmetric, due to the fact that some template rigs were perfectly symmetric, and therefore left and right blendshapes had equal importance factors. If these blendshapes do not overlap, then they are included at the same time, thus creating already symmetric expressions.

To create more plausible expressions automatically, we introduced the symmetry constraint (Figure 3.18, right). To evaluate the final output of our Expression Packing algorithm, we recruited two volunteers to demonstrate posing of the first ten reference expressions of the DigiDouble case. The first expressions are the most tightly packed and therefore the most difficult to pose. Some expressions were slightly modified because the symmetry constraint at times forced the eyes to look in opposite directions. After editing, both eyes looked in the same direction and both volunteers were able to pose all ten expressions within short capturing sessions (Figure 3.14).

3.2.4 Discussion

The main novelty of our work is the adaptation of the weighted set packing algorithm to the problem of finding optimal reference expressions. The automatically created reference expressions are largely plausible even for challenging high-quality rigs consisting of over 250 blendshapes. Furthermore, our artist-friendly refinement method gives the user full control over the created reference expressions, without the need to track minimal overlap of blendshapes or estimate the influence on the optimization for a blendshape.

In this chapter, we proposed a solution for automatic reference expression creation, allowing for the creation of optimal training expressions for example-based blendshape transfer. This is achieved through constrained blendshape packing. Our key contributions include blendshape importance ordering specifically aiming to deal with the weaknesses of the deformation transfer algorithm, and the use of the weighted set packing algorithm for creation of information-dense reference expressions. Combined, these produce as-few-as-possible packed reference expressions, tailored to the characters and blendshapes to which the blendshape transfer will be applied. Using the recommended parameters, our algorithm achieves a remarkable density of blendshape information. With 20 examples, our results showed that almost the full set of blendshapes of our test rigs were covered and 67% of the blendshapes of our highly-expressive production rig.

In the following chapter, we continue investigating the idea of optimising the pipeline using blendshape importance, but focusing our attention on the animation stage of the pipeline, where having large numbers of shapes to blend at run-time is a major bottleneck, and one of the reasons why many games favour bone-based facial animation

for real-time. We propose a novel GPU blendshape animation method allowing high-quality characters with large blendshape sets to be animated on the GPU in real-time. We also propose a blendshape reduction level-of-detail technique, which uses the idea of blendshape importance at its core. By ordering blendshapes and removing only those deemed least important, the impact on the perception of the optimised animation is reduced.

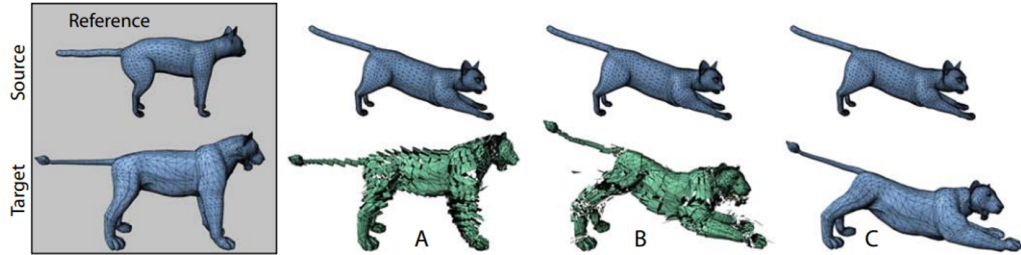


Figure 3.6: Deformation Transfer [SP04] showing the output with (A) only the non-translational transformations, (B) all transformations, and (C) Deformation Transfer, which solves the optimisation problem balancing the non-translational transformations with new translations such that vertex connectivity is maintained.

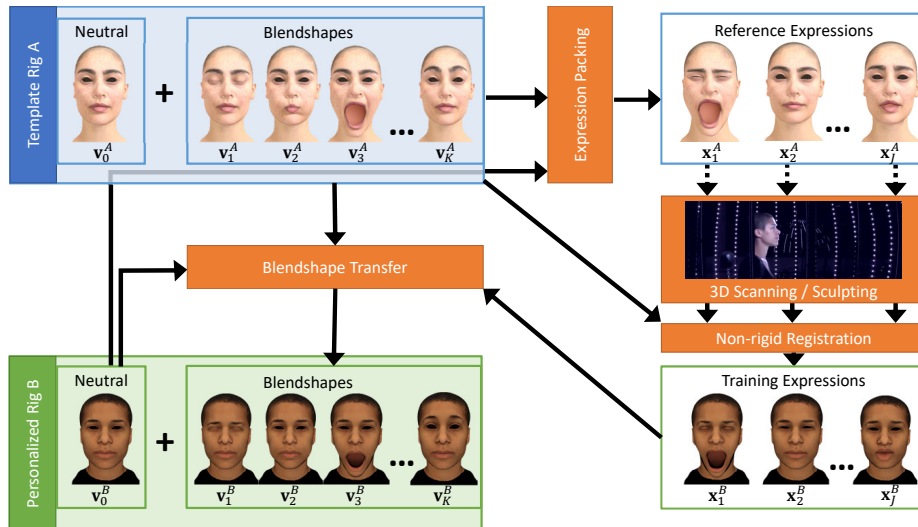


Figure 3.7: Illustration of our expression packing algorithm as part of the digital double pipeline. We run our expression packing algorithm to create as-few-as-possible reference expressions, which help to guide an actor during a scanning session. Using non-rigid registration, e.g. [IBP15, FNH⁺17, LBB⁺17], the mesh topology of the template rig is transferred to the 3D scans. The resulting training expressions are optimal for the blendshape transfer algorithm, since they lead to best results with as-few-as-possible expressions. Finally, the blendshapes are computed for the personalised rig.

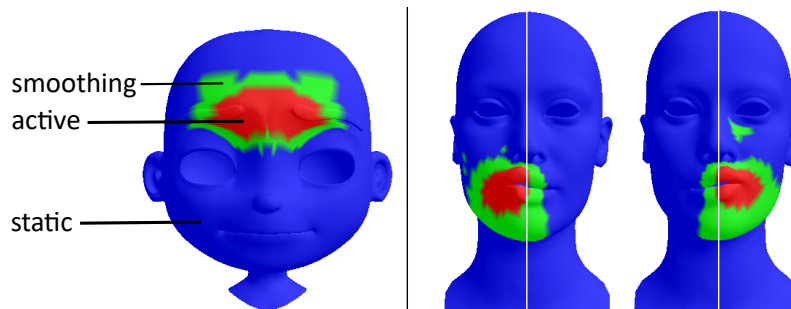


Figure 3.8: Illustration of the different types of vertices for one blendshape: static (blue), smoothing (green) and active (red). The blendshape on the left is self-symmetric, the two blendshapes on the right are symmetric, but have overlapping active vertices.

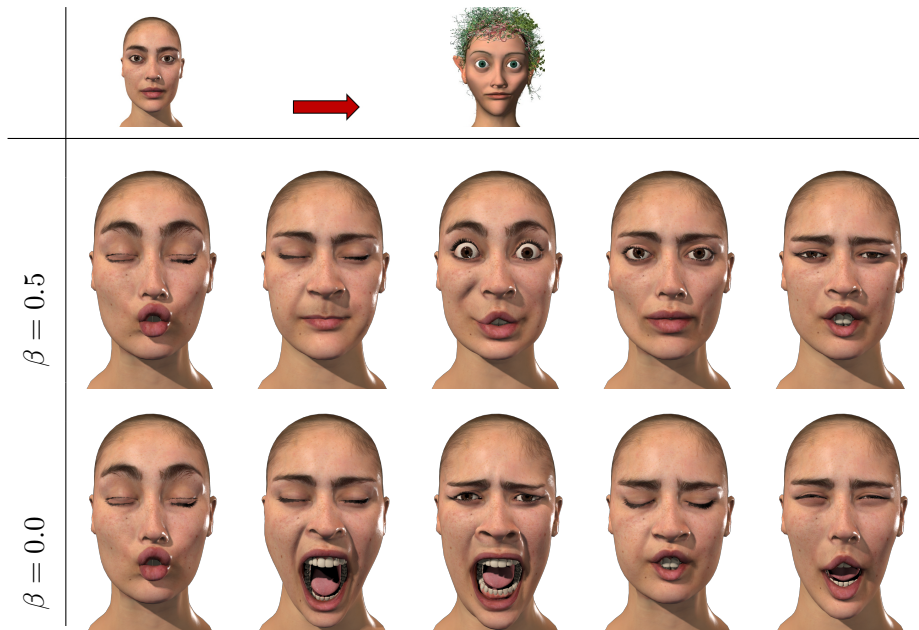


Figure 3.9: First five expressions computed with ($\beta = 0.5$) and without ($\beta = 0.0$) the optional blendshape uniqueness metric.

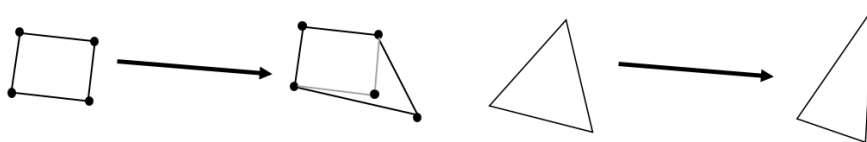


Figure 3.10: Visualisation of Displacement Strength (left) and Triangle Distortion (right).

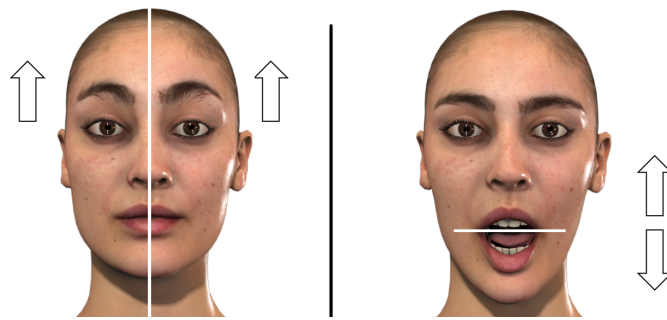


Figure 3.11: Symmetric blendshape activation. Shape symmetry remains while lifting both eyebrows (left) and motion symmetry is created due to perpendicular movement of the lips (right).

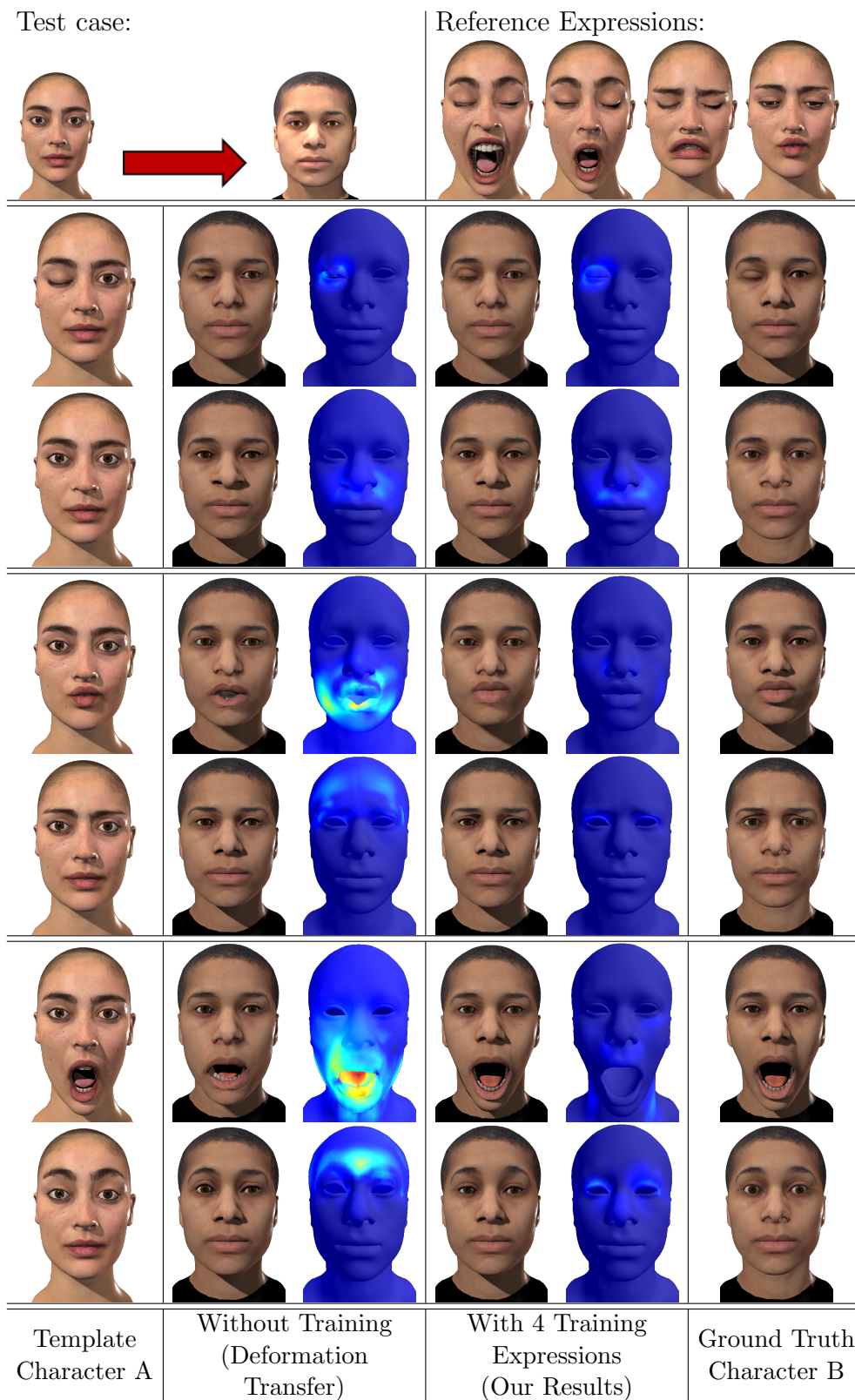


Figure 3.12: Comparison of individual blendshapes after applying blendshape transfer without training expressions (Deformation Transfer) and with only four training expressions based on our reference expressions. Each row shows two blendshapes that have been corrected by one training expression.



Figure 3.13: Output from expression packing algorithm for our Female test-case: the first four computed reference expressions using different thresholds for the definition of binary blendshapes. A threshold of up to $\mu = 0.3$ creates plausible results in most cases even for model with many blendshapes ($K = 265$).


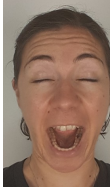
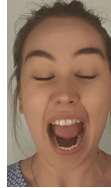

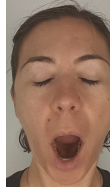
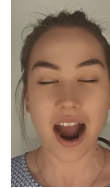

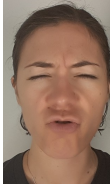
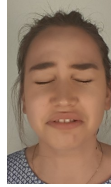

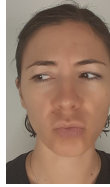
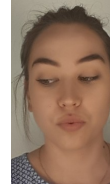

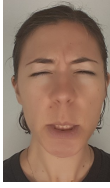
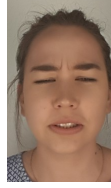

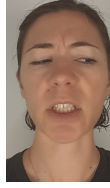
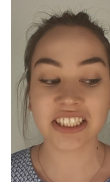

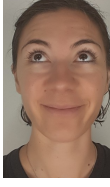
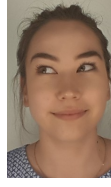

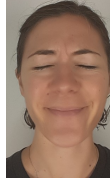


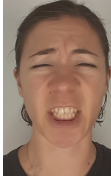



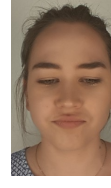
| $\#\delta\mathbf{v}_k$ | | | | | | | $\#\delta\mathbf{v}_k$ |
|------------------------|---|---|---|---|--|---|------------------------|
| 12 |  |  |  |  |  |  | 7 |
| 8 |  |  |  |  |  |  | 7 |
| 6 |  |  |  |  |  |  | 7 |
| 6 |  |  |  |  |  |  | 3 |
| 2 |  |  |  |  |  |  | 6 |

Figure 3.14: Two volunteers demonstrating the pose-ability of the DigiDouble expressions. The number of blendshapes in each expression is shown on the left and right columns.

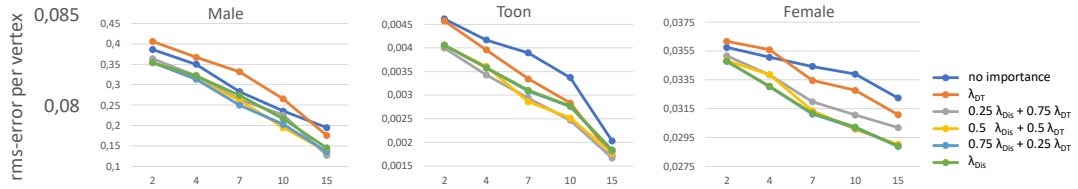


Figure 3.15: Average RMS-error per vertex between ground truth and the output of example-based blendshape transfer for 3 different test-cases. Number of training expressions ranges from 2 to 15, (shown on x-axis). Importance weighting that considers both, vertex displacement (λ_{Dis}) and triangle distortions (λ_{TD}), creates most accurate results across different characters and number of training expressions.



Figure 3.16: First packed reference expression (Figure 3.18) together with the nine individual blendshapes.

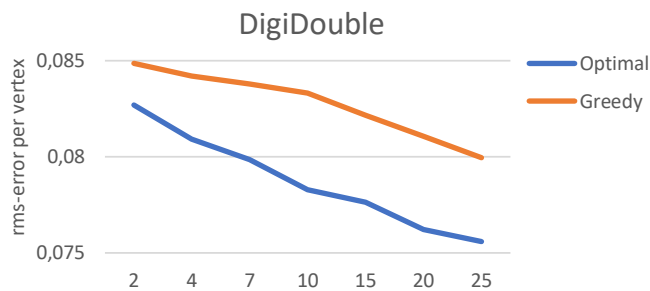


Figure 3.17: RMS-error for our DigiDouble test case ($k = 161$) with different numbers of training expressions as input (ranging from 2 to 25, shown on x-axis). Training expressions were computed with the optimal and greedy algorithms.










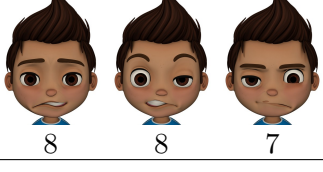
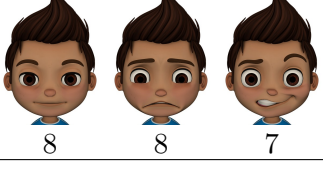
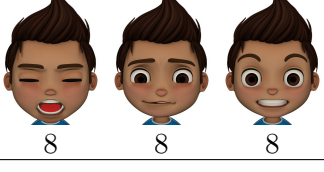

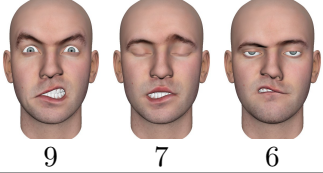
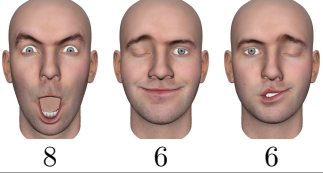
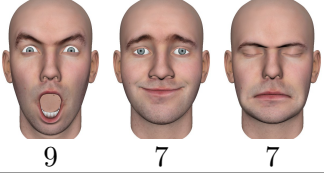
| | | | |
|---|---|--|--|
| DigiDouble  $\#\delta\mathbf{v}_k$ |  12 8 8 |  12 7 7 |  12 7 8 |
| Female  $\#\delta\mathbf{v}_k$ |  16 14 11 |  12 16 7 |  13 11 6 |
| Toon  $\#\delta\mathbf{v}_k$ |  8 8 7 |  8 8 7 |  8 8 8 |
| Male  $\#\delta\mathbf{v}_k$ |  9 7 6 |  8 6 6 |  9 7 7 |
| | No Constraints | Only Importance Weights | Importance Weights With Symmetry |

Figure 3.18: Output from our expression packing algorithm showing the first 3 computed reference expressions. The number below each is the number of blendshapes that have been packed into that reference expression. Expressions were computed (left) with no importance weighting or symmetry constraints, (middle) with importance weights only, and (right) with importance- weights and symmetry constraints.

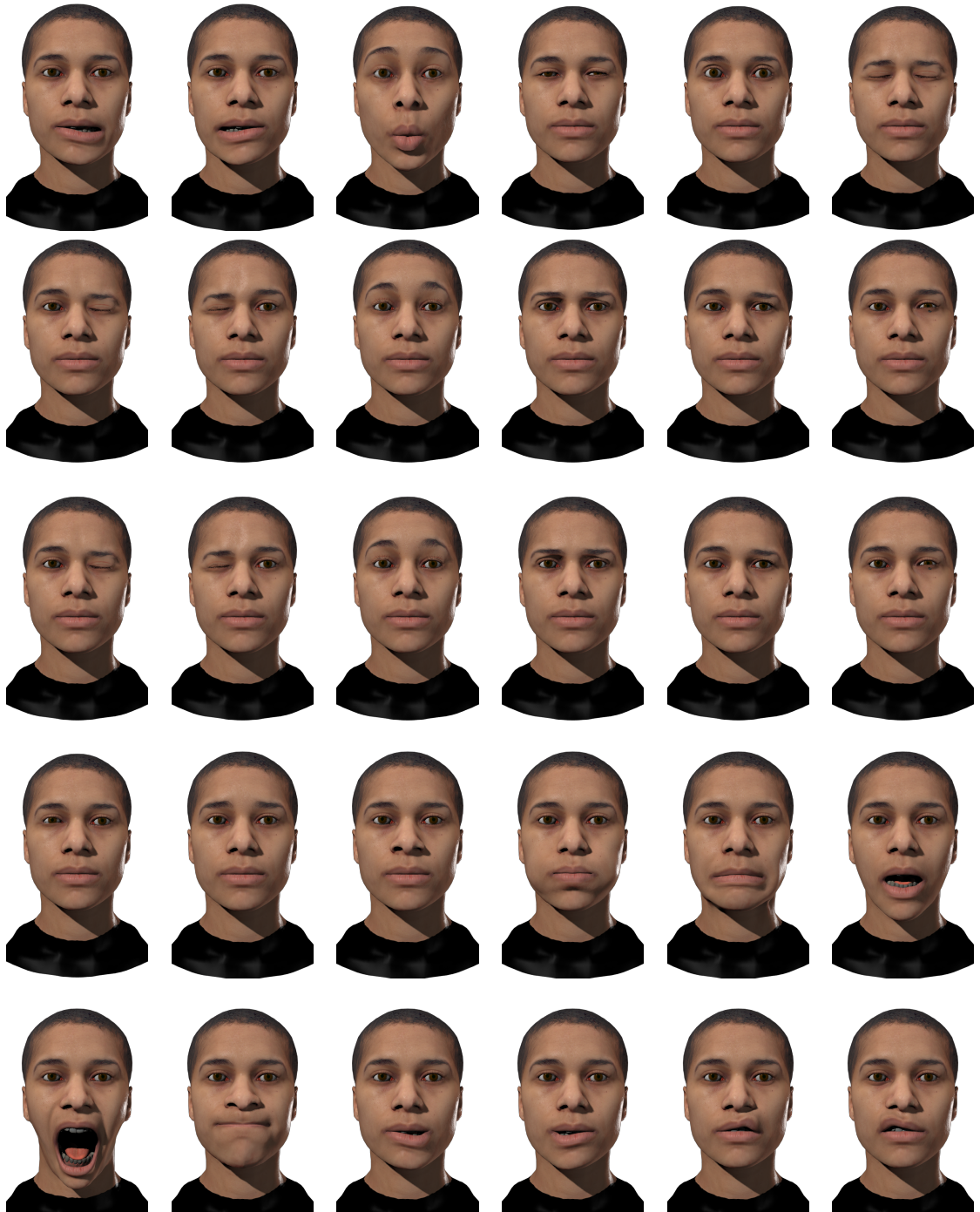


Figure 3.19: Selection of individual blendshapes after applying blendshape transfer in combination with a full set of training expressions, where every blendshape is present in one of the training expressions.

Chapter 4

Real-Time Optimisation

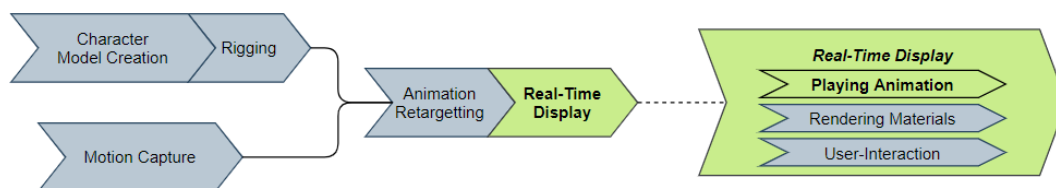


Figure 4.1: The facial animation pipeline with the real-time animation stage highlighted.

In this chapter we explore the animation stage of the pipeline (highlighted in Figure 4.1). For real-time applications, blendshape animations are usually calculated on the CPU, which is a slow process, and are therefore generally limited to only the closest level of detail for a small number of characters in a scene. We present a GPU based blendshape animation technique. By storing the blendshape model and animations on the GPU, we are able to attain significant speed improvements over CPU-based animation. We also find that by using compute shaders to decouple rendering and animation we can improve performance when rendering a crowd animation.

We also present a level of detail method for blendshapes, which is an importance-

based reduction of the number of blendshapes in an animation. Reducing the set of blendshape expressions improves performance, but at the cost of expressiveness. However, the quality impact can be minimised by selecting this subset carefully. We mention similar methods for blendshape reduction which are animation-specific, i.e. they remove blendshapes that are not activated during an animation, and briefly evaluate a number of non-animation-specific blendshape ordering techniques which could be generalisable across various tasks requiring blendshape importance ordering.

I collaborated as third author on the work in this chapter, alongside another PhD student and post-doctoral researcher from Trinity College Dublin. I was involved in the research discussions, data preparation and analysis; and presented the paper at Motion In Games 2016 in San Francisco.

4.1 GPU Blendshape Animation

In real-time rendering, facial animation is usually performed using linear blend skinning which can be readily accelerated using the Graphics Processing Unit (GPU). More realistic results can be achieved through blendshape animation, but at the cost of greater memory consumption and computational expense. Implementing blendshapes on the GPU is not straightforward, and until recently, was quite difficult due to limitations imposed by contemporary graphics hardware and drivers. As a result, most implementations have been on the CPU, such as the popular video-game *Ryse Son of Rome* by Crytek [EMH14] which used corrective blendshapes at close distances. Performance on the CPU is acceptable when rendering small numbers of characters, but with large crowds performance will suffer unless we move the computation over to the highly parallel GPU.

Our contributions are:

- A general purpose GPU (GPGPU) variation to shader-based blendshape animation using compute shaders and texture buffers, supporting very high complexity meshes.
- A system for discarding perceptually less important blendshapes to improve performance for crowds (example crowd shown in Figure 4.2).
- Comparative performance evaluations of CPU and GPU blendshape techniques.

We are interested in leveraging the power of commodity GPUs to allow blendshapes to be used at more than just the closest level of detail, and potentially for multiple characters in a scene. The challenges here are finding out how to arrange the data and animation processing that we need, such that it is congruent with the latest GPU



Figure 4.2: A real-time rendering of 40 blendshape animated heads with 72 blendshapes each using our GPU technique.

rendering pipelines, and then finding a system that suits our scene for discarding animation processing that is less perceptually important to our rendering.

We propose a GPU-based blendshape technique which exploits relatively recent advances in GPU technology such as Texture Buffer Objects (TBO) and compute shaders. TBOs allow arbitrarily large blocks of data to be passed to the shader. Due to the highly parallel nature of the GPU, we can expect significant performance improvements over CPU-based implementations. We investigate further potential speed improvements through the use of general purpose GPU programming in the form of compute shaders and by controlling the level of detail of animations by only processing a subset of the blendshape expressions. Compute shaders give us the advantages of using the GPU for general-purpose programming, which means we are able to compute tasks not necessarily part of the graphics pipeline, in the same fashion as CUDA and

OpenCL interfaces. Compute shaders offer the additional benefit of being built into a rendering API, so are able to transfer the results of computations over to the graphics rendering pipeline within the same workflow.

4.1.1 Method

On older GPU hardware, the most sensible technique for GPU-based blendshape animation would be to pass the expressions in the form of either uniform variables or vertex attributes [Lor07]. The number of uniforms or vertex attributes is quite limited however, and would likely be insufficient for detailed models with a large number of blendshapes. By using relatively recent features of the GPU we are able to overcome these limitations. Our basic method is similar to Larach [Lor07] but differs in how it handles the geometry and animation. The basic structure of our method can be seen in Algorithm 2.



Figure 4.3: The blendshapes are serialised and stacked together in the 1-dimensional Texture Buffer Object.

We are able to pass the 72 blendshapes of our model to the shader by packing them inside a TBO as in Figure 4.3. These are one dimensional textures which allow extremely large collections of arbitrary data [SA12]. The maximum size of the TBO is effectively only limited by the amount of GPU memory available. For our model, we require less than 40 MB of texture memory - comparable to two uncompressed 2k

textures (2048x2048x4 channels), to store all vertex blendshape and animation data. If we wanted to add normal, tangent or texture coordinate blendshapes to the TBO this would increase, but only linearly, and well within the limits of modern GPU memory, which is in the order of gigabytes.

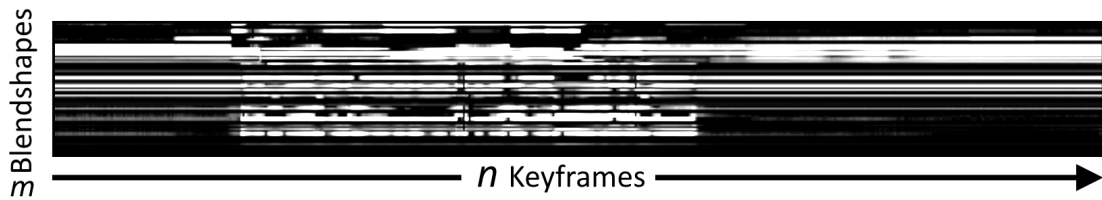


Figure 4.4: The blendshape animation encoded in a 32 bit floating-point texture. Each row contains the animated weight of a specific blendshape over time. We can see a group of blendshapes activate in the middle (bright white).

As the geometry resides on and is manipulated by the GPU, the graphics driver does not need to be stalled to transfer geometry to the video memory. In addition to storing the blendshapes on the GPU, using a similar technique to [Dud07], we also store the entire animation on the GPU as well. This was achieved by packing the animation values into a m by n texture, where m is the number of blendshapes, and n is the number of frames as in Figure 4.4. To ensure precision is preserved, a 32 bit floating-point texture is used. Texture components are usually represented using only 8 bits, which would severely quantise our animation data. Moving the animation onto the GPU removes the need to calculate and pass the animation weights to the shader every frame, the cost of which we find to be significant. Moving these calculations to the vertex shader further reduces the number of operations that might stall the GPU driver and frees the CPU for other purposes.

To reduce the number of iterations, and therefore improve performance, we have modified our blendshape shader to be able to return early, similar to the method of

Algorithm 2 Blendshape Pseudo-code

N_{Bs} = Blendshape Limit
 V_{Idx} = Current Vertex Index
 T_{Anim} = Animation Texture
 T_{Bs} = Blendshape TBO
 N_V = Number of Vertices per blendshape
 t = Current Frame
 $VertexPos$ = Initial neutral vertex position at index V_{Idx}

for $i = 0$ **to** $MAXBLENDSHAPES$ **do**
 $Weight = T_{Anim}[i,t]$
 if $i < N_{Bs}$ **then**
 $T_{Idx} = i * N_V * 3 + V_{Idx} * 3$ //TBO index
 $TempPos = (0,0,0)$
 $TempPos.x = T_{Bs}[T_{Idx}]$
 $TempPos.y = T_{Bs}[T_{Idx} + 1]$
 $TempPos.z = T_{Bs}[T_{Idx} + 2]$
 $VertexPos = VertexPos + TempPos * Weight$
 else
 Escape Loop

Lorach [Lor07]. This can be seen in Algorithm 2 where, inside the `for` loop, a conditional checks against the number of blendshapes, N_{Bs} . If this many iterations have been computed, then the shader returns early. This smaller number of blendshapes is specified by the user. This effectively trades animation quality for animation performance. As can be seen later in Section 4.1.4, this is necessary if we want practical real-time performance in large scenes. Such a system may appear initially unattractive, as a naïve implementation will result in a reduction in quality that can be severe. If, however, an importance order is established for each expression, the effect on quality can be minimised. This will be discussed in more detail in Section 4.1.2 and Chapter 5.

Compute Shader

A disadvantage of performing all blendshape related calculations on the GPU through the vertex shader, is that animation and rendering become inextricably linked. This is because everything now resides within the standard graphics rendering pipeline. This means that all calculations will need to be repeated every draw call whether they are required or not. A CPU-based solution does not share this limitation.

If we render multiple characters with different animation frames, repeating the calculations is not a problem but in a crowd rendering context this does limit us. This is because the perceived size of a crowd can be increased by interspersing duplicates of a small set of unique characters throughout the scene [MLD⁺08]. If we use our current GPU-based system there is no performance advantage to this technique.

It is possible though to decouple animation and rendering while still keeping all calculations on the GPU by using GPGPU programming in the form of compute shaders. These programs can be executed on the GPU, accessing all the same texture and vertex data as our normal method but operate independent of rendering. We are even able to

run the same method as Algorithm 2 with minimal alteration. Compute shaders were first added in OpenGL 4.3 [SA12], two years after DirectX 11.

CPU

For comparison, we also implement a basic CPU blendshape technique as most current implementations of blendshape animation are CPU-based. It should also be stated that CPU-based methods do have some advantages. They are invariably easier to implement and allow for a much simpler graphics pipeline which might be important if GPU rather than CPU time is at a premium.

If we are rendering small numbers of faces, real-time performance can easily be achieved on the CPU. For larger numbers of characters, the disadvantages become apparent. As the geometry must be recalculated on the CPU, it must also be passed to the GPU every frame which can be very costly as the GPU stalls while the transfer is underway.

4.1.2 Level of detail

As discussed in Section 4.1.1, our method, similar to Lorach [Lor07], allows the user to reduce the number of blendshapes being computed by the vertex shader, and in our case compute shader, to improve performance. Lorach only removed inactive blendshapes, ensuring the animations would be identical. Our method goes further and sorts the blendshapes in terms of importance. By removing less important expressions first, we can make much deeper performance improvements while minimising the effect on quality.

Ordering

We investigated using a number of general purpose metrics for ranking or ordering blendshapes by perceptual importance. Every metric was used to compare each blendshape expression to the neutral version of our model’s face, i.e. all blendshape weights set to zero. These metrics can roughly be classified as either geometry or image-based. The geometry-based metrics only consider the vertex positions of the various expressions. They are not influenced by any rendering information such as texture or lighting. In contrast, the image-based metrics only consider the final rendered image and are heavily influenced by texture and lighting.

Through an ad hoc assessment, we found that each metric has advantages and disadvantages. The results are broadly similar although there are differences for certain expressions particularly in the eyebrow region. All methods also give relatively low importance values to eye blinks. In general, we found that these metrics gave acceptable results for small to moderate blendshape reduction but some manual intervention was required for larger reductions.

Geometry-Based : Mean Squared Error MSE is a fairly simplistic metric which compares two objects on a vertex by vertex basis, based on the equation below:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2 \quad (4.1)$$

where n is the number of vertices, \hat{Y}_i is a neutral expression vertex, and Y_i is a vertex of the current expression being evaluated. For our model, we found that this method gave high importance values to mouth and lip expressions but low values to eye and brow expressions.

Geometry : Hausdorff Distance The Hausdorff distance is a more robust metric that is commonly used to compare the differences between two different meshes based on the following equation [ASCE02]:

$$H = \max_{a \in A} \{ \min_{b \in B} \{ d(a, b) \} \} \quad (4.2)$$

where A and B are the meshes being compared, a and b are vertices of their respective meshes and d is the Euclidean distance. Similar to the geometry-based MSE, we found this metric gave high values to the mouth region and lower values to the eye region. It gave higher values to the eyebrows though than most methods.

Image-Based : Mean Squared Error The image-based variant of mean squared error metric is the same as the geometric version although it operates on a pixel by pixel basis.

$$MSE = \frac{1}{wh} \sum_{y=1}^h \sum_{x=1}^w (I_a(x, y) - I_b(x, y))^2 \quad (4.3)$$

Where I_a and I_b represent two 2d input images with dimensions w and h , and pixel coordinates given by x and y , respectively. In our tests, this metric gave high importance values to the mouth and brow region of our model. Blinks were also not too low. This method can be a little too sensitive though. For example, minor asymmetric differences between the left and right of the face texture can result in different importance values for otherwise identical expressions.

Image-Based : Structural Similarity Index The structural similarity index metric (SSIM) tries to model the response of the human vision system [WBSS04]. We found this method gives identical importance values to asymmetric expressions unlike

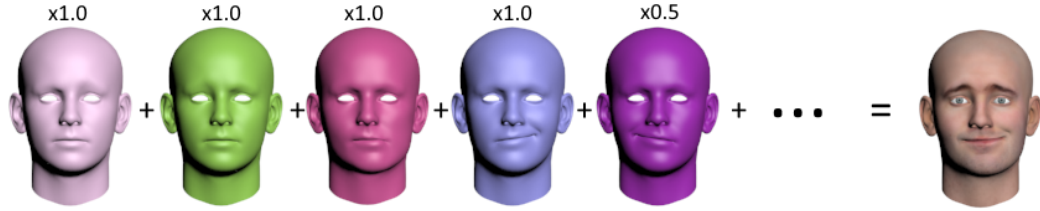


Figure 4.5: Combining expression blendshapes to make a final, composite expression.

the image-based MSE. In terms of importance values, we found the brow expressions of our model were given relatively high values compared with the other metrics although cheek and blink expressions were relatively low.

Image-Based : Perceptual Difference The perceptual difference metric, created by Yee [Yee04] is an attempt to model the human vision system to an even higher degree of accuracy than Wang et al. [WBSS04]. We found this method was sensitive to certain subtle expressions of our model such as cheek motions.

4.1.3 Test Data

To acquire our data for testing, we captured a volunteer performing 6 unique sentences from the TCD-TIMIT [HG15] dataset. We selected this because its sentences are very dense phonetically, which should help ensure that more mouth blendshapes are activated. Facial motion was captured using a tripod mounted *GoPro* camera at 120 fps with a resolution of 1920x1080. This motion was tracked and retargeted to our 3D model (Figure 4.5), which has 72 high quality blendshape expressions closely matching the FACS action units, using the professional pose-based solution by *FaceWare* [Fac16].

4.1.4 Results

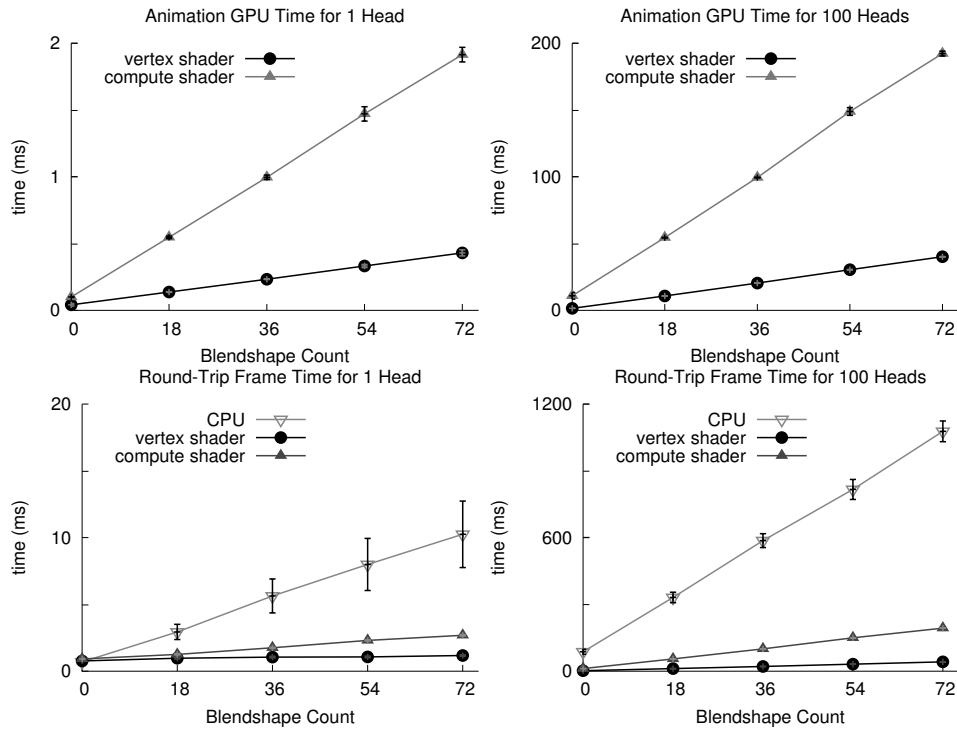


Figure 4.6: Plots compare the basic CPU method, the vertex shader method, and the compute shader method. The top row gives GPU times for rasterisation-disabled drawing for 1 and 100 heads, each head with unique blendshape weights. The bottom row gives corresponding times on the CPU.

Figure 4.6 gives our measured timing results. Our results are based on a sample size of $n = 100$ measurements, and error bars shown are plus and minus 1 standard error from the mean. All plots have error calculated, but most results have negligible variation. The columns show results from 2 scene configurations:

- 1 head rendered with unique blendshape animation frames per head.
- 100 heads rendered with unique blendshape animation frames per head.

This means that, in the case of 100 heads in Figure 4.6, each of our timings also

includes the cost of computing a set of unique blendshape animations for each head - no instancing or cloning is used. The rows show results for each scene for two timers:

- GPU timer queries to measure the time to render the scene on the graphics device (in milliseconds).
- Round-trip frame time on the CPU (in milliseconds), which includes the cost of drawing as well as CPU-side driver overheads, and associated uniform variable updates.

All results are measured on a 64-bit 6-core Intel Core i7 3.5GHz CPU with a Nvidia GeForce GTX 960 GPU, using Windows 10 64-bit driver version 362.00. We explicitly disable rasterisation before drawing, so that our measurements only consider the shader processing costs related to animation, and not any overheads created by rasterisation and fragment shading stages of the pipeline.

In all tested techniques, we observe a linear decrease in drawing time on the GPU as blendshapes are discarded. We note, in particular, that in this general purpose case the vertex shader-based animation technique is always faster than compute-shader and CPU-based animation techniques. All of the techniques have a linear cost increase as both the number of blendshapes computed, and the number of animated heads drawn increases.

We also measured for the case of cloning or reusing animation sequences within a crowd. This means that animations are computed once, and shared with subsequently drawn heads. In this special case we see compute shader animations processing many times faster than other methods on the GPU (Figure 4.7, left), and this gives an interesting case to leverage general computing shaders to performance advantage over the traditional pipeline. The CPU round-trip time (Figure 4.7, right) is negligible for

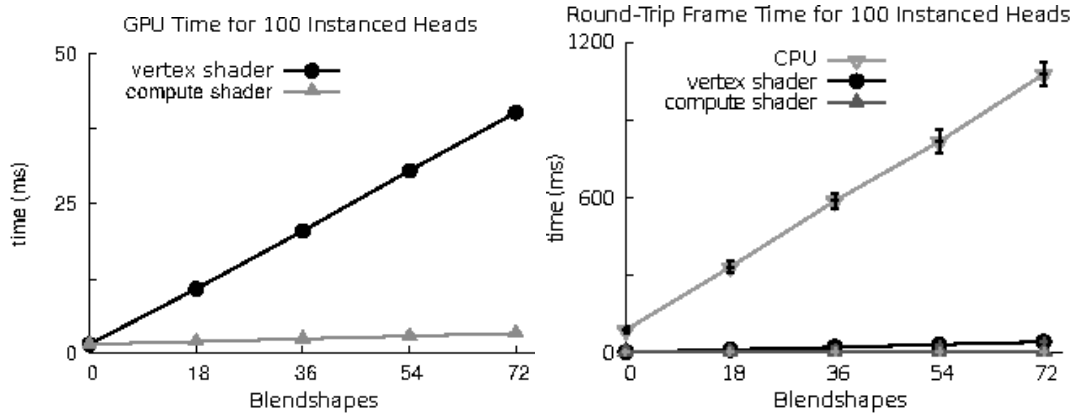


Figure 4.7: The special case of instanced rendering for crowds (sharing weights) on the GPU (left), and round-trip frame time on the CPU (right). We observe performance advantage in leveraging generic computing in shaders here.

both shader-based implementations.

4.1.5 Discussion

We have demonstrated a technique for rendering large numbers of blendshape characters in real-time using the GPU. By storing all geometry and animation information on the GPU as textures, we were able to minimise expensive graphics driver computational overheads inherent in any CPU technique. Our method when coupled with compute shaders, makes larger, fully blendshape animated crowds a possibility.

In terms of performance, we observed that the vertex shader technique was significantly faster than the CPU, or even compute shader methods in most configurations. We found that the compute shader has a much more costly overhead associated with its use, although performance was superior to the CPU technique. The compute shader method has the advantage of decoupling rendering from animation. This means in certain circumstances, such as crowd rendering where instancing is appropriate, significant speed improvements could be gained by reducing the number of compute shader

calls, as can be seen in Figure 4.6.

Performance improvements were found to increase linearly with respect to the number of blendshapes computed, with both vertex and compute shader techniques. Using a smaller subset of blendshapes does, however, reduce the expressiveness of the model as full expressions are being discarded. As a result, using a lower number of blendshapes without ordering gave inconsistent results for our model. We found though that by using sensibly ordered expressions, we were able to reduce the impact on animation quality and achieve much higher blendshape reduction levels. We expect that these reduced blendshapes would be used in less salient background characters.

We performed an ad hoc assessment of a variety of standard geometric and vision-based metrics to help establish a sensible automatic blendshape ordering for our level of detail method. We found that reductions of up to 50% can be achieved for our dataset examples with little manual intervention when using any of the 5 metrics. At a more significant reduction level of 75%, the ability of the mouth to create plausible shapes to match the audio is lost when using the image-based MSE and perceptual difference metrics. This was mostly maintained, however, with the geometric MSE, SSIM and Hausdorff distance metrics. We found that all methods gave small yet perceptually significant expressions such as blinks low importance values. We believe that this implies that using mathematical metrics alone is not a good choice for defining importance of facial blendshapes. Therefore, a perceptual metric needs to be explored that would take into account the special nature of face perception, which is known to be processed differently in the brain.

The technical contributions of this thesis so far provide a wide variety of applications for blendshape ordering methods, as well as multiple suggestions for methods to define this order. As of yet, however, we have not explored the perceptual impact of these

methods. In the following chapter, we conduct a perceptual experiment to investigate the saliency of different blendshapes at various activation levels. We then compare these to the values given to the blendshapes by the geometry-based and image-based metrics which we have used throughout this thesis, with the goal of identifying a correlation between these metrics and perception. The results of this perceptual experiment will inform the choice of metrics to be used in future blendshape importance weighting algorithms. With perceptual verification of these metrics, we can more confidently present them as good metrics for blendshape reduction. Alternatively, if they are found to not represent the perceptual impact of blendshapes well, then our perceptual results could be used either to entirely replace these metrics, or to enhance them for use in blendshape reduction with minimal perceptual impact.

Chapter 5

Perceptual Optimisation

In the previous chapters, we have shown that defining an order of blendshapes is useful for many aspects of optimizing the facial animation pipeline. In Chapter 3, blendshape importance is a vital component of the Expression Packing algorithm which defines optimal training examples for example-based blendshape transfer. In Chapter 4 we saw how blendshape importance could be used to reduce the complexity of animations by removing unused or perceptually less important blendshapes from an animation.

In this chapter, we investigate the idea of blendshape importance more closely. We explore the perceptual importance of blendshapes under different activation levels, and seek to determine if these perceptual results can be predicted by the various geometry and image error metrics we have used throughout this thesis. Of particular interest are our results from the metrics used in Chapter 4, which gave eye expressions a relatively low ranking, which is in contrast to perceptual research showing that eyes are an especially salient area of the face.

Here, we find that the eye area has a high perceptual importance which does not match its numeric difference. Similarly, we find that the eyebrow frown and mouth

frown expressions are less perceptually important than their numeric displacement values suggest. We also find that there is a large variance in the different levels of activation necessary for different expressions to become perceptible.

The work in this chapter was done in collaboration with a post-doctoral researcher from Inria and an academic collaborator from TCD (expert in statistics). I contributed to everything, while my co-authors contributed to the experiment design, analysis, and choosing and implementing the model fit.

5.1 Perceptual Importance of Blendshapes

In psychology literature, the perception of human faces is a much studied area of research. In terms of virtual characters, expressions are generally created using blendshape rigs [LAR⁺14] based on FACS action units [EF78], however these rigs are computationally expensive for real-time applications. The question of importance of blendshapes is therefore of great interest to computer games and other real-time applications, with the aim of prioritising which blendshapes to include in expressions for example-based blendshape rig creation algorithms (Chapter 3), or reducing the number of blendshapes needed for animating a rig (Section 4.1).

Due to the nature of facial perception, and how it is a special form of perception that humans are particularly attuned to [BY13, FWDT98, KMC97], we expect that differences in perception of facial action units will not align with the magnitude of displacement on the mesh caused by the action units. We hypothesise that even small displacements, such as slight eyelid movement will be considered as perceptually more salient than other larger movements, for example the puffing of cheeks, and that these will not be accurately reflected by geometric and image distance metrics.

In this section, we investigate the perceptual impact of a carefully selected range of expressive blendshapes at varying activation levels across a number of characters of different race and sex. We then compare our qualitative perceptual results to quantitative metrics in order to determine whether the perceptual effect can be predicted easily. Geometric and image-based error metrics for triangle meshes are traditionally used for predicting mesh errors such as watermarking, simplification or lossy compression. However, we aim to determine if our question of perceived blendshape importance can be predicted by simply calculating the *error* between the neutral pose and the ac-

tion unit, which is an indication of the overall vertex displacements, using common image and geometry error metrics. We investigate RMS, Hausdorff distance, and triangle distortion for our geometric errors, and MSE and SSIM for our image metrics, as described in Section 5.1.4. We then perform linear regression analysis to determine if facial action unit importance can be predicted using simple error metrics, or if a new perceptual metric specific to facial action units is required.

We run a preliminary experiment on a single character to guide parameter and stimuli choices. After a few small changes, we run the main experiment on a larger and more diverse set of characters.

Questions we address are:

- Are some facial action units more salient than others?
- Does a linear increase in activation of action units (geometry alterations) result in a linear perceptual response for different action units?
- Is the perception of facial action units consistent across faces of different race and sex?
- Can we fit a statistical model in order to predict the perceptual impact of facial action units using standard geometry and image-based error metrics?

Our results in this section could be used for optimisation of blendshape rigs through blendshape reduction, which we first discussed in Chapter 4. Blendshape rigs are memory intensive as they require hundreds of blendshapes, each with tens of thousands of vertices, to be stored in a blendshape matrix in order to display real-time animations. This also makes them computationally expensive due to the matrix-vector multiplication required [SILN11]. By identifying and removing blendshapes of lower visual

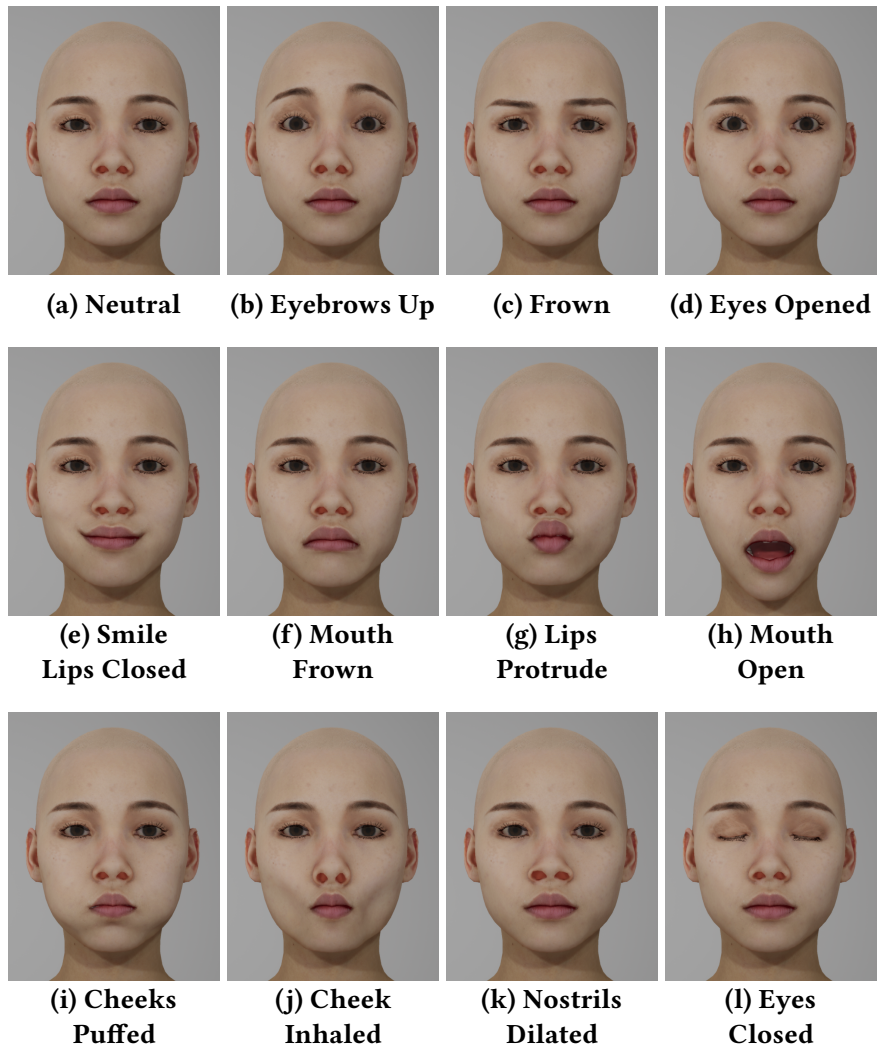


Figure 5.1: The set of blendshapes used in our main experiment, shown on the Asian Female character at full activation (1.0).

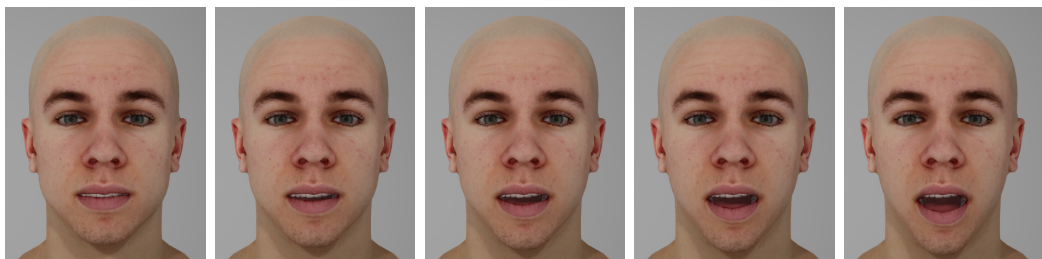


Figure 5.2: The Mouth Open blendshape shown at the 5 activation levels used in our experiments, shown on the White Male character.

saliency, which equates to simply removing rows from the blendshape matrix, we can save both memory and computation required.

5.1.1 Preliminary Experiment

We ran the preliminary experiment in Unreal Engine 4 on 8 participants in order to check the experiment procedure and stimuli and parameter selections. We did not record demographic information. We displayed two faces of the same character side-by-side in full-screen on a monitor approximately 27" in size with the participant sitting approximately one meter from the monitor. The faces took approximately 187mm x 224mm of screen space each. The face on the left showed a neutral expression, while the face on the right changed expression. We displayed the faces on a flat grey background so that there would be sufficient contrast between it and the skin tone of each character. We showed them in a front facing position in order to give the clearest view of all facial features. We asked participants to rate how similar the faces were on a scale from 1 to 9, with 1 being exactly the same and 9 being extremely different. We also recorded response time from the moment the stimulus was displayed on screen until the moment the participant submitted their response, as this could indicate how confident the participants were in their answers.

We showed a total of 12 expressions - 11 action units (AUs) and 1 neutral expression as a control. These action units, as shown in Table 5.1, were chosen to cover all areas of the face, with additional focus on the mouth due to the mouth's ability to produce a wider range of shapes. We also attempted to include opposite movements in each area, e.g. smile and frown. For each of these AUs, we showed 5 activation levels: 20%, 40%, 60%, 80%, 100%, with 100% being the maximum activation of that AU

| AU | AU Name | Area |
|-------|--------------------------------------|----------|
| 2 | Eyebrows Up | Eyebrows |
| 4 | Frown | Eyebrows |
| 5 | Eyes Opened | Eyes |
| 12 | Smile Lips Closed | Mouth |
| 15 | Mouth Frown | Mouth |
| 18 | Lips Protrude | Mouth |
| 26/27 | Mouth Open/ <i>Mouth Wide Opened</i> | Mouth |
| 34 | Cheeks Puffed | Cheek |
| 35 | Cheek Inhaled | Cheek |
| 38 | Nostrils Dilated | Nose |
| 43 | Eyes Closed | Eyes |

Table 5.1: Action Units shown in the experiment. Note: AU 27 (shown in italics alongside AU 26) was shown in the preliminary experiment, but was changed to AU 26 for the main experiment to better match the magnitude of the other AUs.

performed by the actor during the scanning process. In terms of blendshapes, this is simply a linear interpolation from the neutral face to the blendshape, with 100% being the fully activated action unit (e.g. eyes fully closed) and each intermediate step being a transition from neutral to that AU, e.g. 40% of the eyes closed AU would be eyes almost half closed (see Figure 5.2 for an example of activation levels). We showed 3 repetitions of each stimulus. For analysis, we looked at the average response across the repetitions. For each stimulus, we measured the AU’s difference from the neutral using a number of metrics: RMS Error, Hausdorff Distance, and Triangle Distortion (described in Section 5.1.4).

Preliminary Results

We ran a correlation test (Pearson’s Product-Moment) across all blendshapes and activation levels to compare perceptual responses with the mesh error calculations. We found all three methods correlated significantly with the perceived ratings of similarity;

RMS: $r = 0.71$, Hausdorff D.: $r = 0.75$, Triangle D.: $r = 0.84$ (all significant at $p < 0.05$). This result shows that these error metrics can estimate perceptual difference well, however this is something we would like to continue investigating in our main experiment.

We ran a repeated measures ANOVA on perceptual ratings with within factors Blendshape and Activation Level to estimate if there are any blendshapes which have a more prominent perceptual effect. We found that stimuli were rated differently according to Blendshape ($F(11, 77) = 48.66$, $p = 0.00$), Activation Level ($F(4, 28) = 54.32$, $p = 0.00$) and we found an interaction between both ($F(44, 308) = 5.33$, $p = 0.00$). Blendshapes which caused the largest perceptual dissimilarity were Mouth Wide Open and Eyes Closed (Tukey's HSD test: $p < 0.0002$, for all), while the Nostril Dilated, Cheeks Puffed and Frown were indiscernible from the reference neutral mesh. This pattern slightly changed with the Activation Level, where higher levels increased dissimilarity between blendshapes. Perceived difference between 80% and 100% activation was not significant, others were ($p < 0.2$, for all), implying that perceptual difference of action units changes as the activation changes until 80% activation, above which the perceptual difference is negligible. This is visualised in Figure 5.3.

We observed that the Mouth Wide Open AU was much larger in terms of displacement compared to the other blendshapes we chose. In order to have a more balanced set of blendshape stimuli, we choose to change this to a smaller Mouth Open shape in the main experiment.

These results show a promising correlation between error metrics and perceptual results, indicating that error metrics may be a good method for predicting perceptual importance. This is a result we are interested in further investigating by testing across a wide range of characters.

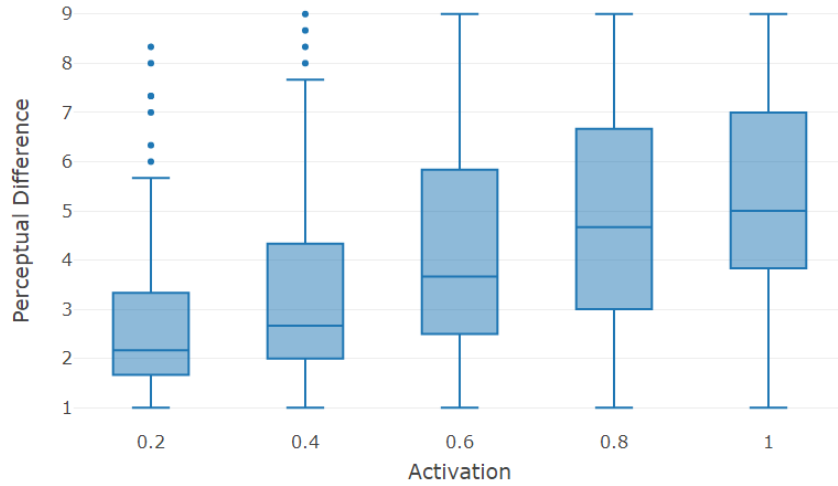


Figure 5.3: The perceptual difference per activation level for our preliminary experiment.

5.1.2 Main Experiment

We used the preliminary results to guide our main experiment, this time with more participants and character models, and some adjustments based on the results of the preliminary experiment. One of the goals of this experiment was to generalise our results beyond the single character we tested in the preliminary experiment, therefore we attempted to create a diverse set of stimuli. We included 2 characters of each Asian, Black, and White race. Within each race group, there was 1 female and 1 male character. In order to cope with this large increase in stimuli, we reduced the repetitions to one.

We changed from using the largest mouth open blendshape (Mouth Stretch AU 27) in the preliminary experiment to a smaller one (Mouth Open AU 26) in order to better match the magnitude of the other shapes. Also, this shape is more common, i.e. you are more likely to slightly open your mouth when talking/smiling than open your jaw



Figure 5.4: The experiment question screen showing the character we used for training, which is also the character used in the preliminary experiment.

as wide as possible. We altered the wording of the question based on feedback from the preliminary experiment. Instead of asking how similar the faces were, we asked how different they were, defining the low end of the slider as “No Difference” and the high end as “Extremely Different”. We also changed the UI to be more comfortable to use, shortening the slider and changing the between-stimuli screen to a 1 second focus cross, instead of a button for the participant to click. An example of the experiment as seen by participants can be seen in Figure 5.4. The rest of the experiment remained the same. We used the same experiment set-up in Unreal Engine 4, only altering the factors mentioned above, and displayed the characters at the same size on the same monitor.

Stimuli Creation

As we mentioned in Section 3.1 in this thesis, low quality stimuli can cause many issues when using blendshape transfer. For this reason, as we did in Section 3.2, we prioritise the need for high quality stimuli.

For stimuli creation, we first explored acquiring a range of high-resolution full-head meshes with semantically-matching AUs and diversity of facial features from open-source databases. However, to our knowledge, no such set exists, therefore we developed a pipeline for creating our own data-set.

We first acquired a high-end photogrammetry-scanned *template model*, created by Eisko¹, a leading Digital Double company. The character had over 200 blendshapes, inspired by the FACS system [EF78] with additional shapes for emotion and speech (Figure 5.4). Our *experiment characters* were a set of 6 neutral faces (Figure 5.5) created utilising high resolution scan data, from 3D Scan Store².

In order to obtain the 11 AUs required for the experiment for each of our experiment characters, we used the Russian 3D Scanner³ Wrap 3.4 to transfer the topology of our template model to each of the neutral characters, using some feature points as guidance so that the semantics of the topology remained the same. We then used this wrapped mesh to warp the 11 AU blendshapes of our template model to the experiment characters, thereby creating 6 new character rigs with equal topology and blendshapes. These characters can be seen in Figure 5.5. We chose not to include any hair on the characters as we are exclusively interested in facial features and wanted to avoid distracting elements.

Participants

We ran the experiment with 20 participants (3 female, 16 male, 1 prefer not to answer; 8 were in the age range 18-27, 10 in 28-37, and 2 in 38-47). All reported medium or high familiarity with computer graphics and video games. As the characters we showed

¹<https://www.eisko.com/>

²<https://www.3dscanstore.com/3d-head-models/>

³<https://www.russian3dscanner.com/>



Figure 5.5: Neutral faces of the characters used in our experiment. Left: white, middle: black, right: asian faces. Top row shows the female faces, while the bottom row shows the male faces.

in this experiment varied in race, and there is a perceptual effect of one’s own race and perception of other races [LJC91, WT03], we asked the participants to disclose their race (5 Asian, 13 White, 0 Black, 2 Other).

5.1.3 Perceptual Results

We ran a 4-way repeated measures ANOVA on the Perceptual Difference results with the within factors Sex, Ethnicity, Blendshape, and Activation Level. Due to the imbalance between participant race and sex groups, we did not include these between-groups factors in the analysis. The ANOVA results can be seen in Table 5.2. We ran post hoc analysis using Tukey’s HSD tests throughout.

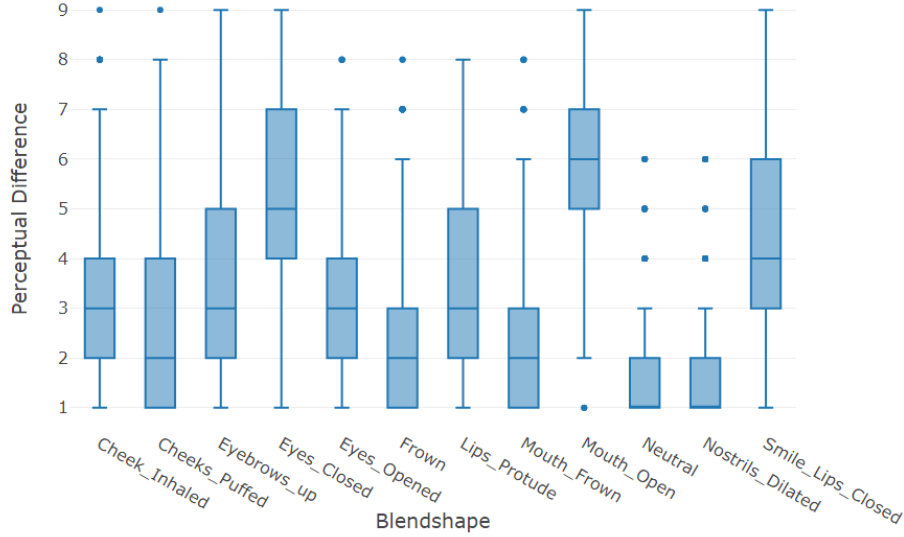


Figure 5.6: Main effect of Blendshape in the Main Experiment.

| Factor | F(DFn, DFd) = F-value | p-value | η_p^2 |
|------------------------|--------------------------|---------|------------|
| Sex | F(1, 19) = 1.727 | 0.2 | |
| Race | F(2, 38) = 4.192 | 0.02* | 0.18 |
| Blendshape | F*(2.93, 55.58) = 123.8 | 0.00* | 0.86 |
| Activation | F*(1.21, 22.90) = 158.2 | 0.00* | 0.89 |
| Sex-Race | F(2,38) = 7.826 | 0.001* | 0.29 |
| Sex-BS | F(11, 209) = 2.99 | 0.001* | 0.14 |
| Race-BS | F(22, 418) = 6.885 | 0.00* | 0.27 |
| Sex-Activation | F(4, 76) = 2.887 | 0.03* | 0.13 |
| Race-Activation | F(8, 152) = 1.581 | 0.14 | |
| BS-Activation | F(44, 836) = 19.29 | 0.00* | 0.50 |
| Sex-Race-BS | F*(6.73, 127.86) = 5.301 | 0.00* | 0.22 |
| Sex-Race-Activation | F(8, 152) = 2.031 | 0.046* | 0.10 |
| Sex-BS-Activation | F(44, 836) = 0.979 | 0.5 | |
| Race-BS-Activation | F(88, 1672) = 1.592 | 0.001* | 0.07 |
| Sex-Race-BS-Activation | F(88, 1672) = 1.68 | 0.00* | 0.08 |

Table 5.2: ANOVA interactions with dependent variable “Perceptual Difference” from the perceptual results. (BS = Blendshape, * represents significant p-values, F* stand for Greenhouse-Geisser correction for violations of sphericity). Effects sizes are reported in the last column (η_p^2).



Figure 5.7: Left: The Frown AU shown on the Asian Female and Black Male characters, Right: The Cheeks Puffed AU shown on the Asian Female and Black Male characters.

Race

We found a main effect of Race, where shape differences were less perceptible for Black characters overall than for Asian characters ($p < 0.02$). An interaction between Race and Sex gave further insight that shape differences were more perceptible for the Asian Female character than other characters except for the White Male ($p < 0.03$ for all). There was an interaction between Race and Blendshape level, which showed that these differences were mainly coming from the Frown and Cheeks Puffed AUs ($p < 0.02$). A comparison of these expressions can be seen in Figure 5.7. This implies that differences in the cheek and frown AUs were less perceptible on Black characters.

Sex

There was no main effect of the Sex of the character. We found some smaller interactions showing some individual differences in the models, but no interesting trends. This result may have been affected by our predominantly male participant pool.

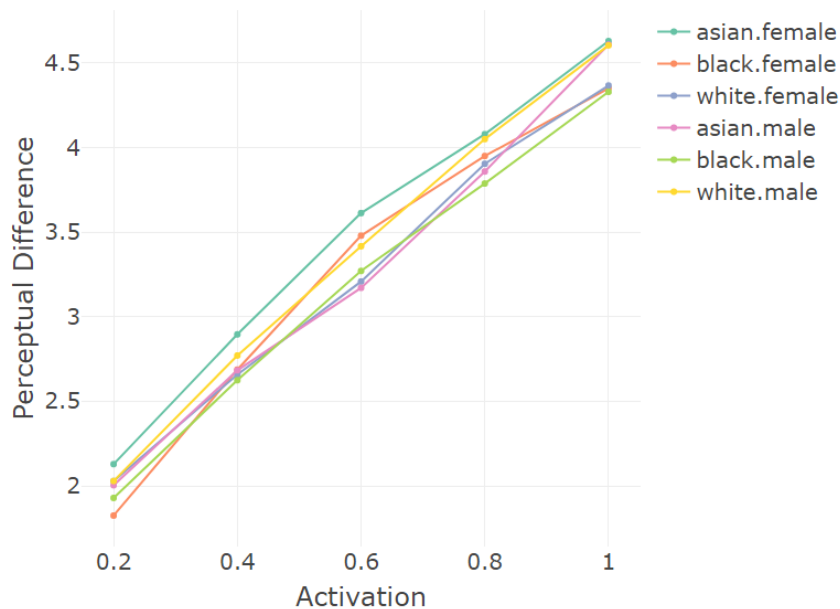


Figure 5.8: Perceived differences of activations between characters.

Activation

A main effect of Activation showed a significant difference between each activation level, with a linear increase in perceived differences as the activation increased.

There was no difference across all characters and blendshapes at the lowest activation level of 20%, however some characters were rated as relatively more different at higher activation levels. There was a significant difference between Asian Female's 60% and that of the White Female, Black Female and Black Male characters, but no significant difference between Asian Female's 60% and their 80% ($p < 0.02$). This is visualised in Figure 5.8. You can see that the perceived difference of the Asian Female AUs at 60% activation (0.6 on the graph) is similar to other character's AUs at 80%.

The only blendshapes which are significantly different from the Neutral at 20% activation are Eyes Closed and Mouth Open ($p < 0.00005$). Cheeks Puffed and Mouth

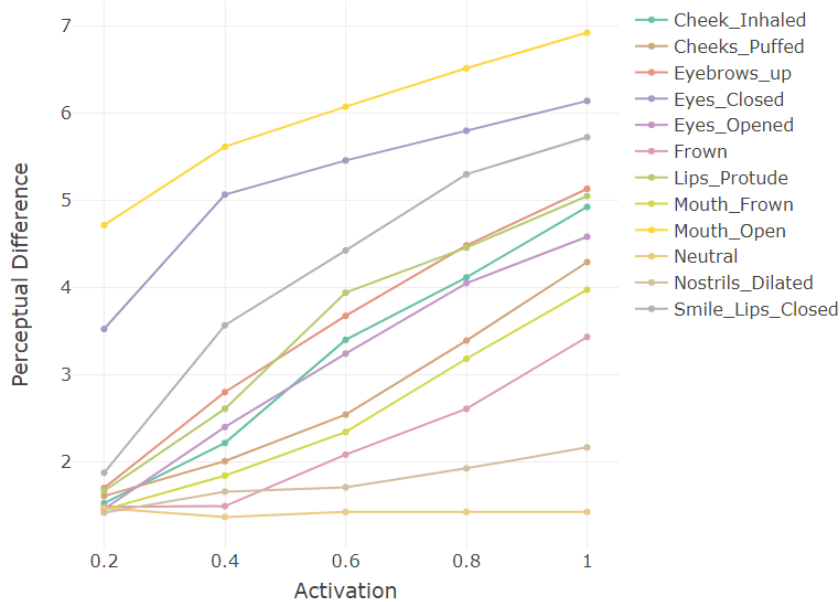


Figure 5.9: Perceived differences of activations between blendshapes.

Frown are only significantly different from the Neutral at 60% activation or higher ($p < 0.0002$), Frown is only significantly different at 80% activation or higher ($p < 0.00005$), and Nostrils Dilated is significantly different at 100% activation ($p < 0.02$). All other shapes were perceptually different from the neutral at 40% activation ($p < 0.005$), indicating that 40% activation seems to be the point at which people can visualise displacements well for most action units at this size on-screen. This is visualised in Figure 5.9.

Blendshapes

We found a main effect of blendshape where Mouth Open and Eyes Closed were significantly more different than every other blendshape for all Characters ($p < 0.005$). Mouth Open was significantly more different from the neutral than every shape ($p < 0.005$) except Eyes Closed for Asian Female and Black Male, where the difference was

insignificant. Nostrils Dilated was not significantly different from Neutral for all characters. Frown was not significantly different from Neutral for White Female, Black Female, and Asian Male.

Mouth Frown was the only blendshape to be rated significantly differently between the sexes ($p < 0.05$), with the female characters being rated as more different. This could be due to the gender stereotype that women are more likely to show happiness and men are more likely to show anger [HAJK04], therefore a frown is more unexpected on a female face.

We found an interaction with Race. Eyes Opened and Eyebrows Up were perceived as significantly more different from the Neutral, and Nostrils Dilated, Mouth Frown and Lips Protrude were perceived as significantly less different from the Neutral for the Black characters than the Asian characters ($p < 0.05$). This could be caused by the higher contrast between the eyes and skin, and lower contrast around the lips, for the Black characters. Eyes Opened was rated as having a significantly lower perceptual difference, and Cheeks Puffed was rated as having a significantly higher difference for the White characters than the Black characters ($p < 0.00005$). There were no significant per-blendshape differences between the White and Asian characters.

We also found an interaction with Race and Sex, showing some per-character differences. Specifically, for both the Asian Female and White Male characters, the Frown and Cheeks Puffed AUs of the Black Female and Asian Male characters, and the Cheeks Puffed of the Black Male character were all rated significantly lower. The Asian Female was also rated as having a higher perceptual difference than the Black Male for the Cheeks Puffed AU ($p < 0.02$ for all). There was also a difference in the Eyebrows Up AU, with the Black characters being rated more different than the Asian Male character ($p < 0.05$). The per-character perceptual differences for these blendshapes can be

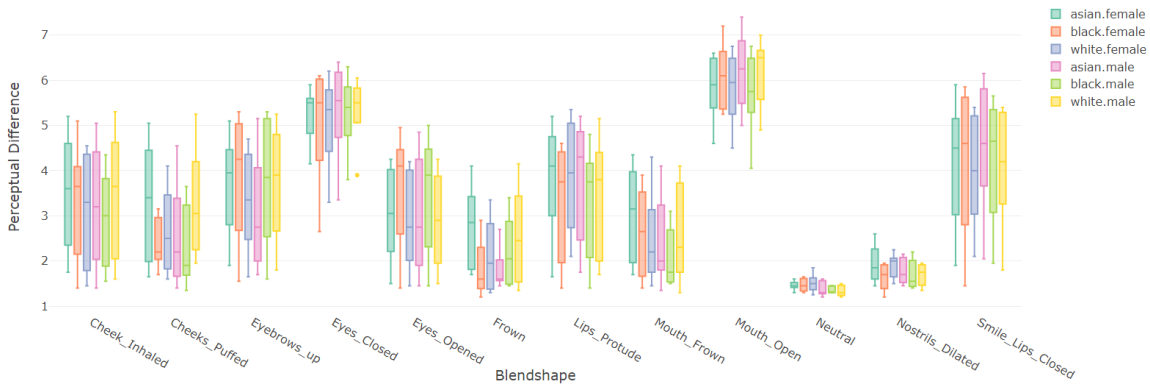


Figure 5.10: Perceived differences of blendshapes between characters.

seen in Figure 5.10.

5.1.4 Error Results

As we discussed at the beginning of this section, we do not only wish to examine the perceptual importance of different action units, we also want to investigate the relationship between numerical metrics and perception. Here, we introduce the error metrics we investigate and begin to discuss their relationship with perception. While we define these as *error metrics* as they are more commonly used to measure error in the fields of geometry and image processing, we use them more as a measure of *displacement*, and refer to them as displacement metrics after this section.

Error Metrics

We calculate each metric for each activation level of each blendshape for each character. Each metric is calculated between an AU and the neutral mesh. Our geometric metrics are calculated using the vertex positions of each stimulus. To calculate our image metric results, we took screenshots of each stimulus during the experiment and cropped out a

large amount of the empty space surrounding each head. An example of the crop can be seen in Figure 5.5. We compare each activation level of each blendshape of each character to the neutral of the same character. MSE and SSIM were calculated using scikit-image [vdWSN⁺14].

Geometric: Root-Mean-Square We calculate the RMS error between two meshes by getting the sum across all vertices $n \in N$ of the square root of the average of the square of each (x, y, z) component of each vertex, as described in Equation 5.1.

$$\delta_{RMS} = \sum_{n=1}^N \sqrt{\frac{1}{3} \mathbf{v}_n \cdot \mathbf{v}_n} \quad (5.1)$$

Geometric: Hausdorff We calculate the bi-directional Hausdorff distance between two meshes of vertex sets N and M using the equation described in Equation 5.2 where $d(n, m)$ is the distance between two vertices.

$$d_H = \max \left\{ \sup_{n \in N} \inf_{m \in M} d(n, m), \sup_{m \in M} \inf_{n \in N} d(n, m) \right\} \quad (5.2)$$

Geometric: Triangle Distortion We define the triangle distortion δ_t of triangle \mathbf{uvw} between two meshes A and B of equal connectivity in Equation 5.3.

$$\delta_t = \max \left\{ \frac{\mathbf{vu}_A}{\mathbf{vu}_B}, \frac{\mathbf{vu}_B}{\mathbf{vu}_A} \right\} + \max \left\{ \frac{\mathbf{vw}_A}{\mathbf{vw}_B}, \frac{\mathbf{vw}_B}{\mathbf{vw}_A} \right\} - 2 \quad (5.3)$$

Image: Mean-Square-Error We calculate MSE by getting the per-pixel average error between images A and B , where N is the total number of pixels in the image, and \mathbf{x}_n^A is the n^{th} pixel of image A .

$$\delta_{MSE} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n^A - \mathbf{x}_n^B \quad (5.4)$$

Image: Structural Similarity Index Metric SSIM is calculated as defined by Wang et al. [WBSS04] and using the default suggested parameters. It is designed to model the response of the human vision system and therefore should correlate better to our perceptual results than MSE. As this metric measures similarity rather than difference, we sometimes invert this metric for the purpose of visualisation for better comparison with our other metrics. In these cases, we refer to it as 1-SSIM, as SSIM returns a value between 0 and 1 so to invert we simply subtract from 1.

Results

We found a correlation between all of our error metrics and the perceptual experiment results using Pearson’s Product-Moment correlation test, shown in Table 5.4. Overall, the geometry metrics correlated better with the perceptual results. This is interesting as image metrics are based upon the exact image that the participants saw, as opposed to the geometry metrics which do not directly translate to a change in what the participant sees. Our results in this section are visualised in Figures 5.11 and 5.12, and for ease of comparison we provide a table of the various orderings of action units in Table 5.3. We also provide a graph of Perceptual Difference against each of our metrics at the end of this section, in Figure 5.16.

Mouth Open had the highest error value when measured using RMS, Hausdorff and MSE, which matches with our perceptual results, however none of these metrics gave particularly high values to Eyes Closed, which was the second highest rated perceptually different AU. The Triangle Difference metric gave Lips Protrude the highest

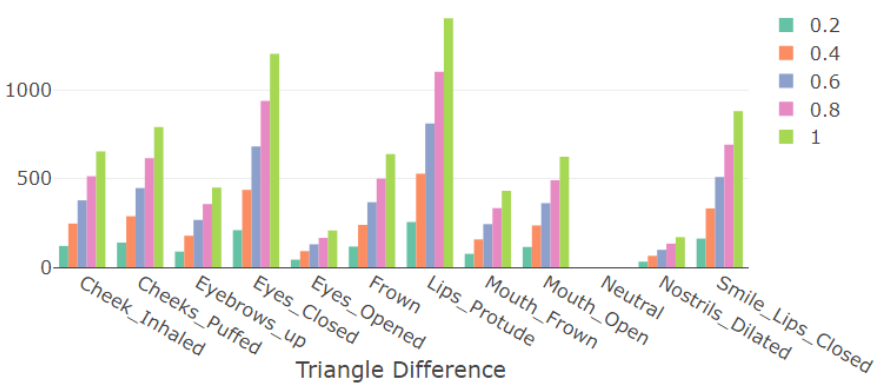
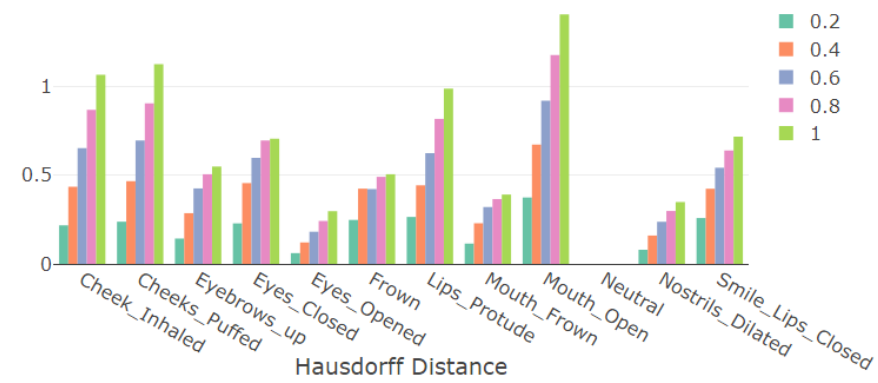
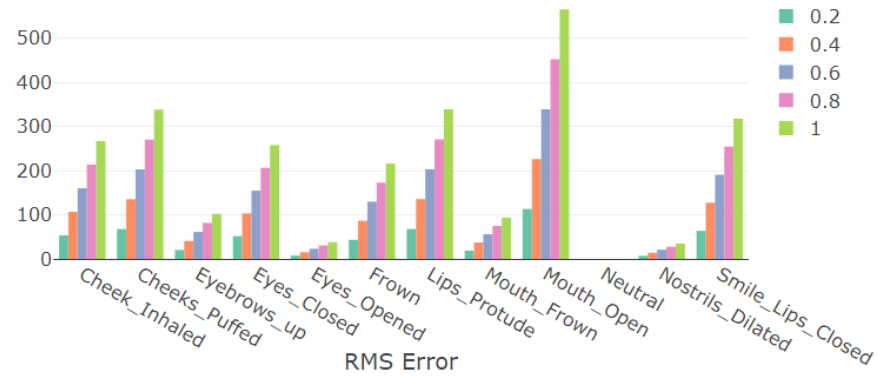


Figure 5.11: Geometry errors averaged across characters.

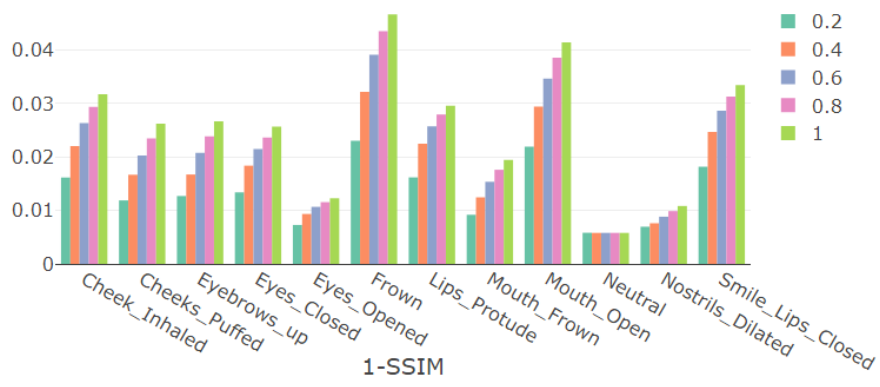
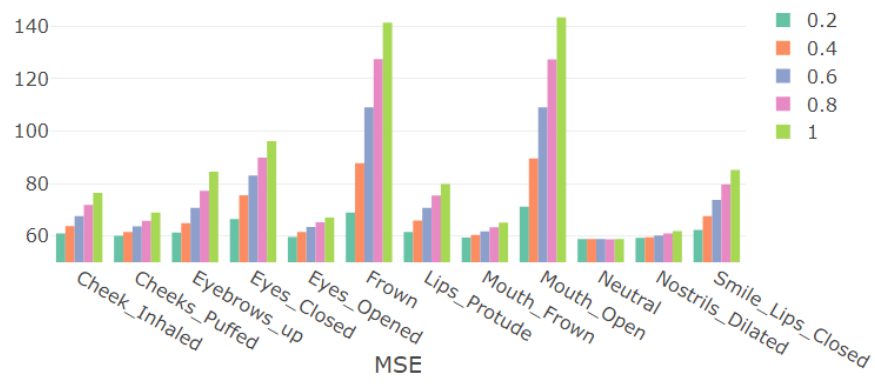


Figure 5.12: Image errors averaged across characters. Note: we use 1-SSIM for our SSIM graphs, as SSIM is a measure of Similarity between 0 and 1, and we are interested in the perceived difference.

error rating, which was rated about average perceptually, and SSIM rated Frown the highest, which is one of the least perceptually different AUs. All metrics rated Eyes Opened low, giving similar importance to Nostrils Dilated which was imperceptible from Neutral, even though Eyes Opened is significantly different from Neutral perceptually. Eyebrows Up was rated relatively low by geometric error metrics although it was considered perceptually as important or more important than all AUs except for Mouth Open and Eyes Closed. The image metrics gave a higher importance to Eyebrows Up. While all of our metrics correlate well with our perceptual results, we see that the action units which are worst predicted by our metrics are mainly eye and eyebrow action units.

We also provide a heatmap visualisation of RMS error, created by assigning a colour gradient to the range of displacements on the meshes, from zero displacement to the maximum displacement across all action units, on all of the action units we tested and ordered them based on their average perceptual difference as rated by our participants (see Figure 5.13). This shows more intuitively how small the RMS error is for certain highly salient action units such as Eyes Closed and Eyes Opened.

5.1.5 Model Fit

Our initial ANOVA analysis focussed purely on the perceptual results. This allowed us to draw conclusions based on the factors of Race, Sex, Blendshape, and Activation. While our conclusions on Activation were interesting, the definition of Activation varies a lot between action units, as 100% activation is simply the maximum that the scanned actor could pose. This does not tell us much about the actual shape or displacement of the action unit. In order to further investigate this data, we need to look at these results

| Order | RMS | Hausdorff | Tri. Diff |
|-------|-------------------|-------------------|-------------------|
| 1 | Mouth Open | Mouth Open | Lips Protrude |
| 2 | Lips Protrude | Cheeks Puffed | Eyes Closed |
| 3 | Cheeks Puffed | Cheek Inhaled | Smile Lips Closed |
| 4 | Smile Lips Closed | Lips Protrude | Cheek Inhaled |
| 5 | Cheek Inhaled | Eyes Closed | Cheeks Puffed |
| 6 | Eyes Closed | Smile Lips Closed | Frown |
| 7 | Frown | Frown | Mouth Open |
| 8 | Eyebrows Up | Eyebrows Up | Eyebrows Up |
| 9 | Mouth Frown | Mouth Frown | Mouth Frown |
| 10 | Eyes Opened | Nostrils Dilated | Eyes Opened |
| 11 | Nostrils Dilated | Eyes Opened | Nostrils Dilated |
| Order | MSE | 1-SSIM | Perceptual |
| 1 | Mouth Open | Frown | Mouth Open |
| 2 | Frown | Mouth Open | Eyes Closed |
| 3 | Eyes Closed | Smile Lips Closed | Smile Lips Closed |
| 4 | Smile Lips Closed | Cheek Inhaled | Eyebrows Up |
| 5 | Eyebrows Up | Lips Protrude | Lips Protrude |
| 6 | Lips Protrude | Eyes Closed | Cheek Inhaled |
| 7 | Cheeks Puffed | Eyebrows Up | Eyes Opened |
| 8 | Cheek Inhaled | Cheeks Puffed | Cheeks Puffed |
| 9 | Eyes Opened | Mouth Frown | Mouth Frown |
| 10 | Mouth Frown | Eyes Opened | Frown |
| 11 | Nostrils Dilated | Nostrils Dilated | Nostrils Dilated |

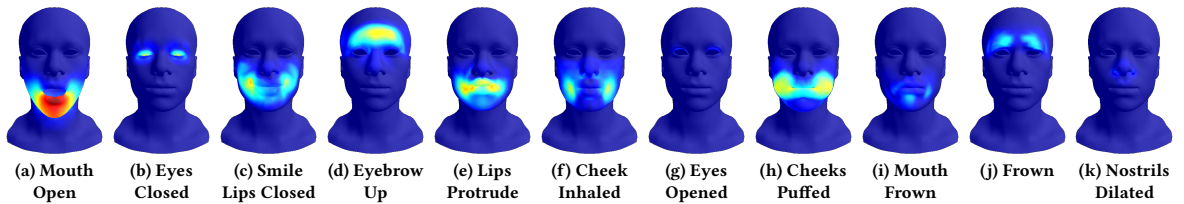
Table 5.3: Our blendshapes ordered from most to least important by each of the different error methods and the perceptual results, averaged across all characters and activation levels.

| Metric | Type | Correlation | p-value |
|---------------------|----------|-------------|---------|
| RMS | geometry | 0.76 | 0.00* |
| Hausdorff | geometry | 0.76 | 0.00* |
| Triangle Distortion | geometry | 0.68 | 0.00* |
| MSE | image | 0.5 | 0.00* |
| SSIM | image | -0.56 | 0.00* |

Table 5.4: Pearson correlations between the error metrics we calculated and the perceptual results. Includes Neutrals, average Similarity result across participants.

| | RMS | Hausdorff | Tri. Diff. | MSE | SSIM |
|-------------------|------|-----------|------------|-------|-------|
| Cheek Inhaled | 0.97 | 0.97 | 0.97 | 0.55 | -0.59 |
| Cheeks Puffed | 0.88 | 0.87 | 0.91 | 0.61 | -0.64 |
| Eyebrows up | 0.96 | 0.94 | 0.96 | 0.45 | -0.75 |
| Eyes Closed | 0.89 | 0.93 | 0.87 | 0.48 | -0.8 |
| Frown | 0.84 | 0.62 | 0.85 | 0.77 | -0.65 |
| Mouth Open | 0.93 | 0.91 | 0.92 | 0.77 | -0.81 |
| Lips Protrude | 0.95 | 0.95 | 0.95 | 0.63 | -0.65 |
| Mouth Frown | 0.92 | 0.81 | 0.93 | 0.38 | -0.63 |
| Nostrils Dilated | 0.79 | 0.75 | 0.82 | 0.32* | -0.63 |
| Eyes Opened | 0.95 | 0.96 | 0.96 | 0.13* | -0.94 |
| Smile Lips Closed | 0.96 | 0.95 | 0.95 | 0.49 | -0.61 |
| Female | 0.76 | 0.77 | 0.69 | 0.5 | -0.55 |
| Male | 0.76 | 0.76 | 0.67 | 0.5 | -0.59 |
| Asian | 0.78 | 0.79 | 0.71 | 0.63 | -0.58 |
| Black | 0.7 | 0.7 | 0.64 | 0.56 | -0.55 |
| White | 0.79 | 0.8 | 0.69 | 0.5 | -0.62 |

Table 5.5: Pearson correlations between our error metrics and the perceptual results broken down by Blendshape, Sex, and Race; $p = 0.00$ for all RMS and Triangle Difference (Tri. Diff.), $p < 0.0003$ for Hausdorff, $p < 0.02$ for MSE, $p < 0.001$ for SSIM. * denotes insignificant results. Similarity results averaged across participants.



Highest Perceptual Difference ————— Lowest Perceptual Difference

Figure 5.13: A heatmap visualisation of RMS error on the set of blendshapes used in our main experiment, shown on the template character at full activation (1.0). The blendshapes are shown in order of Perceptual Difference, from highest to lowest. No displacement is dark blue, while the largest displacement is shown in dark red in (as seen in (a) Mouth Open). The colours in order of displacement are: dark blue, blue, cyan, yellow, red, dark red.

with reference to the various error metrics we calculated. By fitting a Generalised Linear Model to our data using the perceptual difference results and error metrics, we can infer information about the perceptual importance of blendshapes relative to their errors. This model will also be useful in predicting potential visual saliency of these blendshapes on other characters, or potentially new blendshapes that we did not test in this study.

Response time and perceived difference are both collected from participants. We show first that these two quantities are negatively correlated indicating that the stronger the perceived difference the quicker the response time. Then we propose to fit a model to predict the perceived difference from metrics with interactions from Blendshape types and the virtual character Race and Sex.

Response Time and Perceptual Difference

We first analyse response times with relation to perceived difference using ANOVA. We observed a significant negative correlation between the Response Time and the perceived difference recorded from participants (Table 5.7). Neither the character Race and Sex nor the Blendshapes had any impact on these results (Table 5.6). The stronger the perceived difference of the presented blendshape to the neutral blendshape, the quicker the participant is to provide their response. This indicates that it takes more time for participant to notice and score their perceived difference for subtle deformations of the virtual character. This was as expected, and provides a sanity-check on the fact that participants were in-fact rating as quickly as they could.

| | Df | Sum Sq | Mean Sq | F value | Pr(> F) |
|------------------------|------|--------------|--------------|-------------|-----------|
| Difference | 1 | 2.792970e+03 | 2792.9703977 | 102.4093736 | 0.0000000 |
| BS | 11 | 4.079669e+02 | 37.0879041 | 1.3598959 | 0.1846796 |
| Sex | 1 | 2.475704e-01 | 0.2475704 | 0.0090776 | 0.9240979 |
| Race | 2 | 5.707053e+00 | 2.8535267 | 0.1046298 | 0.9006593 |
| Difference:BS | 11 | 3.921903e+02 | 35.6536644 | 1.3073069 | 0.2129383 |
| Difference:Sex | 1 | 3.017544e+01 | 30.1754445 | 1.1064379 | 0.2928936 |
| BS:Sex | 11 | 3.801665e+02 | 34.5605904 | 1.2672273 | 0.2366319 |
| Difference:Race | 2 | 1.054015e+02 | 52.7007635 | 1.9323700 | 0.1448812 |
| BS:Race | 22 | 4.999137e+02 | 22.7233487 | 0.8331932 | 0.6861076 |
| Sex:Race | 2 | 1.007804e+02 | 50.3902075 | 1.8476492 | 0.1576835 |
| Difference:BS:Sex | 11 | 2.091324e+02 | 19.0120384 | 0.6971112 | 0.7426163 |
| Difference:BS:Race | 22 | 5.742463e+02 | 26.1021065 | 0.9570815 | 0.5173894 |
| Difference:Sex:Race | 2 | 5.773405e+01 | 28.8670232 | 1.0584623 | 0.3470441 |
| BS:Sex:Race | 22 | 4.022006e+02 | 18.2818441 | 0.6703373 | 0.8726223 |
| Difference:BS:Sex:Race | 22 | 4.154845e+02 | 18.8856577 | 0.6924772 | 0.8517504 |
| Residuals | 7056 | 1.924355e+05 | 27.2726050 | NA | NA |

Table 5.6: ANOVA test Response Time of participants with relation to Perceptual Difference and blendshape. Only Perceived Difference is a significant regressor for Response Time. (Difference = PerceptualDifference, BS = Blendshapes)

| | Estimate | Std. Error | t value | Pr(> t) |
|-------------|------------|------------|-----------|-----------|
| (Intercept) | 6.2115387 | 0.1146936 | 54.15766 | 0 |
| Difference | -0.2972758 | 0.0293540 | -10.12727 | 0 |

Table 5.7: A linear model is fitted to predict Response Time with relation to Perceptual Difference and it shows a negative correlation between Perceptual Difference and Response Time of participants. (Difference = PerceptualDifference, BS = Blendshapes)

| Model | Gaussian | Poisson |
|---|----------|--------------------|
| RMS*Blendshapes | 24878 | 24688 (Id) |
| RMS*Blendshapes+Race:Sex | 24853 | 24673 (Id) |
| RMS*Blendshapes*Sex*Race | 24810 | 24749 (Id) |
| Hausdorff*Blendshapes*Sex*Race | 24898 | 24814 (Id) |
| TriangleDifference*Blendshapes*Sex*Race | 24819 | 24751 (Id) |
| SSIM*Blendshapes | 26287 | 25591 (Id) |
| SSIM*Blendshapes*Sex*Race | 24841 | 24758 (log) |
| MSE*Blendshapes*Sex*Race | 24839 | 24776 (Id) |
| Activation*Blendshapes*Sex*Race | 24820 | 24759 (Id) |

Table 5.8: Model comparison with AIC↓ to explain the perceived difference. Best link function reported for Poisson between Identity(Id), log and sqrt. Only Id link function has been tested with Gaussian distribution.

Perceptual Difference and Displacement Metrics

Several generalised models were tested and compared using Aikake Information Criterion (AIC) [DB08]. Poisson distribution is compared to Gaussian distribution. Poisson distribution better captures the discrete nature of the perceived difference: all AICs with Poisson distribution are lower than their corresponding model with Gaussian distribution (Table 5.8).

In the geometry domain, we selected RMS as it is the most popular metric to measure mesh deformation. We note that RMS also achieves the lowest AIC in comparison to Hausdorff and Triangle Difference (Table 5.8) indicating that it is the best predictor among the geometry metrics. Similarly in the image domain, SSIM, defined as a perceptual metric [WBSS04], is chosen to create a model for predicting the perceived difference. All models tested in Table 5.8 are acceptable ones for our dataset as validated using their deviance (Poisson distribution) with χ^2 test [DB08].

| | Df | Sum Sq | Mean Sq | F value | Pr(> F) |
|--------------------------|------|--------------|--------------|-------------|-----------|
| RMS | 1 | 11125.300638 | 11125.300638 | 6174.531821 | 0.0000000 |
| Blendshapes | 11 | 5114.028326 | 464.911666 | 258.025555 | 0.0000000 |
| Sex | 1 | 3.555549 | 3.555549 | 1.973326 | 0.1601392 |
| Race | 2 | 24.913623 | 12.456811 | 6.913519 | 0.0010010 |
| RMS:Blendshapes | 10 | 2103.488064 | 210.348806 | 116.743398 | 0.0000000 |
| RMS:Sex | 1 | 7.984011 | 7.984011 | 4.431119 | 0.0353246 |
| Blendshapes:Sex | 11 | 30.062896 | 2.732991 | 1.516807 | 0.1178373 |
| RMS:Race | 2 | 21.928141 | 10.964071 | 6.085049 | 0.0022886 |
| Blendshapes:Race | 22 | 132.014531 | 6.000661 | 3.330361 | 0.0000002 |
| Sex:Race | 2 | 36.295358 | 18.147679 | 10.071946 | 0.0000429 |
| RMS:Blendshapes:Sex | 10 | 10.545043 | 1.054504 | 0.585249 | 0.8274276 |
| RMS:Blendshapes:Race | 20 | 63.216912 | 3.160846 | 1.754266 | 0.0198763 |
| RMS:Sex:Race | 2 | 6.866608 | 3.433304 | 1.905481 | 0.1488276 |
| Blendshapes:Sex:Race | 22 | 140.943779 | 6.406535 | 3.555621 | 0.0000000 |
| RMS:Blendshapes:Sex:Race | 20 | 58.884299 | 2.944215 | 1.634037 | 0.0368748 |
| Residuals | 7062 | 12724.345000 | 1.801805 | NA | NA |

Table 5.9: Perceptual difference with relation to RMS and interaction with character Sex, character Race and Blendshapes. The red rows mark insignificant results.

Perceptual Difference and RMS Table 5.9 shows the ANOVA test when fitting a linear regression to explain the Perceived Difference (response variable) with explanatory variables RMS, Blendshapes, Sex, and Race. As can be seen, most of the perceived difference is explained using RMS and Blendshapes with their interactions (variable highlighted in green, Table 5.9), indicating that Sex and Race have little impact on the perceptual results.

Perceptual Difference and SSIM Table 5.10 shows the ANOVA analysis when using the image metric SSIM, Blendshapes, Sex, and Race. The SSIM is systematically better than the other image metric MSE in explaining the perceived difference hence SSIM is the chosen image metric as a predictor to the perceived difference. SSIM as an image metric (captured in a 2D projective space), is not as powerful as the geometry metric RMS (measuring the deformation in 3D) for explaining the perceived

| | Df | Sum Sq | Mean Sq | F value | Pr(> F) |
|---------------------------|------|-------------|-------------|-------------|-----------|
| SSIM | 1 | 6135.88301 | 6135.883010 | 3393.489386 | 0.0000000 |
| Blendshapes | 11 | 8500.78006 | 772.798188 | 427.400986 | 0.0000000 |
| Sex | 1 | 204.14274 | 204.142736 | 112.902447 | 0.0000000 |
| Race | 2 | 615.24550 | 307.622752 | 170.132733 | 0.0000000 |
| SSIM:Blendshapes | 11 | 924.91321 | 84.083019 | 46.502652 | 0.0000000 |
| SSIM:Sex | 1 | 23.56551 | 23.565513 | 13.033058 | 0.0003082 |
| Blendshapes:Sex | 11 | 405.86809 | 36.897099 | 20.406177 | 0.0000000 |
| SSIM:Race | 2 | 113.56291 | 56.781454 | 31.403347 | 0.0000000 |
| Blendshapes:Race | 22 | 484.94349 | 22.042886 | 12.190959 | 0.0000000 |
| Sex:Race | 2 | 645.40840 | 322.704198 | 178.473623 | 0.0000000 |
| SSIM:Blendshapes:Sex | 11 | 97.79763 | 8.890693 | 4.917055 | 0.0000001 |
| SSIM:Blendshapes:Race | 22 | 148.00950 | 6.727704 | 3.720800 | 0.0000000 |
| SSIM:Sex:Race | 2 | 148.00017 | 74.000085 | 40.926221 | 0.0000000 |
| Blendshapes:Sex:Race | 22 | 297.96831 | 13.544014 | 7.490604 | 0.0000000 |
| SSIM:Blendshapes:Sex:Race | 22 | 100.09184 | 4.549629 | 2.516202 | 0.0001106 |
| Residuals | 7056 | 12758.19241 | 1.808134 | NA | NA |

Table 5.10: Perceptual difference with relation to SSIM and interaction with character Sex, character Race and Blendshapes.

difference (as shown by the Sum of Squares (Sum Sq.) column in first rows in Tables 5.9 and 5.10). While our participants only visualised a projected image directly associated with image metrics, their recorded perceived difference is better explained by geometric metrics computed from 3D meshes. A potential explanation may be that because faces are very familiar objects, a 3D representation is automatically imagined or inferred by participants when viewing 2D facial images. Having a model fitted using image metrics can be useful however a prediction of perceived difference has to be computed from only image information (i.e. when geometry metrics are not available).

Models for prediction Mathematically these models can be written as :

$$\hat{y}_{j,k} = g^{-1}(\alpha_{j,k} + \beta_{j,k} x) \quad (5.5)$$

for $x * Blendshapes * Sex * Race$ or $x * Blendshapes + Sex : Race$ listed in Table 5.8. The simpler models using only metric x and blendshapes (for $x * Blendshapes$ listed in Table 5.8) can be written as:

$$\hat{y}_j = g^{-1}(\alpha_j + \beta_j x) \quad (5.6)$$

where $j = 1, \dots, 12$ indicates the Blendshape, $k = 1, \dots, 6$ indicates the virtual character used (Sex,Race) and g is the link function used. \hat{y} is the predicted perceived difference and x is the chosen metric. These models are shown in Figure 5.14 where graphs (a) and (c) fits a model of the form Equation 5.5 while graphs (b) and (d) use the simpler model Equation 5.6. We note that when using RMS all characters are scored in a similar fashion for each blendshapes (a) and are well summarised by the average fit computed across the 6 virtual characters. In practice this means that the simpler model shown in (b) can be used to predict perceived difference for other virtual characters. On the other hand, SSIM is not as effective for capturing a usable model that generalises for other virtual characters.

Table 5.11 reports the fitted parameters for RMS shown in Figure 5.14 (a). For the Neutral Blendshape, the equation for computing a prediction with relation to RMS x is $\hat{y} = 1.4216667 + 0.0159118 x$ (note that $x = 0$ for RMS for all Neutral shapes therefore the predicted values is $\hat{y} = 1.4216667$ in practice); for the Cheek Inhaled Blendshape, the equation for computing a prediction with relation to RMS x is $\hat{y} = 1.4216667 - 0.8001795 + (0.0159118 + 0.0003925) x \simeq 1.4216667 - 0.8001795 + (0.0159118) x$ considering the uncertainty associated with the parameters; The equation for Cheeks Puffed is $\hat{y} = (1.4216667 - 0.5800126) + (0.0159118 - 0.0064210) x$; etc.

Using the information from Table 5.11, it is possible to use our models to predict

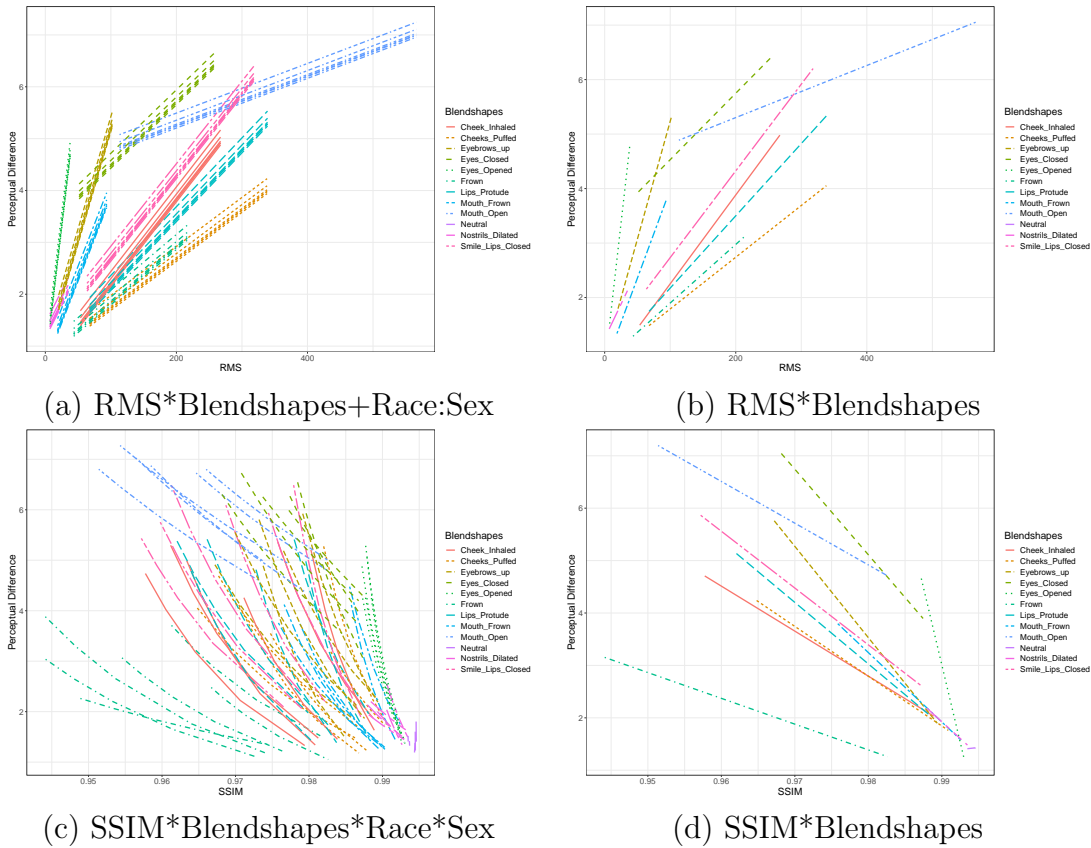


Figure 5.14: Model fits for Perceived difference using geometry metric RMS ((a) and (b)) and image metric SSIM ((c) and (d)) as per models listed Table 5.8. The 6 virtual characters behave in a similar fashion when using RMS (a) and are well captured with the simpler model (b) corresponding to the average model fit across the 6 virtual characters for each blendshape.

| | Estimate | Std. Error | z value | Pr(> z) |
|----------------------|------------|------------|-------------|-----------|
| (Intercept) | 1.4216667 | 0.0486769 | 29.2061637 | 0.0000000 |
| RMS | 0.0159118 | 0.0008862 | 17.9554892 | 0.0000000 |
| Cheek_Inhaled | -0.8001795 | 0.1436953 | -5.5685842 | 0.0000000 |
| Cheeks_Puffed | -0.5800126 | 0.1402300 | -4.1361520 | 0.0000353 |
| Eyebrows_up | -0.5072498 | 0.1540199 | -3.2934038 | 0.0009898 |
| Eyes_Closed | 1.8899715 | 0.2071724 | 9.1226971 | 0.0000000 |
| Eyes_Opened | -0.7117859 | 0.1439472 | -4.9447705 | 0.0000008 |
| Frown | -0.5975084 | 0.1303547 | -4.5837118 | 0.0000046 |
| Lips_Protude | -0.5734539 | 0.1526630 | -3.7563376 | 0.0001724 |
| Mouth_Frown | -0.6914172 | 0.1345563 | -5.1384967 | 0.0000003 |
| Mouth_Open | 2.9246443 | 0.2261913 | 12.9299567 | 0.0000000 |
| Nostrils_Dilated | -0.1725020 | 0.1292436 | -1.3347043 | 0.1819731 |
| Smile_Lips_Closed | -0.2794534 | 0.1667462 | -1.6759209 | 0.0937537 |
| RMS:Cheek_Inhaled | 0.0003925 | 0.0012747 | 0.3078796 | 0.7581739 |
| RMS:Cheeks_Puffed | -0.0064210 | 0.0011175 | -5.7456773 | 0.0000000 |
| RMS:Eyebrows_up | 0.0272924 | 0.0026938 | 10.1313825 | 0.0000000 |
| RMS:Eyes_Closed | -0.0037234 | 0.0015423 | -2.4142767 | 0.0157665 |
| RMS:Eyes_Opened | 0.0904807 | 0.0064230 | 14.0870002 | 0.0000000 |
| RMS:Frown | -0.0051578 | 0.0013076 | -3.9444512 | 0.0000800 |
| RMS:Lips_Protude | -0.0026542 | 0.0011687 | -2.2711484 | 0.0231380 |
| RMS:Mouth_Frown | 0.0166446 | 0.0025212 | 6.6018302 | 0.0000000 |
| RMS:Mouth_Open | -0.0111309 | 0.0010812 | -10.2948576 | 0.0000000 |
| RMS:Nostrils_Dilated | 0.0093119 | 0.0055673 | 1.6725981 | 0.0944064 |

Table 5.11: Coefficients for model in Figure 5.14 (b) using the Neutral expression as reference. Red values indicate models with a bad fit. $\text{Pr}(> |z|)$ indicates the p-value.

the perceptual difference of an action unit as displayed on a virtual character, given the RMS error from the Neutral. We can use information about what area of the face the action unit affects to estimate which model would be best to use, in order for the most accurate results.

5.1.6 Discussion

In this section, we explored the difference between face perception and various error metrics. We found that there was a correlation between all of the error metrics we tested and the perceptual results, with RMS and Hausdorff correlating the best. This suggests that perceived difference of facial expressions can be estimated from error metrics. However, the correlation is not perfect when analysing all blendshapes together, and increases in almost all cases if we look at per-blendshape correlations, shown in Table 5.5. This shows that the increase of perceived difference for each blendshape increases at different rates depending on action unit. A visualisation of this can be seen in Figure 5.15.

A quick further investigation into the direction of movement for each action unit showed that the perceptual difference correlated better with x- and y-axis movements, and not z-axis movements (see Table 5.14). This is likely due to the viewpoint of the characters in our experiment, as they were viewed front-on.

There was no large difference in correlations when viewed at a per-Race or per-Sex level, implying that our results were generally consistent across characters.

Both the Eyes Closed and Eyes Opened AUs were consistently given lower error ratings than perceptual ratings, showing that we are relatively more sensitive to eye AUs than their error metrics indicate, in contrast to other areas of the face.

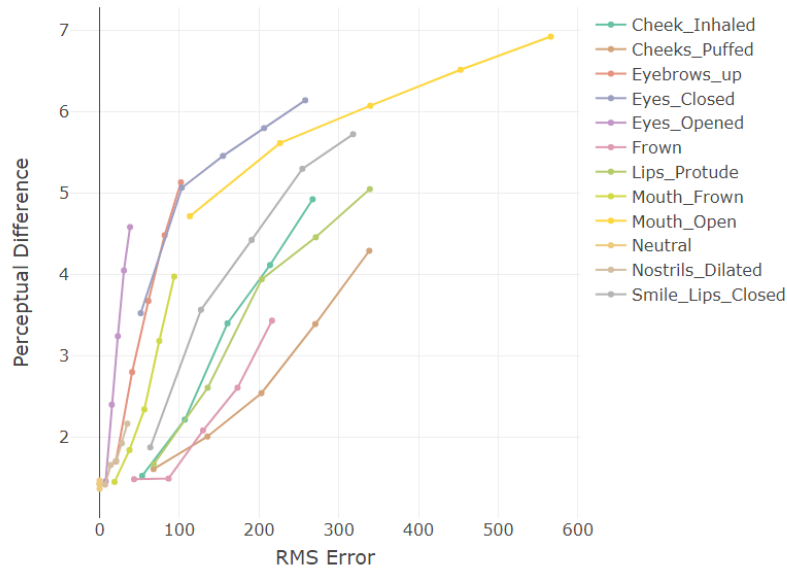


Figure 5.15: Graph showing the increase in perceived difference as RMS error increases for each blendshape. The five points in each line indicate the five activation levels of each blendshape.

Frown was one of the least perceptually different AUs, however it had similar geometric error values compared to other blendshapes (see Figure 5.11), and had either the highest or second-highest error using image-based metrics (see Figure 5.12). Interestingly, the opposing eyebrow AU Eyebrows Up was rated higher than Frown perceptually, but equal or below in every error metric. The mismatch in error and perceptual difference results for the eye and eyebrow area indicate that the importance of these are not well estimated by our metrics and should be weighted manually.

The average perceptual difference of each AU, shown in Table 5.12, gives us a basis for a perceptually-based blendshape ordering. This is beneficial for tasks that require an order of blendshapes, such as importance ordering of blendshapes, such as example creation for blendshape transfer (Section 3.2), or level of detail blendshape reduction methods (Section 4.1).

As well, our observations on which shapes are most perceptually different and how this correlates to various error metrics could be used to estimate the importance of other blendshapes which are not investigated in this section. For this purpose, we provide a table of orderings based on our metrics in Table 5.3. While our Expression Packing algorithm in Section 3.2 is automatic and does not include information about which areas of the face each blendshape affects, this information could be included through an automatic detection method, which would allow us to account for face perception in blendshape importance ordering for training expression creation as well. For our work in Chapter 4, these tables could be used to improve the automatic importance ordering done by the various metrics, by showing which AUs and which areas of the face specifically require manual reordering (either raising or lowering their relative importance) in order to best match perceptual results.

Our linear regression with RMS (shown in Figure 5.14, left) show quite sharp increases in perceptual difference as RMS error increases for Eyes Opened, Eyebrows Up, and Mouth Frown; while Smile Lips Closed, Cheek Inhaled, Cheeks Puffed, Frown, Lips Protrude, and Eyes Closed all had smaller increases in Perceptual Difference as RMS increased. Mouth Open had the smallest slope, although it also had the largest RMS at every level of activation, so this may be explained by a plateau effect. The Nostrils Dilated AU had a large slope, but the increase in both RMS and perceptual difference was quite small, so this area may need to be investigated more with larger expressions.

It is not clear whether we can extrapolate information about blendshapes that lie outside of the tested RMS ranges from this model. Our stimuli showed the maximum level of activation of certain blendshapes as posed by an actor, therefore predicting the perceptual difference of a blendshape with a higher RMS than the maximum we tested is not sensible, as that implies the shape is outside the natural range of motion

| Blendshape | Difference |
|-------------------|------------|
| Mouth Open | 5.97 |
| Eyes Closed | 5.2 |
| Smile Lips Closed | 4.18 |
| Eyebrows Up | 3.56 |
| Lips Protude | 3.55 |
| Cheek Inhaled | 3.24 |
| Eyes Opened | 3.15 |
| Cheeks Puffed | 2.77 |
| Mouth Frown | 2.56 |
| Frown | 2.22 |
| Nostrils Dilated | 1.78 |

Table 5.12: The blendshapes ordered by average perceptual difference.

for a human face. However, larger RMS errors could be caused by larger features of a different actor's face. Another experiment would be necessary to evaluate the impact of this.

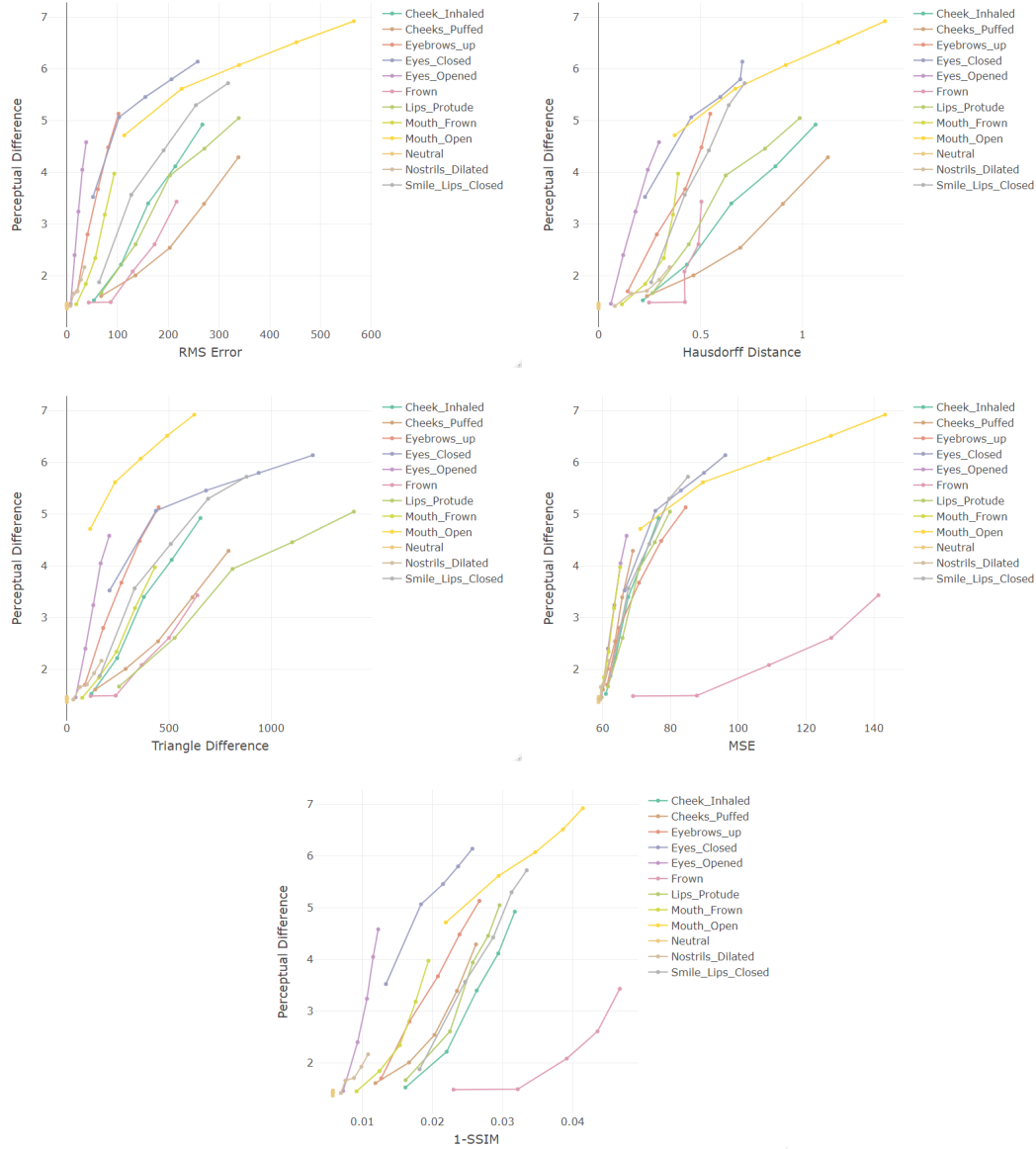


Figure 5.16: Graphs showing the increase in perceived difference as each of our error metrics increases for each blendshape. The five points in each line indicate the five activation levels of each blendshape.

| AU Name | x | y | z |
|-------------------|--------|--------|--------|
| Eyes Closed | 158.43 | 352.20 | 112.32 |
| Eyes Opened | 13.47 | 59.38 | 16.09 |
| Eyebrows Up | 52.71 | 349.16 | 82.84 |
| Frown | 89.92 | 74.3 | 105.05 |
| Nostrils Dilated | 45.86 | 19.77 | 18.69 |
| Cheeks Puffed | 139.12 | 115.17 | 526.33 |
| Cheek Inhaled | 292.24 | 224.94 | 126.76 |
| Mouth Open | 157.04 | 792.38 | 487.33 |
| Smile Lips Closed | 258.38 | 379.56 | 232.3 |
| Mouth Frown | 57.8 | 107.45 | 75.73 |
| Lips Protrude | 188.32 | 311.53 | 400.1 |

Table 5.13: Sum of the absolute values of each component of each vertex difference of each blendshape from the neutral.

| Axis | Correlation | p-value |
|------|-------------|---------|
| x | 0.64 | 0.00 |
| y | 0.86 | 0.00 |
| z | 0.57 | 0.00 |

Table 5.14: Correlations between the X, Y and Z components defined in Table 5.13 and the similarity results.

Chapter 6

Discussion

In this thesis we investigated ways to optimise the facial animation pipeline at both the animation and rig creation stages, identifying a common factor of blendshape importance ordering as a necessary component for optimisation. Our main contribution was a novel algorithm for suggesting input for example-based blendshape transfer algorithms, allowing for more efficient creation of high-quality, personalised blendshape rigs. We also contributed a GPU blendshape animation technique which, when paired with computer shaders and a blendshape reduction level of detail method, allows for large numbers of characters to be animated using blendshapes in a single scene. We conducted a perceptual analysis of the various numeric metrics suggested as ordering methods, as well as providing a perceptual ordering based on our observations. In this chapter, we provide a summary of our conclusions and a discussion of the implication of the results of this thesis as a whole.

6.1 Research Contributions

We presented, to the best of our knowledge, the first training example suggestion algorithm for example-based blendshape transfer methods, which we call Expression Packing (Chapter 3). Our algorithm takes a novel approach by defining the character mesh as a set of vertices, and each blendshape as a subset of those vertices, based on the area of the face which it affects. Expressions may then intuitively be created by solving the Set Packing problem using linear programming methods. We reduce the complexity by only allowing fully activated blendshapes to be included, making it an integer linear programming problem.

This algorithm takes into account the source and target character geometry, as well as the geometry of the blendshape set to be transferred, and created expressions which are densely packed with the necessary information to improve the blendshape transfer. The expressions are produced in order of importance, i.e. the first expression produced will provide the most important information.

As part of the evaluation of this algorithm we focussed on using high-quality virtual characters to replicate real world test cases for blendshape transfer. We considered this integral, as our previous research (Section 3.1) showed that lower quality meshes had issues with artifacts when used with blendshape transfer.

One novelty of this algorithm is the idea of blendshape importance which is integral to the creation of example expressions. Blendshapes are ordered by their overall displacement, as well as by the triangle distortion between the source and target character models in the area affected by the blendshape, as these are factors that can cause errors in blendshape transfer when example expressions are not present.

Notably, we include a symmetry constraint in our algorithm to ensure the created

expressions are feasible to pose by a human, as training expressions are likely to be created by scanning an actor.

We also provide a method for GPU blendshape animation to allow for high-quality blendshape rigs with large numbers of vertices and blendshapes to be animated on the GPU in real time (Chapter 4). We achieve this by passing all of the blendshape and animation information to the GPU as a texture using Texture Buffer Objects. We expand upon this and present a compute shader method, which decouples the animation and rendering calculations, allowing for more control over these individual areas. We compare the typical CPU blendshape animation technique with our new GPU method as well as our GPU method with compute shaders and found substantial computation time reductions using our GPU method. While the compute shader method was slower than the non-compute shader method in most cases, we found it to be faster in the case of instanced crowd animation.

For further optimisation, we discuss a blendshape reduction level of detail technique for GPU blendshape animation. Reducing the amount of information to be sent to and calculated by the GPU naturally increases performance, the reduction in quality caused by removing blendshapes from a rig can be severe. In order to mitigate this loss of quality, we propose a number of methods to order blendshapes by importance. By removing blendshapes deemed least important, we can reduce the loss in quality considerably.

In our perceptual experiment in Chapter 5, we conduct an experiment to investigate the perceptual importance of blendshape at various levels of activation. We first analyse the data using an ANOVA to identify where significant differences appear in the data. From this we concluded that the eyes closed and mouth open expressions were significantly more important perceptually than the other expressions we

tested. The smile and eyebrow frown expressions were also important. Notably, the eye expressions were consistently rated lower by our metrics relative to their perceptual importance. We also concluded that our results were roughly generalisable across the set of characters that we used.

We used a Generalised Linear Model (GLM) to fit our data given each stimulus's perceptual difference and its RMS error. Using this model, we can estimate the perceptual difference a new blendshape will have given its RMS error, and we can predict this more accurately if we have information about what type of blendshape it is.

6.2 Methodology Contributions

We found that, while using error metrics on blendshapes compared to the neutral face is a viable way to define their perceptual importance, it cannot be done naively, as the values for eye and eyebrow blendshapes are often weighted incorrectly. While the metrics discussed in this thesis are commonly used in geometry and image processing for measuring error (e.g. [CLL⁺13]), they are not immediately applicable to perceptual ordering tasks.

In order to get generalisable results, we used a diverse set of characters as stimuli in our perceptual experiment in Chapter 5, including Asian, Black, and White characters with various skin tones. It is worth noting that a large amount of psychology research on perception of faces does not include diverse stimuli sets. We found, however, that some of our choices for displaying the characters could have impacted our results. In this instance, our contribution is a face perception study of virtual characters with a focus on stimuli diversity.

6.3 Material Contributions

We provide the code used for our Expression Packing algorithm from Chapter 3, allowing anyone to run our algorithm given source rig with blendshapes and target character mesh which are in correspondence.

For stimuli creation in Chapter 5, we required high-quality stimuli with shared topology and blendshapes. We provide details on how we created these stimuli such that our stimuli creation pipeline may be reproduced to allow for high quality stimuli to be used in other experiments.

We share the data collected from the perceptual experiment in Chapter 5 along with the code used to analyse it. This can be used as a teaching aid for statistics classes, and the data can be used either to create or test models.

We provide the details of our Generalised Linear Model, created using our perceptual experiment data and geometric error, which is beneficial as a predictor for perceptual importance of blendshapes.

6.4 Guidelines for Character Design

Throughout this thesis we have discussed the idea of blendshape importance as an essential metric for optimisation of the facial animation pipeline. In Chapter 4 we found that error metrics gave low importance to the eyes, which is contradictory to perceptual research showing that the eyes are a very perceptually salient area of the face. We identified through perceptual experimentation that small movements such as those around the eyes and eyebrows are often given relatively low values when importance is calculated automatically using geometry-based or image-based metrics, while

their perceptual importance is relatively much higher. With this in mind, we propose that these blendshapes should be given a higher importance than those assigned using automatic methods, and should not be removed without careful consideration when reducing blendshape sets.

We propose the use of a perceptually-based metric which can account for the specificity of face-perception, rather than purely numeric methods. However, this type of method would require information about which blendshapes affect which area of the face, which is information we did not have in Chapters 3 and 4. An automatic method for identifying the area of the face that each blendshape impacts would be beneficial to this process. With this information, we could appropriately determine the perceptual importance of blendshapes, accounting for both their displacement in terms of vertex movement across the mesh, and for their perceptual impact based on the type of expression being shown.

Chapter 7

Future Work

There were a number of avenues of research we did not explore throughout this thesis due to limitations in time and resources during each project. In this chapter, we briefly discuss a few options for future work.

Our method in Section 4.1 considers the perceived prominence of individual blendshapes in the rig for blendshape reduction, but it is possible that further reductions would be possible depending on the animation type (e.g. conversation, emotion, etc.), as different blendshapes are important depending on the type of animation displayed.

Our initial goal for our work in Section 3.1 was to identify what facial expressions are important to use as training when using Example-Based Facial Rigging to create facial rigs. While our work here has indicated certain parts of the face might require more attention when automatically creating blendshapes, there is room for more investigation into this topic. We suggest an algorithm for optimal training example creation in Section 3.2, however a thorough perceptual investigation using the output of this algorithm and other training example sets would be necessary to undeniably verify the usefulness of our algorithm.

We would like to apply our method from Section 3.2 to a number of characters with equivalent FACS-based rigs in order to propose a generalised expression set which would be applicable to all FACS-based rigs. However, a general set of reference expressions comes at the cost of not being optimal for each individual, because the triangle distortion metric between neutral expressions must be ignored.

In each of our implementation chapters (Chapters 3 and 4), we use numeric metrics to define blendshape importance and discuss how these would need to be further investigated through perceptual experiments. We begin this perceptual verification in Chapter 5, however there are many other aspects of human perception and blendshape animation to delve in to in order to fully round out the research carried out in this thesis.

It is well documented that face perception is a special type of perception that humans are particularly attuned to [BY13, FWDT98, KMC97]. Future work into separating our results from Section 5.1 from non-face specific perception would be worthwhile, for example through experiments with upside-down faces [FWDT98, HUC⁺99].

As our work study into investigating the perceived importance of facial blendshapes, we limited our study to static expressions of single blendshapes. Naturally, perception of animated faces with combined expressions would be more complicated, particularly since specific AUs are important for the perception of emotions (e.g., AU 7 Lid Tightener for anger [WVK⁺17]). It might be the case that even if these AUs are not perceptually important according to our approach, removing them from the rig might alter the interpretation of emotion of the virtual human, which is something that we will study in future work.

We present a model for predicting perceptual impact of blendshapes, however we could not confidently say that these models could predict the perceptual impact of

blendshapes outside of the displacement ranges which we tested. We would like to investigate this further by examining how well the model predicts perceptual impact on other characters.

We noted that diversity is missing from much of the psychology and computer vision research on recognition and perception of faces. Therefore, we included Asian, Black, and White characters with various skin tones. We found an effect of race (see Section 5.1.3), which showed that certain expressions were less perceptible on our Black characters. This result may have been due to our predominantly European and Asian participant pool, indicating that differences in perception of Black characters could be caused by the other-race effect [LJC91, WT03]. However, we did find that mouth and eye expressions were perceived similarly across race, indicating that the effect was not strong. A more diverse participant pool would be interesting to test for future experiments. The effect of race may also be caused by the chosen background and lighting of the characters, which could be biased towards a clearer representation of lighter skinned characters.

We also discuss leveraging blendshape rig reduction for level of detail as a possible application of our work, which means a further experiment varying the viewing distance of the character on the screen would be imperative. However, while animation and distance are important variables to analyse, our current study should already inform the choices for which AUs to include or remove when memory and computation power are limited. It is likely that our results would be conservative estimates for blendshape reduction at further viewing distances.

Another metric which could be investigated is that of eye gaze, by using eye-tracking to determine the blendshapes that elicit greater visual attention and contrast this with their computed saliency based on our method.

The level at which an expression becomes perceptually different from the neutral varied across the blendshapes we chose for this experiment. Further research into the exact point at which that difference occurs, and the differences in perception at a barely perceptible level between expressions would be interesting.

Our experiment in Chapter 5 displayed each blendshape statically with a fixed viewpoint, showing each stimulus front-on. Further investigation into the direction of movement for each blendshape indicates that movement in the x and y-axes, i.e. movement up, down, left, or right on the screen, correlated better with our perceptual results than movement in the z-axis, which would be movement toward or away from the participants. Further work should be done to investigate the impact of facial expressions in 3/4 or profile views.

One use case we posited for blendshape importance is the reduction of blendshapes in a rig while maintaining the expressivity and accuracy of the animations that that rig could produce. Our work focussed on the perceptual saliency of individual facial movements with no context of what animation those movements would be used for. If we take into account the animation that a character will be portraying before removing certain blendshapes from a rig, then the importance of each blendshape would likely change drastically. For example, blendshapes used to create appropriate mouth shapes for speech would likely be the first to be removed if the animation contained no speech. Similarly, blendshapes which are used only for extreme emotional expressions would likely be removed if the animation contained only neutral speech. While we intentionally tried to generalise our results as much as possible in Chapter 5, it would be important to investigate the impact of these results in real-world animation cases. This reduction could take place before the animation is recorded, in which case we would attempt to predict the blendshapes which would be used given the nature of the

animation, or after the animation has been recorded and displayed on the character, in which case we would remove the blendshapes least used in order to minimise the impact on the performance.

Our work may be helpful for optimising marker placement for animation, as those markers which are driving more salient areas of the face could be prioritised. This would require investigation into animated stimuli first however.

Human face perception is a highly interesting topic, covering many different areas that we could use as inspiration for further perceptual experimentation. Limiting the scope to the topic of this thesis, optimisation of the facial animation pipeline, still offers a wide variety of research questions to follow. With the constant improvements in virtual human technology, there will always be a need for perceptual verification of new methods, as well as potential perceptually guided improvements to these methods, which we hope this thesis has made clear.

Bibliography

- [3la18] Siren. <https://www.3lateral.com/projects/siren.html>, Mar 2018.
- [Ado06] Ralph Adolphs. Perception and emotion: How we recognize facial expressions. *Current directions in psychological science*, 15(5):222–226, 2006.
- [Ale01] Marc Alexa. Local control for mesh morphing. In *Proceedings of the International Conference on Shape Modeling & Applications*, SMI '01, page 209, USA, 2001. IEEE Computer Society.
- [Ari06] Okan Arikan. Compression of motion capture databases. In *ACM SIGGRAPH 2006 Papers*, pages 890–897. 2006.
- [ASCE02] Nicolas Aspert, Diego Santa Cruz, and Touradj Ebrahimi. MESH: measuring errors between surfaces using the Hausdorff distance. In *ICME (1)*, pages 705–708, 2002.
- [ATT12] Hillel Aviezer, Yaacov Trope, and Alexander Todorov. Body cues, not facial expressions, discriminate between intense positive and negative emotions. *Science*, 338(6111):1225–1229, 2012.
- [Aud18] William Audureau. Quantic dream, un fleuron du jeu vidéo français aux méthodes de management contestées. <https://>

`//www.lemonde.fr/pixels/article/2018/01/14/quantic-dream-un-fleuron-du-jeu-video-francais-aux-methodes-de-management-contestees_5241506_4408996.html`, 2018.

- [Aut16] Autodesk. Maya, 2016.
- [AWLB17] Jascha Achenbach, Thomas Waltemate, Marc Erich Latoschik, and Mario Botsch. Fast generation of realistic virtual humans. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, pages 1–10, 2017.
- [BCWG09] Mirela Ben-Chen, Ofir Weber, and Craig Gotsman. Spatial deformation transfer. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 67–74. ACM, 2009.
- [BDBP12] Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. Superfaces: A super-resolution model for 3d faces. In *European Conference on Computer Vision*, pages 73–82. Springer, 2012.
- [Bec94] Dominique Bechmann. Space deformation models survey. *Computers & Graphics*, 18(4):571–586, 1994.
- [Bet12] Vinay Bettadapura. Face expression recognition and analysis: the state of the art. *arXiv preprint arXiv:1203.6722*, 2012.
- [BGVS19] Jilliam María Díaz Barros, Vladislav Golyanik, Kiran Varanasi, and Didier Stricker. Face it!: A pipeline for real-time performance-driven facial animation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2209–2213. IEEE, 2019.

- [BI12] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):185–207, 2012.
- [BiRZL⁺17] Roger Blanco i Ribera, Eduard Zell, JP Lewis, Junyong Noh, and Mario Botsch. Facial retargeting with automatic range of motion alignment. *ACM Transactions on Graphics (TOG)*, 36(4):154, 2017.
- [Bla52] H Richard Blackwell. Studies of psychophysical methods for measuring visual thresholds. *JOSA*, 42(9):606–616, 1952.
- [BN10] Benjamin Balas and Charles A Nelson. The role of face shape and pigmentation in other-race face perception: An electrophysiological study. *Neuropsychologia*, 48(2):498–506, 2010.
- [BPT05] Andrew P Bayliss, Giuseppe di Pellegrino, and Steven P Tipper. Sex differences in eye gaze and symbolic cueing of attention. *The Quarterly Journal of Experimental Psychology*, 58(4):631–650, 2005.
- [BS07] Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. *IEEE transactions on visualization and computer graphics*, 14(1):213–230, 2007.
- [BSS07] Tobias Brosch, David Sander, and Klaus R Scherer. That baby caught my eye... attention capture by infant faces. 2007.
- [BSSM⁺13] Dario Bombari, Petra C Schmid, Marianne Schmid Mast, Sandra Birri, Fred W Mast, and Janek S Lobmaier. Emotion recognition: The role

of featural and configural face information. *The Quarterly Journal of Experimental Psychology*, 66(12):2426–2442, 2013.

- [BWP13] Sofien Bouaziz, Yangang Wang, and Mark Pauly. Online modeling for realtime facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):40:1–40:10, 2013.
- [BY86] Vicki Bruce and Andy Young. Understanding face recognition. *British journal of psychology*, 77(3):305–327, 1986.
- [BY12] Vicki Bruce and Andrew W Young. *Face perception*. Psychology Press, 2012.
- [BY13] Vicki Bruce and Andy Young. *Face perception*. Psychology Press, 2013.
- [CBH⁺16] Antoine Coutrot, Nicola Binetti, Charlotte Harrison, Isabelle Mareschal, and Alan Johnston. Face exploration dynamics differentiate men and women. *Journal of vision*, 16(14):16–16, 2016.
- [CBZB15] Chen Cao, Derek Bradley, Kun Zhou, and Thabo Beeler. Real-time high-fidelity facial performance capture. *ACM Transactions on Graphics (ToG)*, 34(4):1–9, 2015.
- [CC13] Francesco Ceccarini and Corrado Caudek. Anger superiority effect: The importance of dynamic emotional facial expressions. *Visual Cognition*, 21(4):498–540, 2013.
- [CET98] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. In *European conference on computer vision*, pages 484–498. Springer, 1998.

- [CET01] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.
- [CFA⁺16] Dan Casas, Andrew Feng, Oleg Alexander, Graham Fyffe, Paul Debevec, Ryosuke Ichikari, Hao Li, Kyle Olszewski, Evan Suma, and Ari Shapiro. Rapid photorealistic blendshape modeling from rgb-d sensors. *Computer Animation and Virtual Worlds*, 2016.
- [CGZ⁺18] Chaona Chen, Oliver GB Garrod, Jiayu Zhan, Jonas Beskow, Philippe G Schyns, and Rachael E Jack. Reverse engineering psychologically valid facial expressions of emotion into social robots. In *Int. Conf. on Automatic Face & Gesture Recognition*, pages 448–452, 2018.
- [CHZ14] Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics (TOG)*, 33(4):43:1–43:10, 2014.
- [CKH10] Darren Cosker, Eva Krumhuber, and Adrian Hilton. Perception of linear and nonlinear motion properties using a face validated 3d facial model. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, pages 101–108, 2010.
- [CKH11] D. Cosker, E. Krumhuber, , and A. Hilton. A face valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling. pages 2296–2303, 2011.
- [CLL⁺13] Massimiliano Corsini, Mohamed-Chaker Larabi, Guillaume Lavoué, Oldřich Petřík, Libor Váša, and Kai Wang. Perceptual metrics for static

and dynamic triangle meshes. In *Computer Graphics Forum*, volume 32, pages 101–125. Wiley Online Library, 2013.

- [CN08] Manuel G Calvo and Lauri Nummenmaa. Detection of emotional faces: salient physical features guide effective visual search. *Journal of Experimental Psychology: General*, 137(3):471, 2008.
- [CN16] Manuel G Calvo and Lauri Nummenmaa. Perceptual and affective mechanisms in facial expression recognition: An integrative review. *Cognition and Emotion*, 30(6):1081–1106, 2016.
- [Com18] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.
- [CPB⁺94] Justine Cassell, Catherine Pelachaud, Norman Badler, Mark Steedman, Brett Achorn, Tripp Becket, Brett Douville, Scott Prevost, and Matthew Stone. Animated conversation: rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 413–420, 1994.
- [CPKC09] Viola Macchi Cassia, Marta Picozzi, Dana Kuefner, and Monica Casati. Short article: Why mix-ups don’t happen in the nursery: Evidence for an experience-based interpretation of the other-age effect. *Quarterly Journal of Experimental Psychology*, 62(6):1099–1107, 2009.
- [CWLZ13] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 3d shape regression

- for real-time facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013.
- [CWZ⁺14] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Face-warehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, March 2014.
- [CZLK99] Jeffrey F Cohn, Adena J Zlochower, James Lien, and Takeo Kanade. Automated face analysis by feature point tracking has high concurrent validity with manual face coding. *Psychophysiology*, 36(1):35–43, 1999.
- [DB08] A. J. Dobson and A. G. Barnett. *An Introduction to Generalized Linear Models*. CRC Press, Third Edition, 2008.
- [DBS18] Katharina Dobs, Isabelle Bühlhoff, and Johannes Schultz. Use and usefulness of dynamic face stimuli for face perception studies—a review of behavioral findings and methodology. *Frontiers in psychology*, 9:1355, 2018.
- [Deb12] Paul Debevec. The light stages and their applications to photoreal digital actors. *SIGGRAPH Asia*, 2(4), 2012.
- [DERC⁺11] Ricardo L Parreira Duarte, Abdennour El Rhalibi, Christopher Carter, Simon Cooper, and Madjid Merabti. An MPEG-4 Compliant Quadric-Based Surface Adaptive LOD. Technical report, School of Computing and Mathematical Sciences, Liverpool John Moores University, UK, 2011.

- [DI419] DI4D. Di4d. <https://www.di4d.com/>, 2019.
- [Did19] Didimo. The breathtaking reality of your digital you. <https://mydidimo.com/>, 2019.
- [DM08] Zhigang Deng and Xiaohan Ma. Perceptually guided expressive facial animation. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 67–76. Eurographics Association, 2008.
- [DMOB10] Ludovic Dutreuve, Alexandre Meyer, Veronica Orvalho, and Saida Bouakaz. Easy rigging of face by automatic registration and transfer of skinning parameters. In *International Conference on Computer Vision and Graphics*, pages 333–341. Springer, 2010.
- [DN08] Zhigang Deng and Junyong Noh. Computer facial animation: A survey. In *Data-driven 3D facial animation*, pages 1–28. Springer, 2008.
- [DRPP+03] Fiorella De Rosis, Catherine Pelachaud, Isabella Poggi, Valeria Carofiglio, and Berardina De Carolis. From greta’s mind to her face: modelling the dynamics of affective states in a conversational embodied agent. *International journal of human-computer studies*, 59(1-2):81–118, 2003.
- [Dud07] Bryan Dudash. Skinned instancing. Technical report, NVidia, 2007.
- [DVM09] Richard Drake, A Wayne Vogl, and Adam WM Mitchell. *Gray’s Anatomy for Students E-Book*. Elsevier Health Sciences, 2009.

- [EA11] Hedwig Eisenbarth and Georg W Alpers. Happy mouth and sad eyes: scanning emotional facial expressions. *Emotion*, 11(4):860, 2011.
- [EC11] Paul Ekman and Daniel Cordaro. What is meant by calling emotions basic. *Emotion review*, 3(4):364–370, 2011.
- [EF78] Paul Ekman and Wallace V. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, 1978.
- [EHEM13] Cathy Ennis, Ludovic Hoyet, Arjan Egges, and Rachel McDonnell. Emotion capture: Emotionally expressive characters for games. In *Proceedings of Motion on Games*, MIG '13, pages 31:53–31:60, New York, NY, USA, 2013. ACM.
- [EI15] Violeta Enea and Sorina Iancu. Processing emotional body expressions: state-of-the-art. *Social Neuroscience*, 0(0):1–12, 2015. PMID: 26513592.
- [Ekm79] Paul Ekman. About brows. *Emotional and conversational signals in: von Cranach, M./Foppa, K. ua (Hrsg.)(1979): Human Ethology*, Cambridge, 1979.
- [Ekm92a] Paul Ekman. Are there basic emotions? 1992.
- [Ekm92b] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.
- [Ekm20] Paul Ekman. Universal facial expressions: Are facial expressions universal? <https://www.paulekman.com/resources/universal-facial-expressions/>, 2020.

- [EMH14] Christopher Evans, Lars Martinsson, and Sascha Herfort. Building an empire: asset production in Ryse. In *ACM SIGGRAPH 2014 Courses*, page 18. ACM, 2014.
- [ESV99] Jihad El-Sana and Amitabh Varshney. Generalized view-dependent simplification. In *Computer Graphics Forum*, volume 18, pages 83–94. Wiley Online Library, 1999.
- [Fac16] Faceware. Faceware Technologies Inc. <http://facewaretech.com/>, 2016.
- [FBT07] Alexandra Frischen, Andrew P Bayliss, and Steven P Tipper. Gaze cueing of attention: visual attention, social cognition, and individual differences. *Psychological bulletin*, 133(4):694, 2007.
- [FHCP19] Zhenfeng Fan, Xiyuan Hu, Chen Chen, and Silong Peng. Boosting local shape matching for dense 3d face correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10944–10954, 2019.
- [FNH⁺17] G. Fyffe, K. Nagano, L. Huynh, S. Saito, J. Busch, A. Jones, H. Li, and P. Debevec. Multi-view stereo on consistent face topology. *Comput. Graph. Forum*, 36(2):295–309, May 2017.
- [Fri88] Nico H Frijda. The laws of emotion. *American psychologist*, 43(5):349, 1988.
- [FWDT98] Martha J Farah, Kevin D Wilson, Maxwell Drain, and James N Tanaka. What is” special” about face perception? *Psychological review*, 105(3):482, 1998.

- [GH97] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216. ACM Press/Addison-Wesley Publishing Co., 1997.
- [Gle98] Michael Gleicher. Retargetting motion to new characters. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 33–42, 1998.
- [GPD09] Qin Gu, Jingliang Peng, and Zhigang Deng. Compression of human motion capture data using motion pattern indexing. In *Computer Graphics Forum*, volume 28, pages 1–12. Wiley Online Library, 2009.
- [GVWT13] Pablo Garrido, Levi Valgaerts, Chenglei Wu, and Christian Theobalt. Reconstructing detailed dynamic face geometry from monocular video. *ACM Trans. Graph.*, 32(6):158–1, 2013.
- [GZC⁺16] Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. Reconstruction of personalized 3d face rigs from monocular video. *ACM Transactions on Graphics (TOG)*, 35(3):28:1–28:15, May 2016.
- [HAJK04] Ursula Hess, Reginald B Adams Jr, and Robert E Kleck. Facial appearance, gender, and emotion expression. *Emotion*, 4(4):378, 2004.
- [HCKH14] Jennifer Hyde, Elizabeth J. Carter, Sara Kiesler, and Jessica K. Hodgins. Assessing naturalness and emotional intensity: A perceptual study of animated facial motion. In *Proceedings of the ACM Symposium on*

Applied Perception, SAP '14, pages 15–22, New York, NY, USA, 2014. ACM.

- [HG15] Naomi Harte and Eoin Gillen. TCD-TIMIT: An audio-visual corpus of continuous speech. *IEEE Transactions on Multimedia*, 17(5):603–615, 2015.
- [HJO⁺10] Jessica Hodgins, Sophie Jörg, Carol O’Sullivan, Sang Il Park, and Moshe Mahler. The saliency of anomalies in animated human characters. *ACM Transactions on Applied Perception (TAP)*, 7(4):22, 2010.
- [HRMO12] Ludovic Hoyet, Kenneth Ryall, Rachel McDonnell, and Carol O’Sullivan. Sleight of hand: perception of finger motion from reduced marker sets. In *Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games*, pages 79–86, 2012.
- [HSL06] Stefan G Hofmann, Michael Suvak, and Brett T Litz. Sex differences in face recognition and influence of facial affect. *Personality and Individual Differences*, 40(8):1683–1690, 2006.
- [HSW⁺17] Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. Avatar digitization from a single image for real-time rendering. *ACM Transactions on Graphics (TOG)*, 36(6):195:1–195:14, November 2017.
- [HUC⁺99] James V Haxby, Leslie G Ungerleider, Vincent P Clark, Jennifer L Schouten, Elizabeth A Hoffman, and Alex Martin. The effect of face inversion on activity in human neural systems for face and object perception. *Neuron*, 22(1):189–199, 1999.

- [IBP15] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (TOG)*, 34(4):45:1–45:14, July 2015.
- [IKNDP16] Alexandru-Eugen Ichim, Ladislav Kavan, Merlin Nimier-David, and Mark Pauly. Building and animating user-specific volumetric face rigs. In *Proc. Symp. on Computer Animation*, pages 107–117, 2016.
- [itS20] itSeez3D. Turn your mobile device into a powerful 3d scanner. <https://itseez3d.com/>, 2020.
- [Iza11] Carroll E Izard. Forms and functions of emotions: Matters of emotion–cognition interactions. *Emotion Review*, 3(4):371–378, 2011.
- [JG07] Jutta Joormann and Ian H Gotlib. Selective attention to emotional faces following recovery from depression. *Journal of abnormal psychology*, 116(1):80, 2007.
- [JMD⁺07] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. Harmonic coordinates for character articulation. *ACM Transactions on Graphics (TOG)*, 26(3):71–es, 2007.
- [JS15] Rachael E Jack and Philippe G Schyns. The human face as a dynamic tool for social communication. *Current Biology*, 25(14):R621–R634, 2015.
- [JSW05] Tao Ju, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. In *ACM Siggraph 2005 Papers*, pages 561–566. 2005.

- [KAL⁺17] 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):1–12, 2017.
- [Kar72] Richard M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.
- [KG08] Scott Kircher and Michael Garland. Free-form motion processing. *ACM Transactions on Graphics (TOG)*, 27(2):12:1–12:13, May 2008.
- [KMC97] Nancy Kanwisher, Josh McDermott, and Marvin M Chun. The fusiform face area: a module in human extrastriate cortex specialized for face perception. *Journal of neuroscience*, 17(11):4302–4311, 1997.
- [Kor13] David Kordalewski. New greedy heuristics for set cover and set packing. *arXiv preprint arXiv:1305.3584*, 2013.
- [KVB⁺07] Ernst HW Koster, Bruno Verschuere, Benjamin Burssens, Roel Custers, and Geert Crombez. Attention for emotional faces under restricted awareness revisited: Do emotional faces automatically attract attention? *Emotion*, 7(2):285, 2007.
- [LA10] JP Lewis and Ken-ichi Anjyo. Direct manipulation blendshapes. *IEEE Computer Graphics and Applications*, (4):42–50, 2010.
- [LAR⁺14] John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H Pighin, and Zhigang Deng. Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)*, 1(8):2, 2014.

- [Lav11] Guillaume Lavoué. A multiscale metric for 3d mesh visual quality assessment. In *Computer Graphics Forum*, volume 30, pages 1427–1437. Wiley Online Library, 2011.
- [LBB⁺17] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics (TOG)*, 36(6):194:1–194:17, November 2017.
- [LCB99] Karen Lander, Fiona Christie, and Vicki Bruce. The role of movement in the recognition of famous faces. *Memory & cognition*, 27(6):974–985, 1999.
- [LCW06] Karen Lander, Lewis Chuang, and Lee Wickham. Recognizing face identity from natural and morphed smiles. *Quarterly Journal of Experimental Psychology*, 59(5):801–808, 2006.
- [Lev11] Robert W Levenson. Basic emotion questions. *Emotion review*, 3(4):379–386, 2011.
- [LH04] Jukka M Leppänen and Jari K Hietanen. Positive facial expressions are recognized faster than negative facial expressions, but why? *Psychological research*, 69(1-2):22–29, 2004.
- [LHJB10] Michael Lewis, Jeannette M Haviland-Jones, and Lisa Feldman Barrett. *Handbook of emotions*. Guilford Press, 2010.
- [Lik32] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

- [LJC91] D Stephen Lindsay, Philip C Jack, and Marcus A Christian. Other-race face perception. *Journal of applied psychology*, 76(4):587, 1991.
- [LKA⁺17] Samuli Laine, Tero Karras, Timo Aila, Antti Herva, Shunsuke Saito, Ronald Yu, Hao Li, and Jaakko Lehtinen. Production-level facial performance capture using deep convolutional neural networks. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 1–10, 2017.
- [LKC07] Yaron Lipman, Johannes Kopf, Daniel Cohen-Or, and David Levin. Gpu-assisted positive mean value coordinates for mesh deformations. In *Symposium on geometry processing*, 2007.
- [LMD12] Binh H Le, Xiaohan Ma, and Zhigang Deng. Live speech driven head-and-eye motion generators. *IEEE transactions on visualization and computer graphics*, 18(11):1902–1914, 2012.
- [LMS10] Francisco Lopez, Ramon Molla, and Veronica Sundstedt. Exploring peripheral LOD change detections during interactive gaming tasks. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, pages 73–80. ACM, 2010.
- [Loo20] Loom.ai. 3d avatar platform for enterprise and developers. <https://loomai.com/>, 2020.
- [Lor07] Tristan Lorach. *DirectX 10 Blend Shapes: Breaking the Limits*, chapter 3, pages 53–67. GPU Gems 3, Addison-Wesley Professional, 2007.

- [LSC⁺04] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rossi, and H. P. Seidel. Differential coordinates for interactive mesh editing. In *Proceedings Shape Modeling Applications, 2004.*, pages 181–190, June 2004.
- [LTO⁺15] Hao Li, Laura Trutoiu, Kyle Olszewski, Lingyu Wei, Tristan Trutna, Pei-Lun Hsieh, Aaron Nicholls, and Chongyang Ma. Facial performance sensing head-mounted display. *ACM Transactions on Graphics (TOG)*, 34(4):1–9, 2015.
- [LVJ05] Chang Ha Lee, Amitabh Varshney, and David W Jacobs. Mesh saliency. In *ACM SIGGRAPH 2005 Papers*, pages 659–666. 2005.
- [LWP10] Hao Li, Thibaut Weise, and Mark Pauly. Example-based facial rigging. In *ACM Transactions on Graphics (TOG)*, volume 29, page 32, 2010.
- [LYYB13] Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics (TOG)*, 32(4):42:1–42:10, 2013.
- [LZD13] Binh H Le, Mingyang Zhu, and Zhigang Deng. Marker optimization for facial motion acquisition and deformation. *IEEE transactions on visualization and computer graphics*, 19(11):1859–1871, 2013.
- [LZX⁺08] Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J Gortler. A local/global approach to mesh parameterization. In *Computer Graphics Forum*, volume 27, pages 1495–1504. Wiley Online Library, 2008.
- [MAF10] Christian Miller, Okan Arikan, and Don Fussell. Frankenrigs: building character rigs from multiple sources. In *Proceedings of the 2010 ACM*

SIGGRAPH symposium on Interactive 3D Graphics and Games, pages 31–38, 2010.

- [Mar03] Chris Maraffi. *Maya character creation: modeling and animation controls*. New Riders, 2003.
- [MB01] Christian A Meissner and John C Brigham. Thirty years of investigating the own-race bias in memory for faces: a meta-analytic review. *Psychology, Public Policy, and Law*, 7(1):3, 2001.
- [McC18] Caty McCarthy. Hellblade’s budget required ninja theory to use their own boardroom as a motion capture studio. <https://www.usgamer.net/articles/ninja-theory-boardroom-budget-hellblade-gdc-2018>, 2018.
- [MG03] Alex Mohr and Michael Gleicher. Deformation sensitive decimation. Technical report, University of Wisconsin Graphics Group, 2003.
- [MGP07] Niloy J. Mitra, Leonidas J. Guibas, and Mark Pauly. Symmetrization. *SIGGRAPH 2007*, 2007.
- [MHK99] Richard J Millar, JR Paul Hanna, and SM Kealy. A review of behavioural animation. *Computers & Graphics*, 23(1):127–143, 1999.
- [MIGC15] Jennifer Murphy, Alberta Ipser, Sebastian B Gaigg, and Richard Cook. Exemplar variance supports robust learning of facial identity. *Journal of Experimental Psychology: Human Perception and Performance*, 41(3):577, 2015.

- [MLD⁺08] Rachel McDonnell, Micheal Larkin, Simon Dobbyn, Stephen Collins, and Carol O’Sullivan. Clone attack! Perception of crowd variety. *ACM Transactions on Graphics*, 27(3):1–8, 2008.
- [MLD⁺16] Wan-Chun Ma, Mathieu Lamarre, Etienne Danvoye, Chongyang Ma, Manny Ko, Javier von der Pahlen, and Cyrus A Wilson. Semantically-aware blendshape rigs from facial performance measurements. In *SIGGRAPH ASIA 2016 Technical Briefs*, pages 1–4. 2016.
- [MMMY15] Soyogu Matsushita, Kazunori Morikawa, Saya Mitsuzane, and Haruna Yamanami. Eye shape illusions induced by eyebrow positions. *Perception*, 44(5):529–540, 2015.
- [MPB04] Karin Mogg, Pierre Philippot, and Brendan P Bradley. Selective attention to angry faces in clinical social phobia. *Journal of abnormal psychology*, 113(1):160, 2004.
- [NCWB08] Manfred Nusseck, Douglas W Cunningham, Christian Wallraven, and Heinrich H Bülthoff. The contribution of different facial regions to the recognition of conversational expressions. *Journal of vision*, 8(8):1–1, 2008.
- [NJ04] Kyunggun Na and Moonryul Jung. Hierarchical retargetting of fine facial motions. In *Computer Graphics Forum*, volume 23, pages 687–695. Wiley Online Library, 2004.
- [NO11] Vaidehi Natu and Alice J O’Toole. The neural processing of familiar and unfamiliar faces: A review and synopsis. *British journal of psychology*, 102(4):726–747, 2011.

- [NSX⁺18] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, Hao Li, Richard Roberts, et al. pagan: real-time avatars using dynamic textures. *ACM Trans. Graph.*, 37(6):258–1, 2018.
- [OBL19] Ipek Oruc, Benjamin Balas, and Michael S Landy. Face perception: A brief journey through recent discoveries and current directions, 2019.
- [OBP⁺12] Verónica Orvalho, Pedro Bastos, Frederic I Parke, Bruno Oliveira, and Xenxo Alvarez. A facial rigging survey. In *Eurographics State of the Art Reports*, pages 183–204, 2012.
- [Oh19] Sahuck Oh. A new triangular mesh repairing method using a mesh distortion energy minimization-based mesh flattening method. *Advances in Engineering Software*, 131:48–59, 2019.
- [Osi07] Jason Osipa. *Stop Staring: Facial Modeling and Animation Done Right*. Wiley Publishing, second edition, 2007.
- [OT90] Andrew Ortony and Terence J Turner. What’s basic about basic emotions? *Psychological review*, 97(3):315, 1990.
- [PC04] Romina Palermo and Max Coltheart. Photographs of facial expression: Accuracy, response times, and ratings of intensity. *Behavior Research Methods, Instruments, & Computers*, 36(4):634–638, 2004.
- [PHL⁺06] Frédéric Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, and David H Salesin. Synthesizing realistic facial expressions from photographs. In *ACM SIGGRAPH 2006 Courses*, page 19. ACM, 2006.

- [Pin19] Pinscreen. The most advanced ai-driven personalized avatars. <https://www.pinscreen.com/>, 2019.
- [PK80] Robert Plutchik and Henry Kellerman. *Emotion, theory, research, and experience*. Academic press, 1980.
- [PP02] Catherine Pelachaud and Isabella Poggi. Subtleties of facial expressions in embodied agents. *The Journal of Visualization and Computer Animation*, 13(5):301–312, 2002.
- [PR07] Romina Palermo and Gillian Rhodes. Are you always on my mind? a review of how face perception and attention interact. *Neuropsychologia*, 45(1):75–92, 2007.
- [PW11] Jaak Panksepp and Douglas Watt. What is basic about basic emotions? lasting lessons from affective neuroscience. *Emotion review*, 3(4):387–396, 2011.
- [RDK⁺16] Shridhar Ravikumar, Colin Davidson, Dmitry Kit, Neill DF Campbell, Luca Benedetti, and Darren Cosker. Reading between the dots: Combining 3d markers and faces classification for high-quality blendshape facial animation. In *Graphics Interface*, pages 143–151, 2016.
- [RGL15] Clément Reverdy, Sylvie Gibet, and Caroline Larboulette. Optimal marker set for motion capture of dynamical facial expressions. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, pages 31–36, 2015.
- [RP06] Mauricio Radovan and Laurette Pretorius. Facial animation in a nutshell:

past, present and future. In *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, pages 71–79, 2006.

- [SA12] Mark Segal and Kurt Akeley. The OpenGL Graphics System: A Specification (Version 4.3 (Core Profile)). Technical report, Khronos Group, 2012.
- [SAD⁺08] Arman Savran, Neşe Alyüz, Hamdi Dibeklioglu, Oya Çeliktutan, Berk Gökberk, Bülent Sankur, and Lale Akarun. Bosphorus database for 3d face analysis. *Biometrics and identity management*, pages 47–56, 2008.
- [Sai13] Jun Saito. Smooth contact-aware facial blendshapes transfer. In *Proceedings of the Symposium on Digital Production*, DigiPro '13, pages 7–12, 2013.
- [SCGS05] Marie L Smith, Garrison W Cottrell, Frédéric Gosselin, and Philippe G Schyns. Transmitting and decoding facial expressions. *Psychological science*, 16(3):184–189, 2005.
- [SCOL⁺04] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, page 175–184, New York, NY, USA, 2004. Association for Computing Machinery.
- [SCR⁺18] José Serra, Ozan Cetinaslan, Shridhar Ravikumar, Veronica Orvalho, and Darren Cosker. Easy generation of facial animation using motion

- graphs. In *Computer Graphics Forum*, volume 37, pages 97–111. Wiley Online Library, 2018.
- [SCSN11] Jaewon Song, Byungkuk Choi, Yeongho Seol, and Junyong Noh. Characteristic facial retargeting. *Computer Animation and Virtual Worlds*, 22(2-3):187–194, 2011.
- [Sey16] Mike Seymour. Put your (digital) game face on. <https://www.fxguide.com/featured/put-your-digital-game-face-on/>, 2016.
- [Sey18] Mike Seymour. Making thanos face the avengers, 2018.
- [Sey19] Mike Seymour. Weta digital’s remarkable face pipeline : Alita battle angel. <https://www.fxguide.com/featured/weta-digitals-remarkable-face-pipeline-alita-battle-angel/>, 2019.
- [SF07] Philip Shilane and Thomas Funkhouser. Distinctive regions of 3d surfaces. *ACM Transactions on Graphics (TOG)*, 26(2):7–es, 2007.
- [SGM16] Ramprakash Srinivasan, Julie D Golomb, and Aleix M Martinez. A neural basis of facial action recognition in humans. *Journal of Neuroscience*, 36(16):4434–4442, 2016.
- [SILN11] Jaewoo Seo, Geoffrey Irving, J. P. Lewis, and Junyong Noh. Compression and Direct Manipulation of Complex Blendshape Models. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA ’11, pages 164:1–164:10, New York, NY, USA, 2011. ACM.
- [SJS03a] Javid Sadr, Izzat Jarudi, and Pawan Sinha. The role of eyebrows in face recognition. *Perception*, 32(3):285–293, 2003.

- [SJS03b] Javid Sadr, Izzat Jarudi, and Pawan Sinha. The role of eyebrows in face recognition. *Perception*, 32(3):285–293, 2003.
- [SLC⁺19] Zhiwen Shao, Zhilei Liu, Jianfei Cai, Yunsheng Wu, and Lizhuang Ma. Facial action unit detection using attention and relation learning. *IEEE Transactions on Affective Computing*, 2019.
- [SLS⁺12] Yeongho Seol, JP Lewis, Jaewoo Seo, Byungkuk Choi, Ken Anjyo, and Junyong Noh. Spacetime expression cloning for blendshapes. *ACM Transactions on Graphics (TOG)*, 31(2):14:1–14:12, April 2012.
- [SMB15] Matthias Schröder, Jonathan Maycock, and Mario Botsch. Reduced marker layouts for optical motion capture of hands. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, pages 7–16, 2015.
- [SML16] Yeongho Seol, Wan-Chun Ma, and J. P. Lewis. Creating an actor-specific facial rig from performance capture. In *Proceedings of the 2016 Symposium on Digital Production, DigiPro '16*, pages 13–17, 2016.
- [SOC16] José Serra, Verónica Orvalho, and Darren Cosker. Behavioural facial animation using motion graphs and mind maps. In *Proceedings of the 9th International Conference on Motion in Games*, pages 161–166, 2016.
- [SP04] Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on Graphics (TOG)*, 23(3):399–405, 2004.
- [SSK⁺11] Yeongho Seol, Jaewoo Seo, Paul Hyunjin Kim, J. P. Lewis, and Junyong

- Noh. Artist friendly facial animation retargeting. In *ACM Transactions on Graphics (TOG)*, volume 30, page 162. ACM, 2011.
- [SSKS17] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.
- [ST11] Azim F Shariff and Jessica L Tracy. What are emotion expressions for? *Current Directions in Psychological Science*, 20(6):395–399, 2011.
- [Sta15] Susan Standring. *Gray’s anatomy e-book: the anatomical basis of clinical practice*. Elsevier Health Sciences, 2015.
- [SWH⁺17] Shunsuke Saito, Lingyu Wei, Liwen Hu, Koki Nagano, and Hao Li. Photorealistic facial texture inference using deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5144–5153, 2017.
- [SYCW17] Canping Su, Yan Yan, Si Chen, and Hanzi Wang. An efficient deep neural networks training framework for robust face recognition. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3800–3804. IEEE, 2017.
- [TBVM⁺14] Chloe Thompson-Booth, Essi Viding, Linda C Mayes, Helena JV Rutherford, Sara Hodsoll, and Eamon J McCrory. Here’s looking at you, kid: Attention to infant emotional faces in mothers and non-mothers. *Developmental science*, 17(1):35–46, 2014.
- [TDITM11] J Rafael Tena, Fernando De la Torre, and Iain Matthews. Interactive

- region-based linear 3d face models. In *ACM SIGGRAPH 2011 papers*, pages 1–10. 2011.
- [TG11] James W Tanaka and Iris Gordon. Features, configuration, and holistic face processing. *The Oxford handbook of face perception*, pages 177–194, 2011.
- [TGRJ19] Marie-Hélène Tessier, Chloé Gingras, Nicolas Robitaille, and Philip L Jackson. Toward dynamic pain expressions in avatars: perceived realism and pain level of different action unit orders. *Computers in Human Behavior*, 96:95–109, 2019.
- [TKC01] Y-I Tian, Takeo Kanade, and Jeffrey F Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 23(2):97–115, 2001.
- [TKY⁺17] 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):1–11, 2017.
- [TL08] Doris Y Tsao and Margaret S Livingstone. Mechanisms of face perception. *Annu. Rev. Neurosci.*, 31:411–437, 2008.
- [TPBF87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 205–214, 1987.
- [TR11] Jessica L Tracy and Daniel Randles. Four models of basic emotions: a

review of ekman and cordaro, izard, levenson, and panksepp and watt. *Emotion Review*, 3(4):397–405, 2011.

- [TZN⁺15] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. Real-time expression transfer for facial reenactment. *ACM Transactions on Graphics (TOG)*, 34(6), 2015.
- [TZS⁺16] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016.
- [TZS⁺18] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Facevr: Real-time gaze-aware facial reenactment in virtual reality. *ACM Transactions on Graphics (TOG)*, 37(2):1–15, 2018.
- [unr18] Epic games and 3lateral introduce digital andy serkis. <https://www.unrealengine.com/en-US/blog/epic-games-and-3lateral-introduce-digital-andy-serkis>, Mar 2018.
- [vdWSN⁺14] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillard, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.
- [VPH⁺11] Alessandro Vinciarelli, Maja Pantic, Dirk Heylen, Catherine Pelachaud, Isabella Poggi, Francesca D’Errico, and Marc Schroeder. Bridging the

gap between social animal and unsocial machine: A survey of social signal processing. *IEEE Transactions on Affective Computing*, 3(1):69–87, 2011.

- [WBCB08] Christian Wallraven, Martin Breidt, Douglas W Cunningham, and Heinrich H Bülthoff. Evaluating the perceptual realism of animated facial expressions. *ACM Transactions on Applied Perception (TAP)*, 4(4):1–20, 2008.
- [WBLP11] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. *ACM Transactions on Graphics*, 30(4):77:1–77:10, 2011.
- [WBSS04] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.
- [WC11] Christian Wallraven and Douglas W Cunningham. *Experimental design: From user studies to psychophysics*. AK Peters/CRC Press, 2011.
- [WJZ13] Nkenge Wheatland, Sophie Jörg, and Victor Zordan. Automatic hand-over animation using principle component analysis. In *Proceedings of motion on games*, pages 197–202. 2013.
- [WT03] Pamela M Walker and James W Tanaka. An encoding advantage for own-race versus other-race faces. *Perception*, 32(9):1117–1125, 2003.
- [WVK⁺17] Martin Wegrzyn, Maria Vogt, Berna Kireclioglu, Julia Schneider, and

- Johanna Kissler. Mapping the emotional face. how individual face parts contribute to successful emotion recognition. *PloS one*, 12(5), 2017.
- [WZLQ16] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.
- [XGS⁺18] Tian Xu, Oliver Garrod, Steven H Scholte, Robin Ince, and Philippe G Schyns. Using psychophysical methods to understand mechanisms of face identification in a deep neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1976–1984, 2018.
- [Yee04] Hector Yee. Perceptual metric for production testing. *Journal of Graphics Tools*, 9(4):33–40, 2004.
- [YGS12] Hui Yu, Oliver GB Garrod, and Philippe G Schyns. Perception-driven facial expression synthesis. *Computers & Graphics*, 36(3):152–162, 2012.
- [YSN⁺18] Shugo Yamaguchi, Shunsuke Saito, Koki Nagano, Yajie Zhao, Weikai Chen, Kyle Olszewski, Shigeo Morishima, and Hao Li. High-fidelity facial reflectance and geometry inference from an unconstrained image. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.
- [YWZ13] Mengzhao Yang, Kuanquan Wang, and Lei Zhang. Realistic real-time facial expressions animation via 3d morphing target. *Journal of Software*, 8(2):418–425, 2013.
- [ZM19] Katja Zibrek and Rachel McDonnell. Social presence and place illusion

are affected by photorealism in embodied vr. In *Motion, Interaction and Games*, pages 1–7. 2019.

[ZSCS08] Li Zhang, Noah Snavely, Brian Curless, and Steven M Seitz. Spacetime faces: High-resolution capture for modeling and animation. In *Data-Driven 3D Facial Animation*, pages 248–276. Springer, 2008.

[ZTG⁺18] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. In *Computer Graphics Forum*, volume 37, pages 523–550. Wiley Online Library, 2018.