

Dissertation

# Anatomically-Constrained Physics-Based Simulations For Facial Animations



Nicolas Wagner

2025



# ANATOMICALLY-CONSTRAINED PHYSICS-BASED SIMULATIONS FOR FACIAL ANIMATIONS

A dissertation to obtain the degree of  
DOCTOR OF NATURAL SCIENCES (DR.RER.NAT.)

presented by  
M. SC. NICOLAS WAGNER  
born on 14 September 1993 in Trier

submitted to  
DOCTORAL CENTER APPLIED INFORMATICS  
RHEINMAIN UNIVERSITY OF APPLIED SCIENCES

## Supervisors

PROF. DR. ULRICH SCHWANECKE  
RheinMain University of Applied Sciences  
PROF. DR. MARIO BOTSCH  
TU Dortmund University

## Examiner

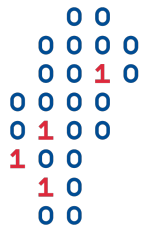
PROF. DR. RALF DÖRNER  
RheinMain University of Applied Sciences  
PROF. DR. MICHAEL WAND  
Johannes Gutenberg-University Mainz

Submission  
30.05.2025

Published  
WIESBADEN  
2025

Disputation  
01.09.2025

Except for the thesis publications, this work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International license and the copyright is held by the author. Copyright and licenses of the thesis publications are stated in Section 7.2.

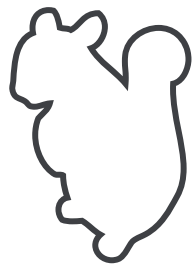


**Promotionszentrum**  
**Angewandte Informatik**

HAW Hessen



Hochschule **RheinMain**



**CVMR**

Computer Vision & Mixed Reality Group



## DECLARATION OF AUTHORSHIP

---

I, Nicolas Wagner, hereby declare that this thesis, entitled “Anatomically-Constrained Physics-Based Simulations for Facial Animations”, is my own work, and all the resources and materials have been properly acknowledged. I affirm that:

1. This thesis was prepared independently and without unauthorized outside help and with no other aids than *grammarly.com* for proof-reading.
2. All text passages taken literally or in spirit from published writings and all information based on verbal information are identified as such.
3. The principles of good scientific practice have been complied with.
4. This thesis has not been submitted for any other degree or qualification at any other institution.
5. This thesis only includes published articles first-authored by me in the original layout.

I understand that any act of plagiarism, fabrication, falsification, or other forms of academic misconduct in connection with this thesis may lead to the rejection of my work, the revocation of my degree, and other appropriate disciplinary actions.

Wiesbaden, 22.09.2025  
Nicolas Wagner





## ABSTRACT

---

This cumulative thesis presents novel advancements in the field of facial animation through the integration of anatomically-constrained physics-based simulations at various stages of the animation workflow. The motivation for this work stemmed from limitations of the currently most widely used animation technique *linear blendshapes* [56]. Although this approach’s computational efficiency and intuitive design are appealing, blendshape animations usually lack anatomical precision and can only partially reproduce nonlinear face characteristics. Among other things, they do not guarantee volume preservation, allow self-collisions, and are not able to incorporate external influences such as gravity or wind. Since physics-based simulations can mitigate these shortcomings, albeit in a slow and intricate manner, our principle goal was to combine the advantages of both concepts while avoiding their respective disadvantages.

Our work encompasses four publications, each addressing distinct animation components and achieving this goal in diverse ways. *SoftDECA* [107] integrates physics-based anatomical corrections into *linear blendshapes*, maintaining their efficiency even on consumer hardware. *SparseSoftDECA* [111] extends *SoftDECA* but creates realistic facial animations from sparsely tracked facial landmarks. *AnaConDaR* [110] provides solutions for facial retargeting, particularly an anatomical *deformation transfer* [105] to create more authentic and lifelike blendshapes. Finally, *NePHIM* [112] investigates real-time simulations of head-hand interactions, which are indispensable for conveying non-verbal communication cues.

All of our contributions are presented and discussed in light of associated research questions and analyzed with regard to conceivable limitations. We also propose potential advancements in our approaches and assess future developments of facial animations in general. This thesis aims to offer readers a concise yet well-substantiated overview of our research results.



## ZUSAMMENFASSUNG

---

Diese kumulative Dissertation integriert Physiksimulationen an verschiedenen Stellen des Workflows zur Animation von Gesichtern. Die Motivation für unsere Arbeit ergab sich aus den Einschränkungen der aktuell am weitesten verbreiteten Animationstechnik *linear blendshapes* [56]. Obwohl die effiziente Laufzeit und das intuitive Design dieses Ansatzes ansprechend sind, mangelt es Blendshape-Animationen in der Regel an anatomischer Präzision und nichtlinearen Eigenschaften. Unter anderem können sie keine Volumenerhaltung garantieren, erlauben Selbstkollisionen und sind nicht in der Lage, externe Einflüsse wie Schwerkraft oder Wind zu berücksichtigen. Physiksimulationen können diese Probleme abmildern, sind jedoch langsam und komplex. Daher war unser Ziel, die Vorteile beider Konzepte zu kombinieren und ihre jeweiligen Nachteile zu vermeiden.

Unsere Arbeit umfasst vier Veröffentlichungen, die sich jeweils mit unterschiedlichen Aspekten befassen und dieses Ziel in vielerlei Hinsicht erreichen. *SoftDECA* [107] fügt simulierte anatomische Korrekturen zu *linear blendshapes* hinzu, wobei deren Effizienz auch auf günstiger Hardware erhalten bleibt. *SparseSoftDECA* [111] wendet ein ähnliches Konzept an, um realistische Gesichtsanimationen aus nur wenigen markanten Gesichtsländern zu erstellen. *AnaConDaR* [110] bietet Lösungen für das Übertragen von Gesichtsausdrücken an. Insbesondere einen volumetrischen *deformation transfer* [105], um authentischere Blendshapes zu erzeugen. Schließlich wird in *NePHIM* [112] eine Echtzeitsimulation von Kopf-Hand Interaktionen vorgestellt, die für die Übermittlung nonverbaler Kommunikationshinweise unerlässlich ist.

Alle unsere Publikationen stellen wir anhand von verbundenen Forschungsfragen vor, diskutieren zugehörigen Resultate und zeigen denkbare Grenzen auf. Außerdem ergründen wir mögliche Weiterentwicklungen und geben einen denkbaren Ausblick auf die Zukunft von Gesichtsanimationen. Insgesamt soll diese Arbeit dem Leser einen kurzen, aber fundierten Überblick über unsere Forschungsergebnisse geben.



## ACKNOWLEDGEMENTS

---

First of all, I would like to thank both of my supervisors, Prof. Dr. Ulrich Schwanecke and Prof. Dr. Mario Botsch, who founded inspiring working environments at RheinMain University of Applied Sciences (Wiesbaden) and TU Dortmund University. Without their trust, patience, and support, this work would not have been possible.

I am deeply grateful to family and friends for always being there for me in a supportive and benevolent way. I especially want to express my gratitude to my wife, who has supported me through all the ups and downs during the time this thesis has been written.

I am very grateful to Prof. Dr. Ralf Dörner and Prof. Dr. Michael Wand, who kindly agreed to be the examiner of this thesis.

I would like to thank my colleagues for the fruitful daily discussions, for their support, and especially for their ideas and contributions, which were indispensable for this thesis.



## CONTENTS

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Research Questions . . . . .	4
1.2.1	Simulation of Blendshapes . . . . .	4
1.2.2	Simulation of Sparse Landmarks . . . . .	5
1.2.3	Simulated Facial Retargeting . . . . .	6
1.2.4	Simulation of External Interactions . . . . .	8
1.3	Summary & Organization . . . . .	9
<b>2</b>	<b>Related Work</b>	<b>11</b>
2.1	Facial Animations . . . . .	11
2.1.1	Personalized Blendshape Rigs . . . . .	11
2.1.2	Generalized 3D Morphable Models . . . . .	13
2.1.3	Implicit & Hybrid Avatars . . . . .	14
2.1.4	Facial Retargeting . . . . .	15
2.2	Physics-Based Simulations . . . . .	16
2.2.1	Simulation of Deformable Objects . . . . .	16
2.2.2	Simulation of Heads . . . . .	18
<b>3</b>	<b>SoftDECA – Simulation of Blendshapes</b>	<b>23</b>
3.1	Method Summary . . . . .	23
3.2	Discussion . . . . .	25
3.3	Publication . . . . .	28
<b>4</b>	<b>SparseSoftDECA – Simulation of Sparse Landmarks</b>	<b>41</b>
4.1	Method Summary . . . . .	41
4.2	Discussion . . . . .	42

4.3	Publication . . . . .	44
<b>5</b>	<b>AnaConDaR – Simulated Facial Retargeting</b>	<b>57</b>
5.1	Method Summary . . . . .	57
5.2	Discussion . . . . .	58
5.3	Publication . . . . .	61
<b>6</b>	<b>NePHIM – Simulation of External Interactions</b>	<b>77</b>
6.1	Method Summary . . . . .	77
6.2	Discussion . . . . .	78
6.3	Publication . . . . .	81
<b>7</b>	<b>Conclusion</b>	<b>97</b>
7.1	Summary . . . . .	97
7.2	Outlook & Impact . . . . .	99
	<b>Overview of Publications</b>	<b>103</b>
	<b>Bibliography</b>	<b>107</b>



## INTRODUCTION

---

### 1.1 MOTIVATION

The technological world was fundamentally different when the work for the contributions of this thesis started in October 2021. For instance, *Chat-GPT* had not yet been released [85], *Meta* was still called *Facebook* [128], and modern volumetric photorealistic rendering methods such as neural radiance fields did not yet exist [51, 74]. However, a noticeable trend towards more mixed or virtual reality in which authentic digital twins of real people can meet and communicate was already underway at this time. As can be seen in Figure 1.1, attention to research activities in related fields has increased almost exponentially since then, and the end is not yet in sight. Unfortunately, unlike two-dimensional video telephony, there are still no real-time capable three-dimensional equivalents that can depict people in mixed realities with the same simplicity and cost-effectiveness. For this reason, the current focus of research on virtual communication is on animating personalized avatars as realistically as possible from sparsely tracked motion information. Facial animations are particularly important here as they can easily pave the way into the Uncanny Valley [76], where animated avatars are often perceived as fake and, at times, even terrifying. The challenge of animating something as visually expressive as faces authentically is generally at odds with the computational efficiency required for virtual realities. Especially if access to realistic digital twins for a large audience is intended, they must be scalable and executable on inexpensive hardware.

#### Principle Objective

Therefore, the principle objective of this thesis is to develop an efficient, accessible, yet authentic framework for animating faces.

To that end, our general approach is to improve the realism of what is currently arguably the most widely used, easiest to control, and fastest

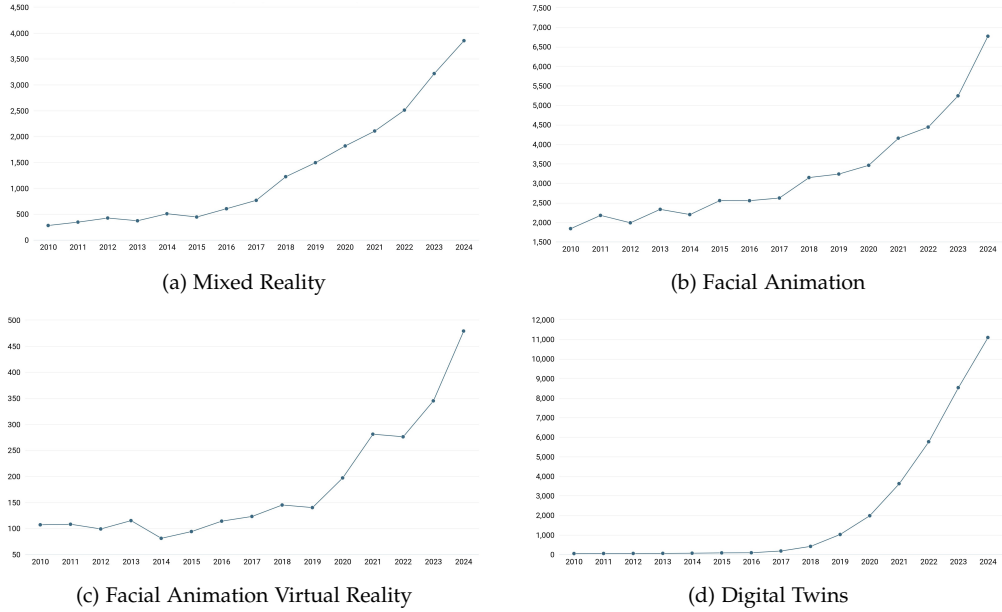


FIGURE 1.1: The total number of citations per year (2010–2024) for publications with certain keywords (subcaptions) in their title as per [28]. A sharp increase of interest in topics relevant to this thesis as of 2020 can be recognized.

method for facial animations: so-called *linear blendshapes* (*LBS*) [56]. *LBS* only requires a few exemplary expressions of a person, i.e., the blendshapes, and then animates the corresponding face by weighted linear interpolation of these examples. The ecosystem around *LBS* is vast and encompasses, for instance, automated algorithmic creation of blendshapes [58], tracking with commodity smartphones [4], as well as support from the most common game and animation engines like *Unity* [106] or *Blender* [10]. As all computer architectures nowadays efficiently implement linear interpolation, even weak hardware can perform *LBS* in just a few milliseconds.

In essence, *LBS* is a reasonable starting point, but it exhibits significant weaknesses concerning our objective. Above all, it usually lacks anatomical and physical plausibility, for which nonlinear animation techniques are better suited. Figure 1.2 depicts a few illustrative problems that arise from the simplistic linear concept of *LBS*. Among others, *LBS* does not guarantee biologically prescribed volume preservation, can not resolve lip

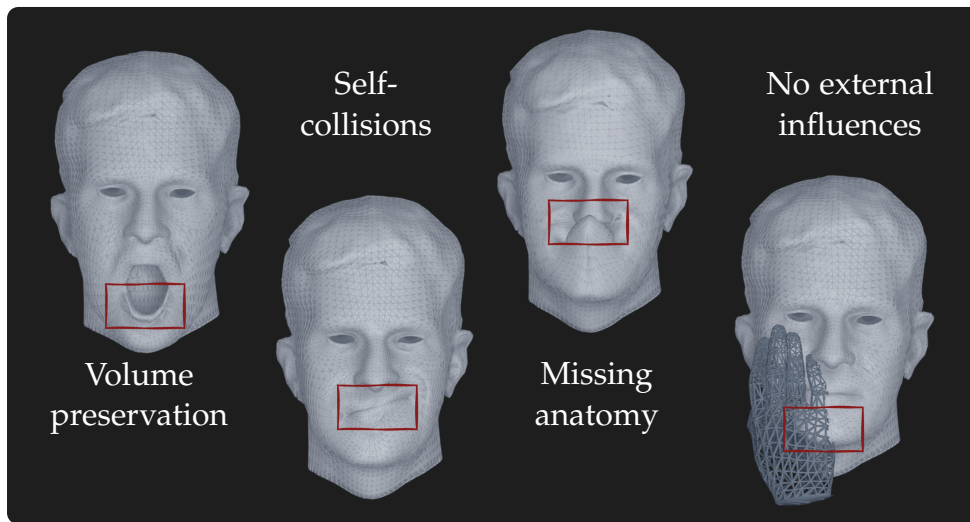


FIGURE 1.2: Some examples of *linear blendshapes* [56] resulting in implausible animations.

collisions, fails to respect anatomical movement restrictions, and ignores external influences like head–hand interactions.

Physics–based simulations (PBSs) provide a different way to animate faces, virtually solving all the aforementioned *LBS* problems. The basic idea of PBSs is to computationally simulate head anatomy to generate facial expressions while including forces from external effects. Figure 1.3 illustrates the representation of the anatomy we use in our works. The primary component is a tetrahedral mesh that comprises soft tissue, muscle tissue, and the skull. Although the precision of the simulation or the level of detail of the anatomical representation influence the quality of the animation, PBSs, in general, enhance plausibility by design. Just as generally, PBSs are not even remotely real–time capable, even on high–end hardware.

#### Principle Idea

Therefore, the principle idea of this thesis is to design a real–time capable and generally applicable framework that leverages PBSs to enhance the anatomical plausibility of *LBS*.

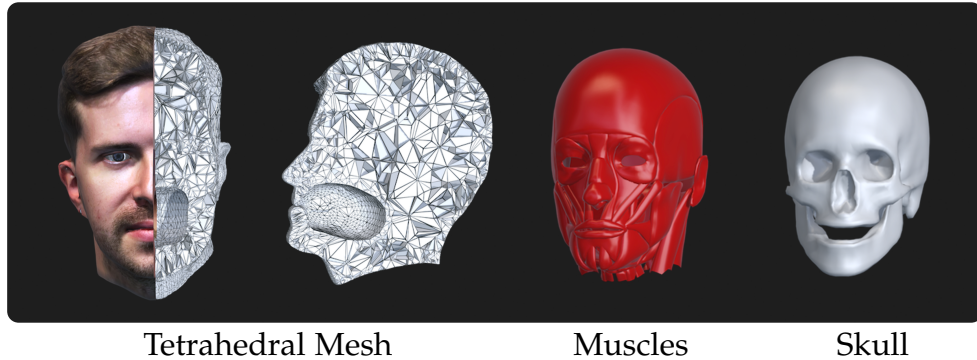


FIGURE 1.3: For the most part, the anatomy in our work is captured as a tetrahedral mesh, where we know for each tetrahedron whether it represents soft tissue, muscle tissue, or the skull. Depending on the application, muscle or skull surfaces are additionally represented as triangular meshes.

Put differently, we strive to integrate the strengths of both *LBS* (effectiveness, accessibility) and PBSs (realism) to achieve the principle objective.

## 1.2 RESEARCH QUESTIONS

Our concept of a framework that implements the principle idea evolves around four pivotal research questions described in the following.

### 1.2.1 SIMULATION OF BLENDSHAPES

The first question is the most elementary and impactful. The inspiration comes from the observation that many proposed head PBSs [6, 45, 44, 7] also attempt to provide the same interface as *LBS* in order to build on the same ecosystem. Roughly summarized, while such PBS models continue to utilize variants of blendshape interpolation, they also incorporate an additional *corrective* simulation phase. Although such corrective procedures can significantly bolster anatomical plausibility, they are likewise not real-time capable [44] or their contributions are tailored to only specific shortcomings of *LBS* [6]. The research question we derive is:

### Research Question 1

Can we accelerate any physics-based correction of *linear blendshapes* to facilitate its usage in real-time applications on consumer-grade hardware?

Successfully addressing this research question would largely implement our principle idea as, by construction, the advantages of the *LBS* are retained, but the disadvantages are compensated for.

#### 1.2.2 SIMULATION OF SPARSE LANDMARKS

Even with a successful answer to Research Question 1, structural challenges persist that limit realism. Chiefly among these issues are oftentimes undersized blendshape systems, which lack both expressiveness and detailed modeling. To illustrate, leading production blendshape systems from companies like *Apple* [4] or *Google* [126] include merely 52 expressions established on low-resolution topology. Even with improved resolution, the majority of people can not take advantage of detailed blendshapes because creating personalized ones commonly requires sophisticated and costly 3D multi-view scanners [25]. Methods for algorithmic personalization of blendshapes [59, 58, 73] are still far from being a substitute. Mainly, as it is difficult to reflect the complex material properties of heads. Fortunately, researchers have achieved substantial progress in the domain of facial landmark tracking in recent years. Please refer to Figure 1.4 for a visual demonstration. Although the depicted landmarks also do not thoroughly portray facial expressions, they furnish much more information than lower-dimensional blendshape rigs and capture the most essential facial contours. What is particularly attractive is that there are real-time capable and publicly available tracking methods [126] which achieve efficiency and accessibility similar to that of *LBS*. However, a method for realistically transforming the sparsely tracked landmarks into a dense facial animation is not yet available. This observation leads us to the second research question:



FIGURE 1.4: Two examples of precise facial landmark tracking of rather extreme facial expressions with *Mediapipe* [126].

#### Research Question 2

Can we develop a real-time, physics-based facial animation technique that forms dense, anatomically plausible facial expressions from sparse facial landmarks?

A successful answer to this research question could not only enhance realism in efficient facial animations but also potentially render the intricate process of (manually or algorithmically) assembling blendshapes obsolete in many cases. Thus, with regard to our principle objective, this research question also affects accessibility.

#### 1.2.3 SIMULATED FACIAL RETARGETING

In contrast to facial landmarks, blendshape systems typically organize facial animations using reasoned semantic principles that allow for an intuitive animation design. Therefore, they will maintain their status as a preferred method in production for a presumably long time to come. Consequently, enhancing the realism of our framework can not only focus on the simulation of blendshapes or landmarks, but must also focus on the effortless creation of more authentic blendshapes.

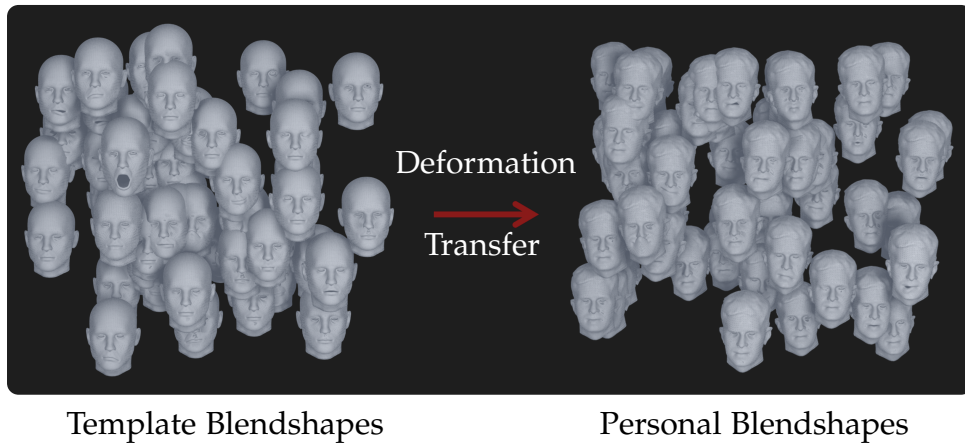


FIGURE 1.5: Examples of *deformation transfer* [13, 105] which maps deformations of a set of template blendshapes to a different character.

Latest smartphone-based methods [116] are at least able to create high-resolution neutral 3D avatars in minutes with low effort. Starting from the associated neutral facial expression, blendshapes can subsequently be generated algorithmically. The established method to that end is *deformation transfer* (*DT*) [13, 105], which personalizes manually sculpted template blendshapes. Expressed in rough terms, *DT* maps geometric deformations between the neutral template surface and a template blendshape to the neutral surface of the targeted person. Figure 1.5 visualizes examples of such personalizations. However, *DT* entirely overlooks anatomical restrictions, volumetric deformations, and visual perception. Therefore, we formulate the following preliminary research question:

#### Research Question

Can we develop a physics-based simulation that implements a volumetric and anatomically consistent *deformation transfer* while preserving the perception of expression characteristics?

More generally, the task of mapping expressions from a source to a target character is known as facial retargeting. In our basic scenario, we rely solely on the neutral head of the target without any additional information. A more complex challenge arises when there are *either* no example

expressions *or* a few available for the target. The gold standard for this challenge has long been *example-based facial rigging* (*EBFR*) [58], which effectively combines *DT* and *LBS*. As we already identified the potential for improving both algorithms through PBSs, the obvious, more general, third research question we formulate is:

#### Research Question 3

Can we develop a physics-based simulation that implements a volumetric and anatomically consistent facial retargeting while preserving the perception of expression characteristics?

A favorable answer to this question could bring considerable benefits to the realism of our framework without impairing efficiency or access. Not only could we then correct the plausibility of *LBS* animations (Research Question 1), but we could additionally sculpt the underlying blendshapes more lifelike.

#### 1.2.4 SIMULATION OF EXTERNAL INTERACTIONS

One major feature making PBSs attractive for facial animations is the simulation of dynamic interactions with heads – a factor we have not yet considered. Although most interactions are likely negligible, one stands out as particularly important: humans interact with their own faces dozens of times per hour [93, 77, 54] through hand actions like touching, stroking, scratching, rubbing, pulling, tugging, squeezing, grooming, caressing, poking, and so forth. Assumably, everyone can confirm, based on personal experiences, that these interactions amplify feelings or expressions (Figure 1.6) to a considerable extent. From an algorithmic perspective, it is not only the actual simulation of head–hand interactions that is a computational challenge, but also the ability to detect contacts in the first place. For this reason, we formulate our final research question as:

#### Research Question 4

Can we accelerate physics-based simulation and detection of head–hand contacts to function in real-time on commodity hardware while ensuring their animation is visually compelling?





FIGURE 1.6: Two examples of how simulated head–hand interactions can amplify the expressiveness of facial expressions.

A successful answer to this question would add a component to facial animation that is essential for authenticity but has been neglected in research until lately [101].

### 1.3 SUMMARY & ORGANIZATION

By giving answers to the above four research questions, we develop a physics-based facial animation framework in this thesis, which achieves the principle objective through the following summarized contributions:

- Real-time physics-based corrections of *linear blendshapes* for physically and anatomically more plausible facial animations (Chapter 3).
- Real-time physics-based simulation of sparse landmarks to generate corresponding dense facial expressions (Chapter 4).
- Anatomically and physically more plausible algorithmic facial retargeting while respecting the perception of facial expressions (Chapter 5).

- Real-time simulated head-hand interactions within facial animations (Chapter 6).

Before delving into detailed discussions of the individual contributions in the upcoming chapters, we first review the current state of research in related fields (Chapter 2).

In each chapter associated with a specific contribution (Chapters 3 – 6), we start off by summarizing the affiliated publication before reflecting on its results in the context of the corresponding research question. Moreover, we evaluate our contributions by comparing them with related work and analyze relevant publications that have emerged since ours. Based on these assessments, we also derive promising directions for future work. All contribution chapters aim to offer concise summaries of our work that enable readers to contextualize research questions, methods, and results even without a full grasp of the underlying publication.

In the concluding Chapter 7, we reflect on our work in its entirety, discuss its potential impact, and contemplate future directions of facial animation. A list of all publications by the author of this thesis and an overview of contributions to the thesis publications by other authors follow this conclusion.

## RELATED WORK

---

We continue with a thorough review of the research fields in the context of which the contributions of this thesis have been published. The review comprises seminal works as well as recent developments up to the current state-of-the-art. At first, we address methods for animating faces (Section 2.1) and subsequently outline physics-based simulations (PBSs) in general and with regard to facial animations (Section 2.2). The differentiation of our contributions in the light of related work happens in the corresponding discussions of Chapters 3 – 6.

### 2.1 FACIAL ANIMATIONS

The history of facial animation in digital worlds is as long as it is diverse. We, therefore, recommend the survey of Parke and Waters [89] for a comprehensive overview of earlier animation techniques and confine ourselves to recent techniques for animating human-like characters. Our overview of facial animations first examines approaches akin to *linear blendshapes* (*LBS*) [56], which can attain high animation quality through individualized facial rigs (Section 2.1.1). Afterwards, we examine the somewhat opposing idea: statistical 3D morphable models (3DMMs). 3DMMs are meant to be universally applicable for many people and, at least today, thereby only partially allow for personalized nuances (Section 2.1.2). In Section 2.1.3, we inspect the latest trend, implicit and hybrid avatars, which are able to create photorealistic and individualized facial animations but typically still rely on *LBS* or 3DMMs. Finally, we have a more detailed look at the subproblem of facial retargeting (Section 2.1.4), i.e., transferring facial expressions from a source to a target character.

#### 2.1.1 PERSONALIZED BLENDSHAPE RIGS

The foundational idea of blendshape rigs is to capture a person or a fictional character through the geometry of exemplary facial expressions and

to animate them by combining these examples. The most basic and common variant is *LBS* [56], which creates animations by weighted linear interpolation. Accordingly, a user can steer the animation by manipulating the interpolation (blendshape) weights. To this day, researchers have developed a multitude of enhancements rooted in *LBS* to cope with the fact that faces are, unfortunately, highly nonlinear. We categorize influential developments in terms of four fundamental segments: the acquisition or modeling of blendshapes, the type of blendshapes, the combination operation, and the control of the animation.

#### BLENDSHAPE ACQUISITION

The industry standard [23] is, and has been for years, to capture a person in a sophisticated 3D scanner [25] and to subsequently transform the 3D scans into standardized blendshapes with a great deal of manual effort. While there are methods that produce blendshapes entirely without (or only a handful of) 3D scans [59, 84, 58, 13, 105], automatically convert 3D scans into blendshapes [11, 62], or map single-view depth images to blendshapes [73], these often lack a sufficient level of personal details.

#### BLENDSHAPE TYPES

Alongside blendshape systems that operate directly on the geometry of facial expressions, indirect approaches also exist. One is the *BlendForces* [7] idea, which interpolates accelerations instead of locations and integrates them over time in a PBS. Muscular blendshape rigs [5, 72] generate facial expressions by simulating interpolated muscle contractions. Both concepts intend to overcome the problems of *LBS* by mimicking nonlinear anatomical properties. For more details on head simulations, please see Section 2.2.2.

#### BLENDSHAPE COMBINATION

Another manner to compensate for the nonlinearity of faces is to replace the plain weighted linear interpolation of blendshapes. For example, with corrective blendshapes [46], which are added to the original linear interpolation if predefined blendshape combinations occur. The weight of a corrective blendshape is the product of the linear weights that are part of the associated combination, resulting in a nonlinear augmentation. The

authors of [61] adopt a similar approach by adding pose-specific corrective blendshapes [61] to the original linear interpolation in a nonlinear dependence on the pose of the jaw and the eyes. More recently, *patchwise LBS* [18] became a popular approach, which linearly interpolates consistently connected face patches with separate interpolation weights.

#### BLENDSHAPE CONTROL

A common concern of blendshape rigs is the question of which expressions to include as blendshapes and what semantic meaning the interpolation weights consequently carry. Smaller systems, such as the 52 *Apple ARKit* [4] blendshapes, attempt to isolate facial movements, while more complex frameworks as *Animatomy* [23] (178 blendshapes) imitate muscle contractions, for instance. Particularly in studio productions, this kind of interpretability is usually not a necessity. Instead, such productions utilize even more blendshapes to increase the expressiveness [94] ( $\sim 1000$  blendshapes), whereby, however, the applicability for rather inexperienced users dwindles.

#### 2.1.2 GENERALIZED 3D MORPHABLE MODELS

Put simply, 3DMMs center around a template 3D head that is deformable in a latent space with parameters for identity, expression, and others, depending on the respective model. To that end, a training process structures the latent space by registering the template to a comprehensive dataset of exemplary facial expressions. An animation can be created by manipulating the expression parameters. There are at least two main differences from blendshape systems. On the one hand, 3DMMs are designed to generalize and usually do not provide an accurate representation of personal details in comparison to elaborately created blendshapes. On the other hand, the training process often does not allow for an intuitive and semantically meaningful control of the latent space as blendshape rigs do. For an overview of the various facets of such models, please refer to the survey by Egger et al. [30].

The most influential 3DMM is undoubtedly *FLAME* [61], which is even a foundation for today’s state-of-the-art implicit facial animations [90] (Section 2.1.3). *FLAME* is a linear model primarily based on principle component analysis of facial 3D scans. A widespread extension of *FLAME*

is *DECA* [32], which additionally applies nonlinear facial deformations through a neural network trained on real 2D images of faces. Recently, the first physics-based 3DMM was introduced [125], which also respects anatomical constraints.

The hybrid model by Li et. al. [59] employs a neural network trained to automatically generate personalized blendshapes. Yet, it does not generalize over identities but requires a neutral 3D head scan of a person as input.

### 2.1.3 IMPLICIT & HYBRID AVATARS

Both of the aforementioned concepts, blendshape rigs and 3DMMs, aim to create high-quality facial animations primarily through explicit geometry. Over the last few years, however, there has virtually been a paradigm shift towards photorealistic animations on implicit density fields. Therefore, the already rapid developments in the area of facial animations accelerated once again.

The “implicit trend” began with the introduction of *neural radiance fields (NeRFs)* [74], which continuously represent volume density and view-dependent colors within a given scene through a neural network. Classical ray tracing techniques [48] can then render the neural volumetric representation into images of the scene. This generic idea achieves impressive photorealistic results and has been extended to dynamic scenes [88, 87], as well. The logical consequence of positioning heads within such dynamic scenes and conditioning them to animatable expression parameters also arrived quickly [33].

Due to the fundamental problem of *NeRFs* being inefficient and slow as they learn and process void space, many ideas for speeding them up have emerged [51, 80]. Probably the most important for facial animation at the moment is *Gaussian splatting* [51]. Here, 3D Gaussian splats capture the scene space, but only in areas with nonzero density. *GaussianAvatars* [90], for instance, implements this concept for facial animations by attaching Gaussian splats to the 3DMM *FLAME* [61]. As a result, the animation can easily be controlled in the associated expression space and a rough density distribution is predetermined, enabling fast training and inference. The animation of *GaussianAvatars* reaches a photorealism previously unattained. The same photorealistic extension does not only exist for 3DMMs but also

for *LBS* [70]. Nonetheless, both of the latter hybrid methods emphasize how vital the long-researched explicit techniques (Sections 2.1.1, 2.1.2) are even for the latest implicit photorealistic animations.

#### 2.1.4 FACIAL RETARGETING

Facial retargeting, as a subproblem of facial animation, can be further broken down. For example, whether it is to be done in real-time, whether it involves only human characters, or whether it is data-driven. For a full taxonomy and an overview of earlier work, we recommend referring to [127]. As throughout the thesis, here, we focus on humanoid characters.

In principle, the same concepts as for facial animation also apply to facial retargeting. To retarget using blendshapes, for instance, “solely” corresponding blendshape systems of the source and target characters are required, between which blendshape weights can then simply be transferred [18, 23]. With 3DMMs, retargeting is inherent, as the animation of different identities can be driven by the usually shared expression space [32, 61]. The hybrid animations described in Section 2.1.3 can therefore be retargeted straightforwardly, too.

However, similar problems arise as before for facial animation in general. Among other things, it is laborious to compose corresponding blendshape systems and define correspondences semantically. It is, therefore, not surprising that the latest developments rely on *patchwise LBS*, which can get by with a small number of blendshapes due to their nonlinearity [18]. The lack of personal details in 3DMMs and their restriction to the characteristics of the training data may also result in low retargeting quality or even prevent them from being applied at all. Above all, dissimilar mesh tessellations pose a key challenge. Consequently, the most advanced model in this context is a 3DMM that has been trained on a wide variety of data resources with varying tessellations, thus enabling independence from particular data characteristics [92].

Both blendshapes and 3DMM are fundamentally data-driven approaches. Yet, access to the necessary data is often limited, which prompted the development of heuristic retargeting approaches, as well. Among these, *deformation transfer (DT)* [13, 105] is particularly influential. *DT* captures a source expression through deformation gradients with respect to the neutral source face and retargets by applying the gradients to the neutral

target face. As  $DT$  does not inherently account for personal characteristics, several extensions [84, 122, 9] have been developed to incorporate personal details and enhance the authenticity of the retargeting.

## 2.2 PHYSICS-BASED SIMULATIONS

As with facial animations, the history of PBSs goes back to the beginnings of computer graphics. Researchers developed countless methodologies for the simulation of rigid bodies [8], fluids [114], and elastic objects [14, 78] over the course of time. Since this thesis centers around the simulation of heads, concepts for the simulation of deformable elastic objects are of particular interest. In the following section, we will accordingly provide a comprehensive review of the literature in this area, starting with heuristic foundations to modern data-driven approaches (Section 2.2.1). We then discuss how these approaches can be applied for simulating heads and outline simulations developed specifically for facial animations (Section 2.2.2).

### 2.2.1 SIMULATION OF DEFORMABLE OBJECTS

#### HEURISTIC SIMULATIONS

Highly simplified heuristic models mainly characterize the beginnings of PBSs of deformable elastic objects. Presumably, the most basic are mass-spring systems, where objects are volumetrically captured by point masses that influence each other via massless springs [19, 65]. Although these systems are intuitive and easy to use, they are not very physically accurate and only allow for simulating simple materials. For more complex ones, approaches derived from the continuum mechanics perspective have proven advantageous. Here, especially finite element methods (FEMs) are noteworthy [43, 82, 26, 17], which can be considered as a generalization of mass-spring systems. FEMs model objects as continuously connected volumes and discretize them as irregular meshes, e.g., tetrahedral meshes. The actual simulation involves solving a partial differential equation derived from Newton’s second law that, in simplified terms, relates internal (material) and external (interaction) forces to accelerations via masses. The accelerations can then be converted into deformations of the underlying mesh by adopting a time integration scheme.



As it is often considerably more intuitive to interact with vertex positions of a mesh than with accelerations, the *position-based dynamics* (*PBD*) paradigm was proposed [78]. This paradigm employs explicit Euler integration to incorporate external forces and directly adjusts vertex positions through a gradient-based optimization scheme to account for internal forces. Despite its high popularity, the explicit integration scheme, the dependence of the gradient method on various hyperparameters, and the sequential consideration of internal forces turned out to be challenging. As a result, along with improvements in *PBD* [71], another paradigm emerged: *projective dynamics* (*PD*) [14]. It uses a more robust implicit integration scheme and operates much faster by considering all forces simultaneously. The latter property mainly stems from a restrictive form of internal forces and their associated energy potentials. Subspace methods, which run *PD* on low-dimensional latent representations [15], or other optimization accelerators, such as multigrid methods [121], can further increase the efficiency of *PD* in general.

Both *PD* and *PBD* share the shortcoming that they cannot, at least canonically, make use of modern, realistic collision handling [60]. While more straightforward contact mechanisms can be integrated into *PD* with only little computational overhead [113, 68], state-of-the-art methods result in significantly increased runtimes [57].

It is safe to say that the still ongoing developments, extensions, and improvements [22, 21, 79, 121, 66] of *PD* as well as *PBD* underline their fundamental importance for the simulation of deformable objects even nowadays. We recommend the very recent survey by Holz et al. [40] for a more in-depth discussion of both paradigms.

## MACHINE LEARNING FOR SIMULATIONS

Common to all the heuristic methods mentioned above is that material models and associated parameters require manual specifications. Modeling complex deformable objects such as heads in this manner is challenging and sometimes infeasible. For this reason, methods based on machine learning evolved that learn to approximate physical simulations from exemplary data [64, 99, 39]. However, such models are not guaranteed to obey physical laws and, therefore, usually do not generalize sufficiently [29]. This observation, in turn, led to hybrid methods which rely on the user to specify material models but are able to learn material parameters

[69, 91, 103, 29, 35, 41, 37, 42]. Due to their simplicity and efficiency, differentiable versions of *PBD* [104] and *PD* [63, 29] gained notable recognition, as well. For a more comprehensive overview of differentiable PBSs, predominantly used in soft robotics, please refer to the survey Newbury et al. [83].

### 2.2.2 SIMULATION OF HEADS

#### HEURISTIC SIMULATIONS

Alongside the more general concepts of simulating deformable objects, an extensive body of work conceptualizes head simulations. The purposes of such simulations are manifold and range from anatomical model identification [47] and expression tracking [5] to facial retargeting [110] and animation [44]. Generally, head simulations can be categorized as forward and inverse simulations. Forward simulations convert the actuation of muscles and/or the position of the skull into facial expressions. Inverse simulations identify the deformations of the head that can cause facial expressions.

The pioneering work of Sifakis et al. [102] was the first to use a volumetric FEM to capture and animate the anatomical behavior of a head. To this end, they represent soft tissue as a tetrahedral mesh, incorporate muscle fiber direction fields, model the jaw as well as the cranium as triangle surface meshes, and rely on customized forward and inverse simulation solvers. A later extension, which represents muscle contractions as manipulable B-spline trajectories, further enhances the quality and controllability of the forward simulation [72]. Unfortunately, their approach can only be applied to a limited extent in real-world applications, as manually modeling a person’s anatomy requires several days, and simulating a frame takes several minutes.

The *Phace* [45] model enriches Sifakis et al. [102] with an automated positioning of the anatomy in the form of a single tetrahedral mesh, advanced material models, especially for the muscles, and with much more efficient forward and inverse simulations that take only seconds instead of minutes. *Phace* is controllable like blendshapes via interpolation weights, although the forward simulation linearly interpolates and evaluates muscle contractions.

Besides the aforementioned more intricate anatomical simulations [45, 102], methods that can be summarized as volumetric blendshapes exist

[53, 44]. These methods only consider a general volumetric tissue and the skull, but refrain from using a sophisticated material model of the muscles. As opposed to *LBS* (Section 2.1.1), where the geometry of facial expressions is interpolated directly, here, volumetric deformations are interpolated and subsequently simulated. While volumetric blendshapes are admittedly faster due to their simpler structure and the help of efficient solvers such as *PD* [44], we found in our experiments that they are still not real-time capable out-of-the-box even on modern high-end hardware. In principle, more advanced simulation solvers enable real-time simulations [6] of volumetric blendshapes, yet only for severely limited anatomical precision.

Simulations based on thin shells in lieu of volumetric representations have also been developed [52, 7]. Although such simulations tend to be more efficient due to the reduced complexity of the underlying model, by construction, they can not adequately simulate many important anatomical characteristics (e.g., preservation of tissue volume).

#### HAND-HEAD INTERACTIONS

A contribution of this thesis is the simulation of hand-head interactions, which the approaches presented above neglected. This fact is somewhat remarkable as PBSs allow for such interactions, unlike most other approaches to facial animation (Section 2.1). To the best of our knowledge, there is only a single work addressing this topic called *Decaf* [101]. However, the simulation of *Decaf* is simplistic and can solely handle basic interactions without long-term time dependencies or dragging movements. Although there are more general techniques for imitating complex hand-object interactions [97, 98], there are none for efficient hand-head interactions.

#### ANATOMICAL MODELS

As already indicated before, every volumetric head simulation first faces the difficulty of registering anatomy into a head. Regardless of the actual representation forms (e.g., tetrahedral meshes, triangle meshes, fiber fields, etc.), the main concern is the positioning of the soft tissue, the muscles, and the skull. In the most straightforward but at the same time rarest case, MRI and CT scans of a targeted head exist. Then, one can either manually [102] or automatically [47] reconstruct the anatomy with precise

accuracy. In most cases, however, only the neutral surface of the head is known, requiring a heuristic or data-driven estimation of the interior. For instance, *Phace* [45] relies on the heuristic *Anatomy Transfer* [2], which warps an anatomical template into any head based on a predefined fat distribution. Due to the limited availability of appropriate data sources, data-driven methods are predominantly focused on the positioning of the skull and still require a heuristic positioning of the musculature [24, 75, 1]. Recently, Keller et al. introduced the *HIT* model [49], which can also place muscle tissue. Yet, *HIT* results in an implicit prediction that must be converted for explicit simulations.

#### MACHINE LEARNING FOR HEAD SIMULATIONS

Many current head simulations based on machine learning are closely related to volumetric blendshapes [44] regarding the conceptual idea. Both have in common that muscles are neither directly controlled nor modeled. Instead, the pathbreaking approach of Srinivasan et al. [103] employs a volumetric mesh representation for all tissue types and an explicit surface-based description of the skull. For training, they register the anatomy to numerous exemplary expressions of a person using an adapted differentiable *PD* solver [29]. During the training, the skull can only move rigidly, and the tissue can only deform to a regularized extent. Contrary to volumetric blendshapes, the deformations determined in this manner are not directly interpolated and simulated to generate facial animations. Instead, a neural network approximates them as a function of a latent space. The control of facial expressions can then be performed in the latent space, and the geometry of an expression is finally obtained by simulating the predicted deformations. This method was extended to implicit volumetric representations [123] while an application to several persons simultaneously was developed, too [124]. By leveraging fast simulation-free differentiable loss functions that capture anatomical properties, Yang et al. [125] can exploit considerably more training data at once, leading to a similar model as Srinivasan et al. [103] but with a generalization across head shapes.

Further concepts also apply machine learning techniques to head simulations with embedded muscle descriptors. For example, the previously mentioned B-spline trajectories [72] were redesigned to be differentiable [5], even enabling the inverse simulation to 2D RGB images of real facial expressions.

Park and colleagues [86] propose to use deep learning as an accelerator of head simulations: a head is animated in real-time by a low-resolution plus anatomically inaccurate PBS and mapped by a neural network (with likewise real-time capabilities) into a corresponding expression of a slow but high-resolution plus anatomically very accurate simulation.



## Citation

**SoftDECA: Computationally Efficient Physics-Based Facial Animations**

Nicolas Wagner, Mario Botsch, and Ulrich Schwanecke  
Proceedings of the 16th ACM SIGGRAPH Conference on Motion,  
Interaction, and Games, 2023  
DOI: [10.1145/3623264.3624439](https://doi.org/10.1145/3623264.3624439)

### 3.1 METHOD SUMMARY

In the publication *SoftDECA*, we responded to Research Question 1 (Section 1.2.1).

The core concept of *SoftDECA* is to train an efficient neural network that effectively corrects facial *linear blendshapes* (*LBS*) animations with respect to an anatomically constraining physics-based simulation (PBS). To that end, it adapts a deep hypernetwork [36] architecture that executes in milliseconds on consumer-grade CPUs. The training of this network relies on a training data generation pipeline that enables *SoftDECA* to generalize across diverse head shapes, facial expressions, and material properties. We outline the key components of our approach in more detail in the following paragraphs.

#### PHYSICS-BASED SIMULATION

Despite the fact that the network architecture we adopt does not impose structural constraints on the correcting PBS, we focus on a dynamic, volumetric, and inverse finite element simulation, which we solve with *projective dynamics* (*PD*) [14]. Given an *LBS*-generated facial expression, the simulation restores physical and anatomical properties. For instance, it

respects the volume preservation of soft and muscle tissues, the strain of skin and soft tissue, the rigidity of the skull, and resolves self-collisions.

The simulation builds on a novel representation of the anatomy, the so-called *layered head model (LHM)*, which encapsulates the skin and soft tissue, the muscles, and the skull within enveloping wraps. The wraps and, thus, the enclosed anatomical structures can be placed within a head in a data-driven manner. We also propose a simple mechanism to prevent layers from intersecting during the data-driven placement so that no anatomical inconsistencies can occur.

#### NETWORK

*SoftDECA*'s hypernetwork that approximates the simulation is split into two components. The first, a larger component, processes the neutral head shape and material properties. Its output defines the weights for a smaller component that runs the actual simulation. To remain compatible with standard *LBS*, blendshape weights still control the animation. As only the smaller component executes per frame, *SoftDECA* achieves rapid animation speeds.

#### TRAINING DATA

The pipeline for creating training data for the hypernetwork comprises multiple steps, starting with randomly selecting neutral head surfaces from the high-resolution *DECA* head model [32]. Next, we align the *LHM* with the sampled heads to enable our simulation. We further use *deformation transfer* [105] to automatically map the 52 *ARKit* blendshapes [4] to the head samples. Finally, we can create the actual training examples. As the training input, we animate all heads via the mapped blendshapes. The animations originate from blendshape weights recorded with a customized *iOS* app in ten dyadic conversations of real people. For this purpose, we mounted an iPhone in front of each participant's face. The corresponding ground truth results from applying our simulation to the *LBS* animations with random material parameters.



## 3.2 DISCUSSION

### RESULTS

*SoftDECA* provides an effective, comprehensive, and successful answer to the posed Research Question 1. The hypernetwork we trained learned to generalize over nearly all *DECA* model heads, even those not used for training. Deviations from the underlying corrective simulation are within the low submillimeter range, and visual results indicate that these deviations do not depend on the facial expression. Visual results also evidently demonstrate that eliminating self-collisions and incorporating other anatomical properties facilitate nonlinear improvements of *LBS* animations.

In addition, *SoftDECA* successfully accomplishes the intended generalization over material properties. This aspect allows for various manipulations, such as adjusting a person’s weight for therapeutic applications, simulating surgical interventions, reflecting conditions like paralysis, or the generation of realistic wrinkles. The latter can be taken to the extreme, and even “zombifications” can be conducted, underscoring *SoftDECA*’s proficiency in learning and replicating even high-frequency details. Besides internal properties, *SoftDECA* allows for incorporating and manipulating constant external influences. Such influences, in turn, can improve realism; for example, a selfie taken standing up looks very different from one taken lying down due to the direction of gravity. Furthermore, as *SoftDECA*’s simulation is dynamic, effects such as the wobbling of soft tissue can be animated.

Despite the comprehensive capabilities, *SoftDECA* executes in less than 10 milliseconds per frame, even on CPUs with limited processing power. This performance level is adequate for real-time applications and meets the frame rate requirements essential for virtual reality applications.

Ultimately, it is worth emphasizing that the foundational hypernetwork of *SoftDECA* follows the *ONNX* [27] standard so that the network can be readily loaded into common animation frameworks like *Unity* [106]. This feature significantly increases the usability of our work in addition to the controllability through blendshape weights.

The by-product of *SoftDECA*, the *LHM* that positions the head anatomy in a data-driven manner, also achieves high precision. The observed placement errors in areas crucial for facial animation are in the range of only a few millimeters. Only in rather unimportant parts, such as the neck,

errors can be as large as one centimeter.

#### LIMITATIONS

A partially unresolved aspect of Research Question 1 remains whether *SoftDECA* is feasible to accelerate all PBSs for facial animation. While we can not make a definitive claim, *SoftDECA* combines various PBS components of previous state-of-the-art head simulations [45, 72]. We can, therefore, at least assure high visual quality. Moreover, no theoretical limitations exist that would hinder the application of the *SoftDECA* training concept to other types of simulations.

Arguably, the most significant concerns of our work stem from the training data we utilized. We focused solely on *DECA* heads equipped with algorithmically generated blendshapes. Hence, the crucial questions arise: Can *SoftDECA* handle manually created or scanned blendshapes? Can *SoftDECA* process heads that fall outside the *DECA* distribution? A more detailed explanation of how the hypernetwork functions is necessary to address the first question. The hypernetwork does not directly predict facial expressions but relative geometric deformations as corrections to the underlying *LBS* animation. As a result, *SoftDECA* is unaffected by how the underlying blendshapes are sculpted. To investigate the second question, we evaluated *SoftDECA* on an external head dataset that comprises blendshapes personalized through scanned facial expressions [58]. Admittedly, this evaluation resulted in higher approximation errors, but the visual results remained convincing. In summary, *SoftDECA* demonstrated substantial robustness beyond its training data. However, it is crucial to note that no theoretical guarantees ensure generalization.

In any case, we created *SoftDECA* solely on the wide-spread semantic structure of the *ARKit* [4] blendshape system. To adopt a different structure, one must reproduce the training data and retrain the hypernetwork, a process that takes approximately five days on a high-end workstation. Likewise, changing the type or number of material parameters also necessitates the entire training procedure to be repeated.

Moreover, in contrast to the approximated PBS, the learned hypernetwork can not incorporate dynamically changing external influences, like head-hand interactions, as these are absent from the training process. Generating suitable training data that reflects such interactions is intricate. Nevertheless, we propose a solution to this challenge in Chapter 6.

Finally, it is noteworthy that the implementation of second-order effects in *SoftDECA* relies on a linearization, resulting in a simplified yet mostly adequate representation of dynamics.

#### RELATED WORK

Previous work related to *SoftDECA* primarily focuses on developing either highly realistic PBSs [45, 72] or fast ones with restricted authenticity [6, 7, 44]. However, accelerating realistic ones with machine learning remains largely unexplored. While there are some universal prior investigations into this idea [39, 15], to our knowledge, *SoftDECA* represents the first effort to accelerate complex PBS for facial animations with deep learning. Above all, we are unaware of any previous work that curates such an extensive training dataset as we do.

Most closely related to our approach is the *Generalized Physical Face Model* by Yang et al. [125], which appeared after *SoftDECA*. This model is trained on approximately 13000 high-resolution 3D scans of real facial expressions via a differentiable PBS. Due to its extensive data foundation, it achieves a higher level of realism and is as generally applicable as our approach. However, it is not designed for fast facial simulations and can not be used in real-time applications.

The *LHM* of *SoftDECA* builds on data from the Achenbach et al. [1] skull predictor, which does not prevent collisions between the skull and head, unlike ours. Other works related to the *LHM* emerging around or after *SoftDECA* primarily focus on the kinematic skeleton of the entire body [117, 24, 50], lacking precision for head bones. However, a notable recent work, the *HIT* model [49], accurately positions even muscles in the head. Unfortunately, *HIT* is defined implicitly, making it challenging to extract standardized surface meshes – an important premise for our *PD*-based anatomical simulation.

#### FUTURE WORK

The most obvious advancement of our method would be to train *SoftDECA* on an equally extensive collection of real 3D scans as Yang et al. [125]. Nonetheless, there are inevitable hurdles as their dataset remains unpublished, and we are unaware of any comparable publicly available dataset. Creating such an extensive set of scans requires not only costly multi-view 3D scanning technology but also a substantial amount of man-

ual labor by skilled digital artists to purify the scanned information. It is, therefore, understandable that almost only large corporations can afford to do so and keep the resulting data for themselves. In other words, future developments will benefit significantly from using more real data. To accomplish this effectively, however, we must first improve and speed up photogrammetry techniques. Another promising advancement of our approach would combine the implicit anatomical model *HIT* [49] with our robust *LHM*.

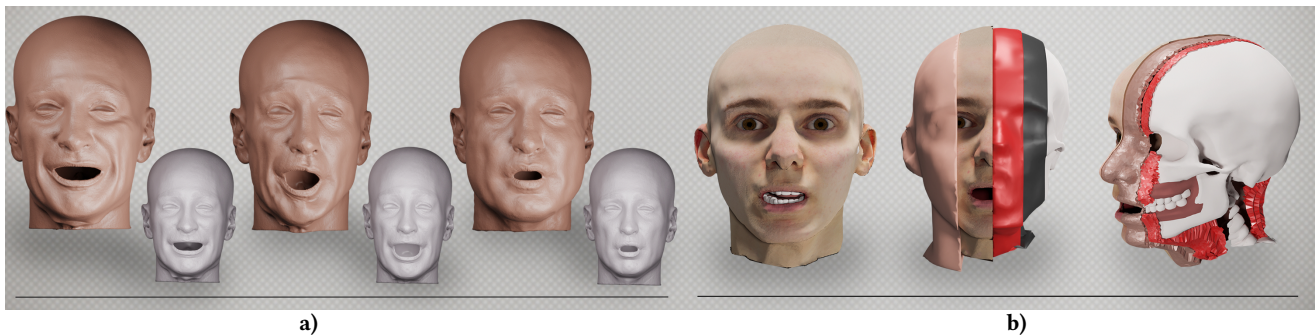
### 3.3 PUBLICATION

# SoftDECA: Computationally Efficient Physics-Based Facial Animations

Nicolas Wagner  
nicolas.wagner@tu-dortmund.de  
TU Dortmund University  
Dortmund, Germany

Ulrich Schwanecke  
ulrich.schwanecke@hs-rm.de  
RheinMain University of Applied  
Sciences  
Wiesbaden, Germany

Mario Botsch  
mario.botsch@tu-dortmund.de  
TU Dortmund University  
Dortmund, Germany



**Figure 1:** a) SoftDECA (brown) compared to linear blendshapes (gray): More realistic non-linear facial animations (left), biomechanical restrictions like Bell's Palsy (middle), and interactive manipulations like an increase in weight (right) are only a few examples that can be efficiently animated. b) The layered head model that encapsulates the skin, the muscles, and the skull with wraps that builds the foundation of SoftDECA and for which we present a data-driven fitting algorithm.

## ABSTRACT

Facial animation on computationally weak systems is still mostly dependent on linear blendshape models. However, these models suffer from typical artifacts such as loss of volume, self-collisions, or erroneous soft tissue elasticity. In addition, while extensive effort is required to personalize blendshapes, there are limited options to simulate or manipulate physical and anatomical properties once a model has been crafted. Finally, second-order dynamics can only be represented to a limited extent.

For decades, physics-based facial animation has been investigated as an alternative to linear blendshapes but is still cumbersome to deploy and results in high computational cost at runtime. We propose SoftDECA, an approach that provides the benefits of physics-based simulation while being as effortless and fast to use as linear blendshapes. SoftDECA is a novel hypernetwork that efficiently approximates a FEM-based facial simulation while generalizing over the comprehensive DECA model of human identities, facial expressions, and a wide range of material properties that can be locally adjusted without re-training. Along with SoftDECA, we introduce a pipeline for creating the needed high-resolution training data. Part of this pipeline is a novel layered head model

that densely positions the biomechanical anatomy within a skin surface while avoiding self-intersections.

## CCS CONCEPTS

• Computing methodologies → Physical simulation; Neural networks.

## KEYWORDS

Facial Animation, Physics-Based Simulation, Deep Learning

## ACM Reference Format:

Nicolas Wagner, Ulrich Schwanecke, and Mario Botsch. 2023. SoftDECA: Computationally Efficient Physics-Based Facial Animations. In *ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG '23)*, November 15–17, 2023, Rennes, France. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3623264.3624439>

## 1 INTRODUCTION

At present, research in the field of head avatars and facial animation is mainly concerned with obtaining photorealistic results through neural networks [Athar et al. 2022; Cao et al. 2022; Grassal et al. 2022; Zielonka et al. 2023] which can be operated on computationally rich systems. What currently falls short, however, is the inclusion of less capable hardware setups and circumstances in which geometry-based processing must be applicable. For this, various adaptations of linear blendshape models [Lewis et al. 2014] are still the usual means in production. Although linear facial models have been intensively researched and improved over the past decades,



This work is licensed under a Creative Commons Attribution International 4.0 License.

MIG '23, November 15–17, 2023, Rennes, France  
© 2023 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-0393-5/23/11.  
<https://doi.org/10.1145/3623264.3624439>

there are still known shortcomings like physically implausible distortions, loss of volume, anatomically impossible expressions, missing volumetric elasticity, or self-intersections. Physics-based simulations have been proposed that overcome most artifacts of linear blendshapes and allow for manifold additional functionalities [Barielle et al. 2016; Choi et al. 2022; Cong 2016; Ichim et al. 2017, 2016; Srinivasan et al. 2021; Yang et al. 2022]. Among them are medical applications such as visualization of weight changes, paralysis, or surgeries but also visual effects like aging, zombifications, gravity changes, and second-order effects. Moreover, it has recently been shown [Yang et al. 2022] that simulations with detailed extracted material information lead to much more realistic facial animations than linear models. The downside of physics-based facial animation models, however, is that these characteristically cause considerable computational overhead, giving rise to a body of literature on acceleration techniques. At this, the focus has been mostly on the evaluation of simulations in either manually constructed [Brandt et al. 2018] or learned subspaces [Holden et al. 2019; Santesteban et al. 2020] as well as on corrective blendshapes [Ichim et al. 2016]. The learned subspace methods [Holden et al. 2019] have proven to be more general and flexible, which is why in SoftSMPL [Santesteban et al. 2020] they have already been successfully applied to full-body animations. Nonetheless, so far there is still no method that transfers these advancements in fast physics-based simulations to facial animations. The principal contribution of this work is closing this gap with a deep learning approach which we call SoftDECA.

SoftDECA is a novel neural network architecture that efficiently animates faces while closely following a dynamic physics-based model. Although our method is universal in the sense that arbitrary physics-based facial animations can be considered, we focus on approximating a combination of state-of-the-art anatomically plausible and volumetric finite element methods (FEM) [Cong and Fedkiw 2019; Cong 2016; Ichim et al. 2017, 2016]. For this, we propose a novel adaption of hypernetworks [Ha et al. 2016] which yields inference times of about 10ms on consumer-grade CPUs and has the same programming interface as standard linear blendshapes. More precisely, we train SoftDECA to be applied as an add-on to arbitrary human blendshape rigs that follow the ARKit system<sup>1</sup>.

At the same time, SoftDECA is easily deployable without the need for elaborated personalizations or retraining, as we collect an extensive corpus of training examples. These examples cover a reasonable domain of the targeted FEM and bring together multiple data sources such as CT head scans to reflect the anatomy of heads, 3D head reconstructions in the wild that capture diverse head shapes (DECA [Feng et al. 2021]), and facial expressions in the form of recorded ARKit blendshape weights from dyadic conversational situations. The resulting overall training set facilitates a strong generalization of SoftDECA across human identities, facial expressions, and broad areas of the parameter manifold of the targeted FEM model. In contrast to earlier methods [Holden et al. 2019; Santesteban et al. 2020], the ability to generalize across FEM parameters makes extensive and efficient artistic interventions possible, with SoftDECA even supporting localized material adjustments.

As an additional contribution, we present a novel layered head model (LHM) that represents all training instances in a standardized way. Unlike fully or partially tetrahedralized volumetric meshes conventionally used for FEM, the LHM has additional enveloping wraps around bones, muscles, and skin. Based on these wraps, we describe a data-driven fitting procedure that positions muscles and bones within a neutral head while avoiding intersections of the various anatomic structures. A characteristic that was mostly not of concern in previous manually crafted physics-based facial animations but can otherwise lead to numerical instabilities in our automated training data generation approach.

## 2 RELATED WORK

### 2.1 Personalized Anatomical Models

Algorithms that create personalized anatomical models can essentially be distinguished according to two paradigms: *heuristic-based* and *data-driven*. Considering heuristic-based approaches, Anatomy Transfer [Ali-Hamadi et al. 2013] applies a space warp to a template anatomical structure to fit a target skin surface. The skull and other bones are only deformed by an affine transformation. A similar idea is proposed by Gilles et al. [2010]. While they also implement a statistical validation of bone shapes, the statistics are collected from artificially deformed bones. In [Ichim et al. 2016; Kadlecěk et al. 2016], an inverse physics simulation was used to reconstruct anatomical structures from multiple 3D expression scans. Saito et al. [2015] simulate the growth of soft tissue, muscles, and bones. A musculoskeletal biomechanical model is fitted from sparse measurements in [Schleicher et al. 2021] but not qualitatively evaluated.

There are only a few data-driven approaches because combined data sets of surface scans and CT, or CT and DXA images are hard to obtain for various reasons (e.g. data privacy or unnecessary radiation exposure). The recent work OSSO [Keller et al. 2022] predicts full body skeletons from 2000 DXA images that do not carry precise 3D information. Further, bones are positioned within a body by predicting only three anchor points per bone group and not avoiding intersections between skin and skull. A model that prevents skin-skull intersections and also considers muscles is based on fitting encapsulating wraps instead of the anatomy itself [Komaritzan et al. 2021]. However, no accurate algorithm based on medical imaging but a BMI (body mass index) regressor [Maalin et al. 2021] is used to position the wraps. A much more accurate, pure face model, was developed by Achenbach et al. [2018]. Here, CT scans are combined with optical scans by a multilinear model (MLM) which can map from skulls to faces and vice versa. As before, no self-intersections are prevented and only bones are fitted. Building on the data from [Achenbach et al. 2018] and following the idea of a layered body model [Komaritzan et al. 2021], we create a statistical layered head model including musculature that avoids self-intersections.

### 2.2 Physics-Based Facial Animation

A variety of techniques for animating faces have been developed in the past [Bradley et al. 2010; Ichim et al. 2015; Parke 1991; Zhang et al. 2008]. Data-driven models [Ichim et al. 2016; Lewis et al. 2014, 2005], which have recently been significantly improved by deep learning [Athar et al. 2022; Cao et al. 2022; Feng et al. 2021; Garbin

<sup>1</sup><https://developer.apple.com/>

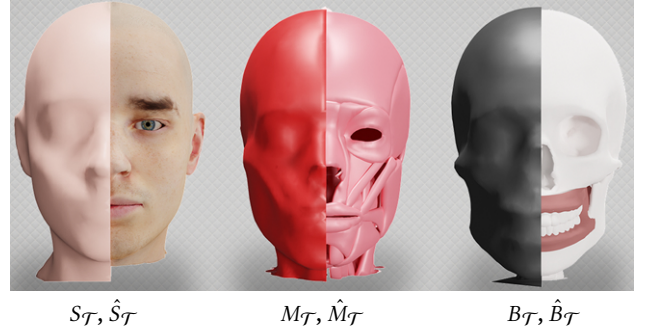
et al. 2022; Song et al. 2020; Zheng et al. 2022], are certainly dominant. Due to their simplicity and speed, linear blendshapes [Lewis et al. 2014] are still most commonly used in demanding applications and whenever no computationally rich hardware is available. Physics-based models have been developed for a long time [Sifakis et al. 2005] and avoid artifacts like implausible contortions and self-intersections, but due to their complexity and computational effort, they are rarely used. The pioneering work of Sifakis et al. [2005] is the first fully physics-based facial animation. The simulation is conducted on a personalized tetrahedron mesh, which can only be of a limited resolution due to a necessary dense optimization problem. With Phace [Ichim et al. 2017], this problem was overcome by an improved physics simulation. An art-directed muscle model [Bao et al. 2019; Cong and Fedkiw 2019; Cong 2016] additionally represents muscles as B-splines and allows control of expressions via trajectories of spline control points. A solely inverse model for determining the physical properties of faces was proposed in [Kadleček and Kavan 2019].

Hybrid approaches add surface-based physics to linear blendshapes for more detailed facial expressions [Barrielle et al. 2016; Bickel et al. 2008; Choi et al. 2022; Kozlov et al. 2017]. However, by construction, they can not model volumetric effects. With volumetric blendshapes [Ichim et al. 2016], a hybrid approach has been presented that combines the structure of linear blendshapes with volumetric physical and anatomical plausibility but can only achieve real-time performance through personalized corrective blendshapes.

Considering soft bodies in general, deep learning approaches have been investigated to approximate physics-based simulations. For instance, in [Casas and Otaduy 2018; Santesteban et al. 2020] the SMPL (Skinned Multi-Person Linear Model) proposed in [Loper et al. 2015] was extended with secondary motion. Recently, [Choi et al. 2022; Srinivasan et al. 2021; Yang et al. 2022] developed methods to learn the particular physical properties of objects and faces. However, these approaches must be retrained for unseen identities and are slow in inference. A fast and general approach for learning physics-based simulations is introduced in [Holden et al. 2019]. Unfortunately, they focused on reflecting the dynamics of single objects with limited complexity. We present a real-time capable deep learning approach to physics-based facial animations that does not need to be retrained and maintains the control structure of standard linear blendshapes. Additionally, none of the previously described deep learning methods tackle the challenging creation of facial training data, which we also address in this work.

### 3 METHOD

The foundation of the SoftDECA animation system is a novel layered head representation (Section 3.1). Starting from there, we design a FEM-based facial animation system (Sections 3.2 & 3.3) and demonstrate how to distill it into a defining dataset (Section 3.4). With this dataset, we train a newly designed hypernetwork (Section 3.5) as a real-time capable approximation of the animation system.



**Figure 2: All components of the layered head model template  $\mathcal{T}$ . Skin  $S_{\mathcal{T}}$ , skin wrap  $\hat{S}_{\mathcal{T}}$ , muscles  $M_{\mathcal{T}}$ , muscles wrap  $\hat{M}_{\mathcal{T}}$ , skull  $B_{\mathcal{T}}$ , and the skull wrap  $\hat{B}_{\mathcal{T}}$ .**

#### 3.1 Layered Head Model

**3.1.1 Structure.** We represent a head  $\mathcal{H} = \rho_{\mathcal{H}}(\mathcal{T})$  with neutral expression through a component-wise transformation  $\rho_{\mathcal{H}}$  of a layered head model template

$$\mathcal{T} = (S_{\mathcal{T}}, B_{\mathcal{T}}, M_{\mathcal{T}}, \hat{S}_{\mathcal{T}}, \hat{B}_{\mathcal{T}}, \hat{M}_{\mathcal{T}}), \quad (1)$$

that consists of six triangle meshes.  $S_{\mathcal{T}}$  describes the skin surface including the eyes, the mouth cavity, and the tongue,  $B_{\mathcal{T}}$  the surface of all skull bones and teeth,  $M_{\mathcal{T}}$  the surface of all muscles and the cartilages of the ears and nose.  $\hat{S}_{\mathcal{T}}$  is the skin wrap, i.e. a closed wrap enveloping  $S_{\mathcal{T}}$ ,  $\hat{B}_{\mathcal{T}}$  the skull wrap that envelops  $B_{\mathcal{T}}$ , and  $\hat{M}_{\mathcal{T}}$  the muscle wrap that envelops  $M_{\mathcal{T}}$ . Other anatomical structures are omitted for simplicity. The template structures  $S_{\mathcal{T}}$ ,  $B_{\mathcal{T}}$ , and  $M_{\mathcal{T}}$  were designed by an experienced digital artist. The skin, skull, and muscle wraps  $\hat{S}_{\mathcal{T}}$ ,  $\hat{B}_{\mathcal{T}}$ , and  $\hat{M}_{\mathcal{T}}$  have the same triangulation and were generated by shrink-wrapping a sphere as close as possible to the corresponding surfaces without intersections. The complete template is shown in Figure 2.

Due to the shared triangulation, the wraps of the LHM also define a soft tissue tet mesh  $\mathbb{S}_{\mathcal{T}}$  (i.e. between the skin and the muscle wraps) and a muscle tissue tet mesh  $\mathbb{M}_{\mathcal{T}}$  (i.e. between the muscle and the skull wraps). For this, each triangle prism that can be spanned between corresponding wrap faces is canonically split into three tets. The complexities of all template components are given in the supp. material. In the following, we will state the number of vertices of a mesh as  $|\cdot|_v$  and the number of faces as  $|\cdot|_f$ .

**3.1.2 Fitting.** Later on, creating training data requires finding

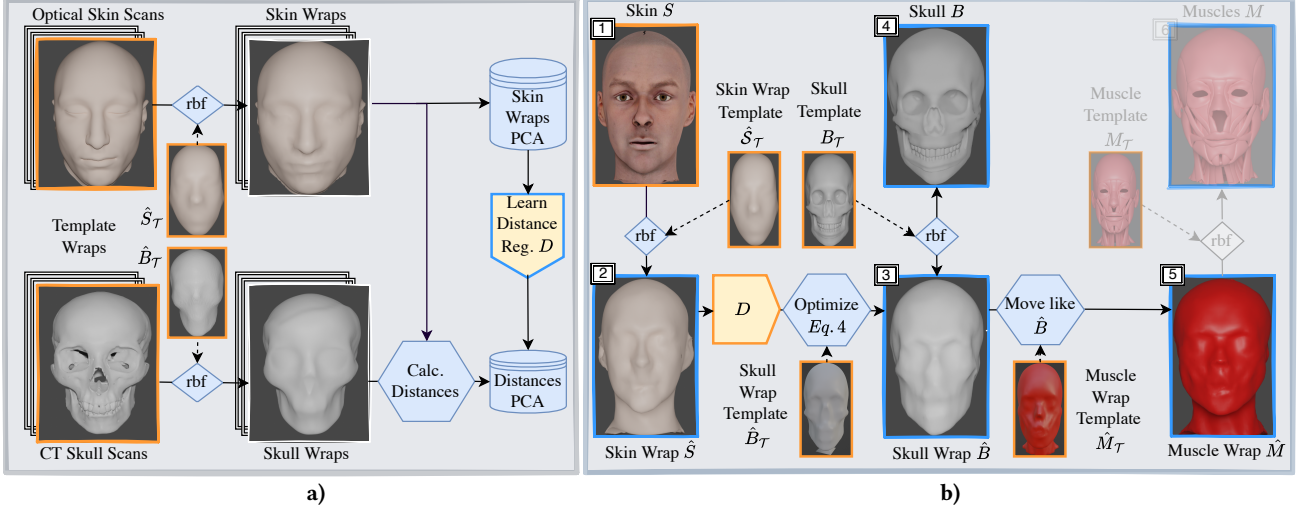
$$(S, B, M, \hat{S}, \hat{B}, \hat{M}) = \rho_{\mathcal{H}}(\mathcal{T}) \quad (2)$$

when only the skin surface  $S$  of the head  $\mathcal{H}$  is known. To this end, we rely on a hybrid approach that positions the skull in a data-driven manner while the remaining template components are fitted by heuristics that ensure anatomic plausibility and avoid self-intersections.

As the first of the remaining template meshes, we fit the skin wrap by setting

$$\hat{S} = \text{rbf}_{S_{\mathcal{T}} \rightarrow S}(\hat{S}_{\mathcal{T}}). \quad (3)$$

The RBF function is a space warp based on triharmonic radial basis functions [Botsch and Kobbelt 2005] that is calculated from the



**Figure 3: a) The training scheme of the skin to skull wrap distances regressor  $D$ . b) Procedural overview of the layered head model fitting algorithm. Orange frames indicate input, blue frames output. The enumeration reflects the fitting order. Step 6 is shown only for the sake of completeness.**

template skin surface  $S_{\mathcal{T}}$  to the target  $S$  and subsequently applied to the template skin wrap  $\hat{S}_{\mathcal{T}}$ . By the construction of RBFs, the skin wrap will be warped semantically consistent and stick close to the targeted skin surface.

Next, we fit the skull wrap  $\hat{B}$  by invoking a linear regressor  $D$  that predicts the distances from the vertices of  $\hat{S}$  to the corresponding vertices of  $\hat{B}$  and subsequently minimizing with projective dynamics [Bouaziz et al. 2014]

$$\arg \min_X w_{\text{rect}} E_{\text{rect}}(X, \hat{S}_{\mathcal{T}}) + w_{\text{dist}_2} E_{\text{dist}_2}(X, \hat{S}, D(\hat{S})) + w_{\text{curv}} E_{\text{curv}}(X, \hat{B}_{\mathcal{T}}). \quad (4)$$

Here,  $E_{\text{dist}_2}$  ensures that the predicted distances are adhered to,  $E_{\text{curv}}$  is a curvature regularization of the skull wrap, and  $E_{\text{rect}}$  avoids shearing between corresponding skin and skull wrap faces. The distances are set to a minimum value if they fall below a threshold, thus, avoiding skin-skull intersections. To ease the flow of reading, we give formal descriptions of the energy components in the supplement. The optimization is initialized with  $X = \hat{S} - D(S) \cdot n(\hat{S})$  where  $n(\hat{S})$  are area-weighted vertex normals.  $D$  is trained on the dataset of [Gietzen et al. 2019] (SKULLS) that relates CT skull measurements to optical skin surface scans. In Figure 3 a) the linear regressor training is depicted.

The muscle wrap  $\hat{M}$  is fitted by positioning its vertices at the same absolute distances between the corresponding skin and skull wrap vertices as in the template, and only passing on ten percent of the relative distance changes compared to the template. This approach assumes that the muscle mass in the facial area is only moderately affected by body weight and skull size.

The skull mesh is placed by setting

$$B = \text{rbf}_{\hat{B}_{\mathcal{T}} \rightarrow \hat{B}}(B_{\mathcal{T}}). \quad (5)$$

The properties of the RBF space warp ensure that the skull mesh remains within the skull wrap if the wrap is of sufficient resolution.

The muscle mesh could be placed in a similar fashion but is not needed in our pipeline any further.

Finally, the soft and muscle tissue tet meshes  $\mathbb{S}$  and  $\mathbb{M}$  can be constructed as described before. On average, the complete fitting pipeline takes about 500ms on an AMD Threadripper Pro 3995wx processor. Figure 3b) visualizes the overall fitting process.

### 3.2 SoftDECA Animation System

Building on the LHM representation, we now introduce the SoftDECA animation system. For this, the classical concept of linear blendshapes is reviewed first. Thereupon, the dynamic physics-based facial simulation system which is at the core of SoftDECA is derived.

For a specific head, a linear blendshape model consists of  $n$  surface blendshapes

$$\{S^i\}_{i=1}^n \quad (6)$$

which animate an unknown facial expression  $S_t$  as a linear combination

$$S_t = \sum_{i=1}^N w_t^i S^i, \quad (7)$$

where the blending weights  $w_t$  determine the share of each blendshape in the expression at frame  $t$ .

To achieve the same animation with a physical model  $\phi$ , one typically differentiates between forward and inverse methods. Without loss of generality, we consider the inverse method in the following. Here, the expression  $S_t$  is converted into the (in the Euclidean sense) closest  $\phi$ -plausible solution by  $\phi^\dagger$  to

$$T_t = \phi^\dagger(S_t, \mathbf{p}), \quad (8)$$

where  $\mathbf{p}$  is a vector of material and simulation parameters on which  $\phi$  depends. For including second-order effects as well, Equation (8) expands to

$$T_t = \phi^\dagger(\gamma S_t + 2\alpha T_{t-1} - \beta T_{t-2}, \mathbf{p}). \quad (9)$$



The SoftDECA animation system operates in the same manner, but the right-hand side is approximated by a computationally efficient neural network  $f$ .

Next, we will describe our realization of  $\phi^\dagger$  and how to create representative examples. Nonetheless, please note that SoftDECA is not restricted to a particular realization of  $\phi^\dagger$ .

### 3.3 Physics-Based Simulations

We implement anatomically plausible inverse physics  $\phi^\dagger$  as a projective dynamics energy  $E_{\phi^\dagger}$ . At this, state-of-the-art FEM models [Cong 2016; Ichim et al. 2017; Kadleček and Kavan 2019] are merged by applying separate terms for soft tissue, muscle tissue, the skin, the skull, and auxiliary components.

**3.3.1 Energy.** Considering the soft tissue  $\mathbb{S}$ , we closely follow the model of [Ichim et al. 2017] and impose

$$E_{\mathbb{S}} = w_{\text{vol}} \sum_{t \in \mathbb{S}} E_{\text{vol}}(t) + w_{\text{str}} \sum_{t \in \mathbb{S}} \mathbb{1}_{\sigma_{F(t)} > \epsilon} E_{\text{str}}(t), \quad (10)$$

which for each tet  $t$  penalizes change of volume and strain, respectively. Strain is only accounted for if the largest eigenvalue  $\sigma_{F(t)}$  of the stretching component of the deformation gradient  $F(t) \in \mathbb{R}^{3 \times 3}$  grows beyond  $\epsilon$ .

To reflect the biological structure of the skin, we additionally formulate a dedicated strain energy

$$E_S = \sum_{t \in \mathbb{S}} E_{\text{str}}(t) \quad (11)$$

on each triangle  $t$  of the skin which, to the best of our knowledge, has not been done before.

For the muscle tets  $\mathbb{M}$ , we follow Kadleček et al. [2019] that capturing fiber directions for tetrahedralized muscles is in general too restrictive. Hence, only a volume-preservation term

$$E_{\mathbb{M}} = w_{\text{vol}} \sum_{t \in \mathbb{M}} E_{\text{vol}}(t) \quad (12)$$

is applied for each tet in  $\mathbb{M}$ .

The skull is not tetrahedralized as it is assumed to be non-deformable even though it is rigidly movable. The non-deformability of the skull is represented by

$$E_B = \sum_{t \in B} E_{\text{str}}(t) + \sum_{x \in B} E_{\text{curv}}(x, B), \quad (13)$$

i.e. a strain  $E_{\text{str}}$  on the triangles  $t$  and mean curvature regularization on the vertices  $x$  of the skull  $B$ . We do not model the non-deformability as a rigidity constraint due to the significantly higher computational burden.

To connect the muscle tets as well as the eyes to the skull, connecting tets are introduced similar to the sliding constraints in [Ichim et al. 2017]. For the muscle tets, each skull vertex connects to the closest three vertices in  $\mathbb{M}$  to form a connecting tet. For the eyes, connecting tets are formed by connecting each eye vertex to the three closest vertices in  $B$ . On these connecting tets, the energy  $E_{\text{con}}$  with the same constraints as in Equation (10) is imposed. By this design, the jaw and the cranium are moved independently from each other through muscle activations but the eyes remain rigid and move only with the cranium.

Finally, the energy

$$E_{\text{inv}} = \sum_{x \in S} E_{\text{tar}}(x, S_t) \quad (14)$$

of soft Dirichlet constraints is added, attracting the skin surface  $S$  vertices to the targeted expression  $S_t$ .

The weighted sum of the aforementioned energies gives the total energy

$$E_{\phi^\dagger} = w_{\mathbb{S}} E_{\mathbb{S}} + w_{\mathbb{M}} E_{\mathbb{M}} + w_B E_B + w_{\text{mstr}} E_{\text{mstr}} + w_S E_S + w_{\text{con}} E_{\text{con}} + w_{\text{inv}} E_{\text{inv}} \quad (15)$$

of the inverse model  $\phi^\dagger$ . Altogether,  $\phi^\dagger$  results in an expression  $T_t$  that in a Euclidean sense is close to the target  $S_t$  but is plausible w.r.t. the imposed constraints.

**3.3.2 Collisions.** Finally, self-intersections are resolved between colliding lips or teeth in a subsequent projective dynamics update as in [Komaritzan and Botsch 2018].

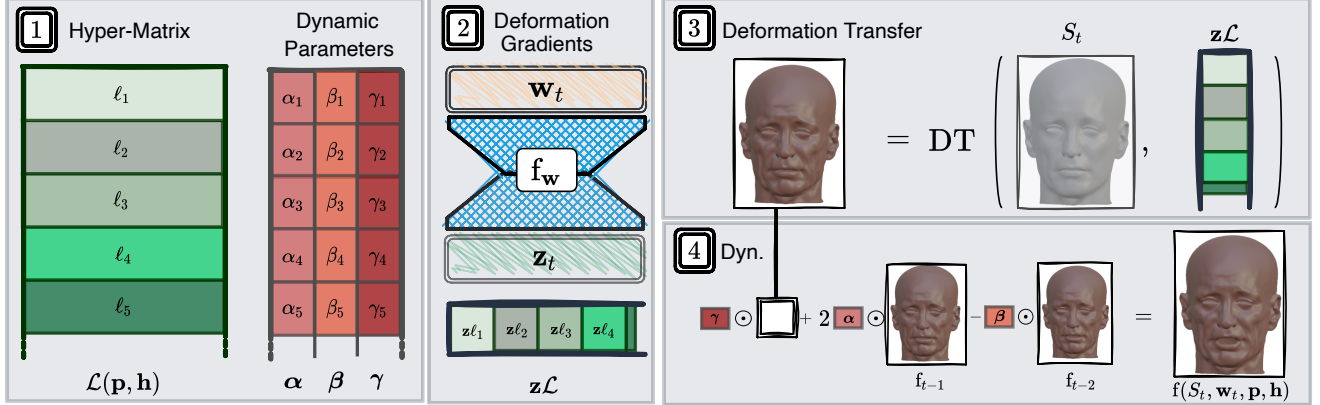
**3.3.3 Parameters.** The construction of  $\phi^\dagger$  also implies parts of the parameter vector  $\mathbf{p}$ . As such, the dynamics parameters  $\alpha, \beta, \gamma$ , weights  $w_*$  of all the constraints, but also other attributes of the constraints can be considered. For example, the target volume in  $E_{\text{vol}}$  or scaling factors of the skull bones. Additionally, we include constant external forces like gravity strength and direction into  $\mathbf{p}$ . An overview of all parameters we use and the corresponding value ranges is given in the supp. material.

### 3.4 Training Data

By the definition of the animation system in Equation (9), a representative training dataset  $\mathcal{D}$  must consist of examples that relate diverse facial expressions created via linear blendshapes to the corresponding surfaces that conform  $\phi$ . Further, to capture dynamic effects, the exemplary facial expressions have to form reasonable sequences. This dataset must also cover a variety of distinct head shapes as well as simulation parameters.

In the following, we describe a pipeline for creating instances of such a dataset, which can be roughly divided into six high-level steps.

- (1) We start by randomly drawing a neutral skin surface  $S$  from DECA [Feng et al. 2021], a comprehensive high-resolution face model. More specifically, we randomly draw an image from the Flickr-Faces-HQ [Karras et al. 2019] dataset and let DECA determine the corresponding neutral head shape as well as a latent representation  $\mathbf{h}$ .
- (2) Next, the template LHM  $\mathcal{T}$  is aligned with the skin surface  $S$  as described in Section 3.1.
- (3) In the third step, deformation transfer [Botsch et al. 2006] is used to transfer ARKit surface-based blendshapes to  $S$ .
- (4) Subsequently, we create an expression sequence  $\mathbf{S} = (S_t)_{t=0}^m$  of length  $m+1$  by applying a sequence of blendshape weights  $\mathbf{w} = (w_t)_{t=0}^m$ . The blendshape weights are obtained from 8 around 10 minutes long dyadic conversations recorded with a custom iOS app.
- (5) As the final step before the  $\phi$ -plausible counterpart of  $\mathbf{S}$  can be generated, simulation parameters have to be sampled on a proper domain. For continuous parameters, we expect the



**Figure 4: An overview of the SoftDECA facial animation. In Step 1), the hyper-tensor and the dynamic parameters are determined once for an animation. Subsequently, steps 2-4 are repeatedly evaluated per frame. In Step 2), per-face deformation gradients are calculated which are applied in Step 3) to form a facial expression. In Step 4), dynamic effects are added.**

user to specify lower and upper bounds beforehand. Subsequently, for each parameter in  $\mathbf{p}$ , we independently sample a value between the respective bounds with uniform distribution. Discrete parameters are handled in the same way but without respecting particular constraints.

- (6) Finally,  $\mathbf{T} = (\phi^\dagger(S_t))_{t=0}^m$  is computed and  $(\mathbf{T}, \mathbf{S}, \mathbf{w}, \mathbf{p}, \mathbf{h})$  is added to  $\mathcal{D}$ . Evaluating one time step takes approximately 10 seconds on an AMD Threadripper Pro 3995wx.

### 3.5 Hypernetwork

**3.5.1 Architecture & Training.** Having training data, we can now design a computationally efficient neural network  $f$  to approximate the physics-based simulation from Equation 9. Irrespective of a particular architecture, the training goal implied by  $\mathcal{D}$  is to optimize on each frame

$$\min_{\mathbf{f}} \sum_{(\mathbf{T}, \mathbf{S}, \mathbf{w}, \mathbf{p}, \mathbf{h}) \in \mathcal{D}} \sum_{t=0}^m \|T_t - f(S_t, \mathbf{w}_t, \mathbf{p}, \mathbf{h})\|_2. \quad (16)$$

In words,  $f$  is trained to approximate the  $\phi$ -conformal expressions from the the linearly blended expressions  $S_t$ , the blending weights  $\mathbf{w}_t$ , simulation parameters  $\mathbf{p}$ , and the head descriptions  $\mathbf{h}$ . Hence, leaving out dynamic effects to begin with, the probably most naive approach would be to learn  $f$  to directly predict vertex positions. However, this would not allow the usage of personalized blendshapes at inference time that have not been used in the curation of  $\mathcal{D}$ . Therefore, we separate  $f$  into two high-level components

$$f(S_t, \mathbf{w}_t, \mathbf{p}, \mathbf{h}) = \text{DT}(S_t, f_{DG}(\mathbf{w}_t, \mathbf{p}, \mathbf{h})), \quad (17)$$

where  $\text{DT}$  is a deformation transfer function as in [Sumner and Popović 2004] that applies  $3 \times 3$  per-face deformation gradients (DGs) predicted by  $f_{DG}(\mathbf{w}_t, \mathbf{p}, \mathbf{h}) \in \mathbb{R}^{|S| \times 9}$  to the linearly blended  $S_t$ . By doing so,  $f$  can also be applied to a facial expression  $S_t$  which has been formed by unseen personalized blendshapes while still achieving close approximations of  $\phi^\dagger$ . Fortunately, the evaluation of  $\text{DT}$  is not more than efficiently finding a solution to a pre-factorized linear equation system.

To implement the DG prediction network  $f_{DG}$ , we evaluated multiple network architectures such as set transformers [Lee et al.

2019], convolutional networks on geometry images, graph neural networks [Scarselli et al. 2008], or implicit architectures [Mildenhall et al. 2021], but all have exhibited substantially slower inference speeds while reaching a similar accuracy as a multi-layer perceptron (MLP). Nevertheless, a plain MLP does not discriminate between inputs that change per frame  $t$  and inputs that have to be computed only once. Therefore, we propose an adaptation of a hypernetwork MLP [Ha et al. 2016] to implement  $f_{DG}$  in which the conditioning of  $f_{DG}$  with respect to the simulation parameters as well as the DECA identity is done by manipulating network parameters. Formally, we implement

$$f_{DG}(\mathbf{w}_t, \mathbf{p}, \mathbf{h}) = \mathbf{z}_t \mathcal{L}(\mathbf{p}, \mathbf{h}), \quad (18)$$

where  $\mathcal{L}(\mathbf{p}, \mathbf{h}) \in \mathbb{R}^{32 \times |S| \times 9}$  returns a tensor that only has to be calculated once for all frames and  $\mathbf{z}_t = f_w(\mathbf{w}_t) \in \mathbb{R}^{32}$  is the result of a small standard MLP that processes the blending weights at every frame  $t$ . Each matrix  $\ell_i \in \mathbb{R}^{32 \times 9}$  in  $\mathcal{L}(\mathbf{p}, \mathbf{h})$  corresponds to a face in  $S$  and the entries are calculated as

$$\ell_i = f_{\text{ph}}(\mathbf{p}, \mathbf{h}, \pi(i)). \quad (19)$$

Again,  $f_{\text{ph}}$  is a small MLP and  $\pi$  is a trainable positional encoding. Please consult the supp. material for detailed dimensions of all networks and see Figure 4 for a structural overview of  $f$ .

**3.5.2 Localization.** The architecture described above offers extensive possibilities for artistic user interventions at inference time. For instance, different simulation parameters  $\mathbf{p}_i$  can be used per face  $i$  by changing Equation (19) to

$$\ell_i = f_{\text{ph}}(\mathbf{p}_i, \mathbf{h}, \pi(i)), \quad (20)$$

which enables a localized application of different material models. The  $\text{DT}$  function ensures that the models are smoothly combined.

**3.5.3 Dynamics.** Given that locally differing simulation parameters are not reflected in the training data, existing approaches to integrate dynamics in deep learning [Holden et al. 2019; Santesteban et al. 2020], cannot be adopted. Therefore, we again use the hypernetwork concept to achieve a piecewise-linear dynamics

approximation. More precisely, we recursively extend  $f$  to

$$\begin{aligned} f(S_t, \mathbf{w}_t, \mathbf{p}, \mathbf{h}) = & \gamma \odot \text{DT}(S_t, f_{DG}(\mathbf{w}_t, \mathbf{p}, \mathbf{h})) \\ & + 2\alpha \odot f(S_{t-1}, \mathbf{w}_{t-1}, \mathbf{p}, \mathbf{h}) \\ & - \beta \odot f(S_{t-2}, \mathbf{w}_{t-2}, \mathbf{p}, \mathbf{h}), \end{aligned} \quad (21)$$

where  $\alpha, \beta, \gamma \in \mathbb{R}^{32 \times |S|_v}$  contain per-vertex dynamics parameters. The first row of Equation (21) is the same as in Equation (17) but the second and third rows allow for dependencies on the previous two frames. Each entry of  $\alpha, \beta, \gamma$  is calculated as in Equation (20) but with dedicated MLPs  $f_\alpha, f_\beta, f_\gamma$ . As a result,  $\alpha, \beta, \gamma$  are again not time-dependent and only have to be calculated once.

## 4 EXPERIMENTS

Before demonstrating the accuracy and efficiency of SoftDECA (Section 4.2), we first evaluate the fitting precision of the LHM (Section 4.1).

### 4.1 LHM Fitting

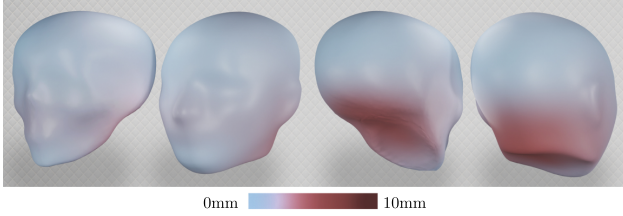


Figure 5: The per-vertex mean L2-error of the LHM fitting.

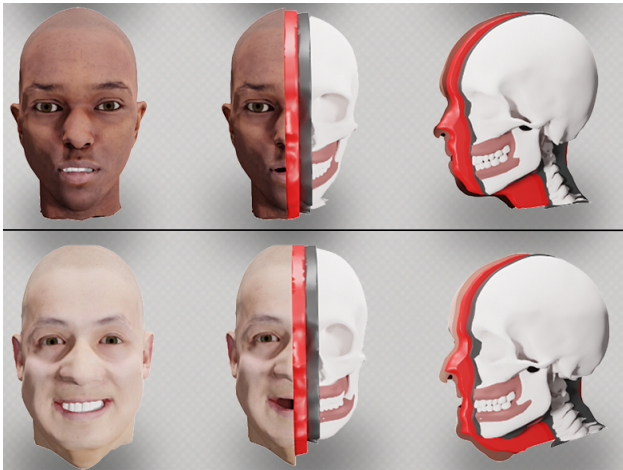


Figure 6: Exemplary fits of the LHM components skull wrap, muscle wrap, and skull.

The fitting of the LHM is mainly composed of the data-driven positioning of the skull wrap and the subsequent heuristic fitting of the muscle wrap. We evaluate the crucial fitting of the skull wrap with the open-source CT SKULLS [Gietzen et al. 2019] dataset. Since this dataset consists of 43 instances only, a leave-one-out validation is performed in which the vertex-wise L2 errors are measured. Earlier methods that position the skull within the head, mainly use

sparse soft tissue statistics measured in normal directions starting from very few points on the skull [Beeler and Bradley 2014; Ichim et al. 2016]. We compare our approach to the multilinear model of Achenbach et al. [2018; 2019], who have shown a more robust and precise positioning by capturing dense soft tissue statistics as radii of spheres surrounding the skull.

Both models cannot achieve a medical-grade positioning with errors between approximately 2 mm and 4 mm. The MLM achieves a higher precision with a mean error of 1.98 mm than our approach that dispositions the skull by 3.83 mm on average. However, the MLM cannot prevent collisions that might crash physics-based simulations. Also, our fitting algorithm produces large errors only in regions that are of less importance for facial simulations as can be seen in Figure 5. The errors are predominately distributed in the back area of the skull since here the rectangular constraints of our fitting procedure can presumably no longer be aligned well with the skin wrap. Figure 6 displays fitting examples.

### 4.2 SoftDECA

**4.2.1 Dataset & Training.** To train and evaluate  $f$ , we assemble a dataset of 500k training and test instances by using the pipeline from Section 3.4. The parallelized dataset creation took five days and required one terabyte of storage. To match the uneven sizes of the parameter spaces, 75% of the produced data is static data in which all but the dynamic parameters  $\alpha, \beta, \gamma$  are sampled and only the remaining 25% of the data is simulated dynamically. As a result, 6250 dynamic sequences have been generated, each of which has a length of 16 while the static examples consist of only one frame per example. To initialize the dynamic sequences with a reasonable velocity, a longer sequence of length 2048 has been simulated with fixed dynamics parameters a priori. For each dynamic sequence, a random observed velocity of the long sequence is drawn as the initialization. The dataset is split in 90% for training and 10% for testing while neither the same identity nor the same simulation parameters nor the same facial expression occurs in both.

For training, the Adam optimizer performs 200k update steps with a learning rate of 0.0001. The learning rate is linearly decreased to 0.00005 over the course of training and a batch size of 128 is applied. In total, the training specifications result in an approximate runtime of 8 hours on an NVIDIA A6000. The comparatively short training time can straightforwardly be explained by the efficient network design and the less noisy training data than usually encountered for instance in image-based deep learning. We quantitatively evaluate SoftDECA based on the L2 reconstruction error with respect to the targeted physics-based simulation and the computational runtimes. Besides, we compare it against the Subspace Neural Physics (SNP) [Holden et al. 2019] and the SoftSMPL [Sansteban et al. 2020] architectures adapted to facial simulations. These are, to the best of our knowledge, state-of-the-art methods for fast approximations of physics-based simulations. An overview of all results is given in Table 1. The stated runtimes are averages of ten runs measured on a consumer-grade Intel i5 12600K processor. All implementations rely on PyTorch<sup>2</sup>.

<sup>2</sup><https://pytorch.org>

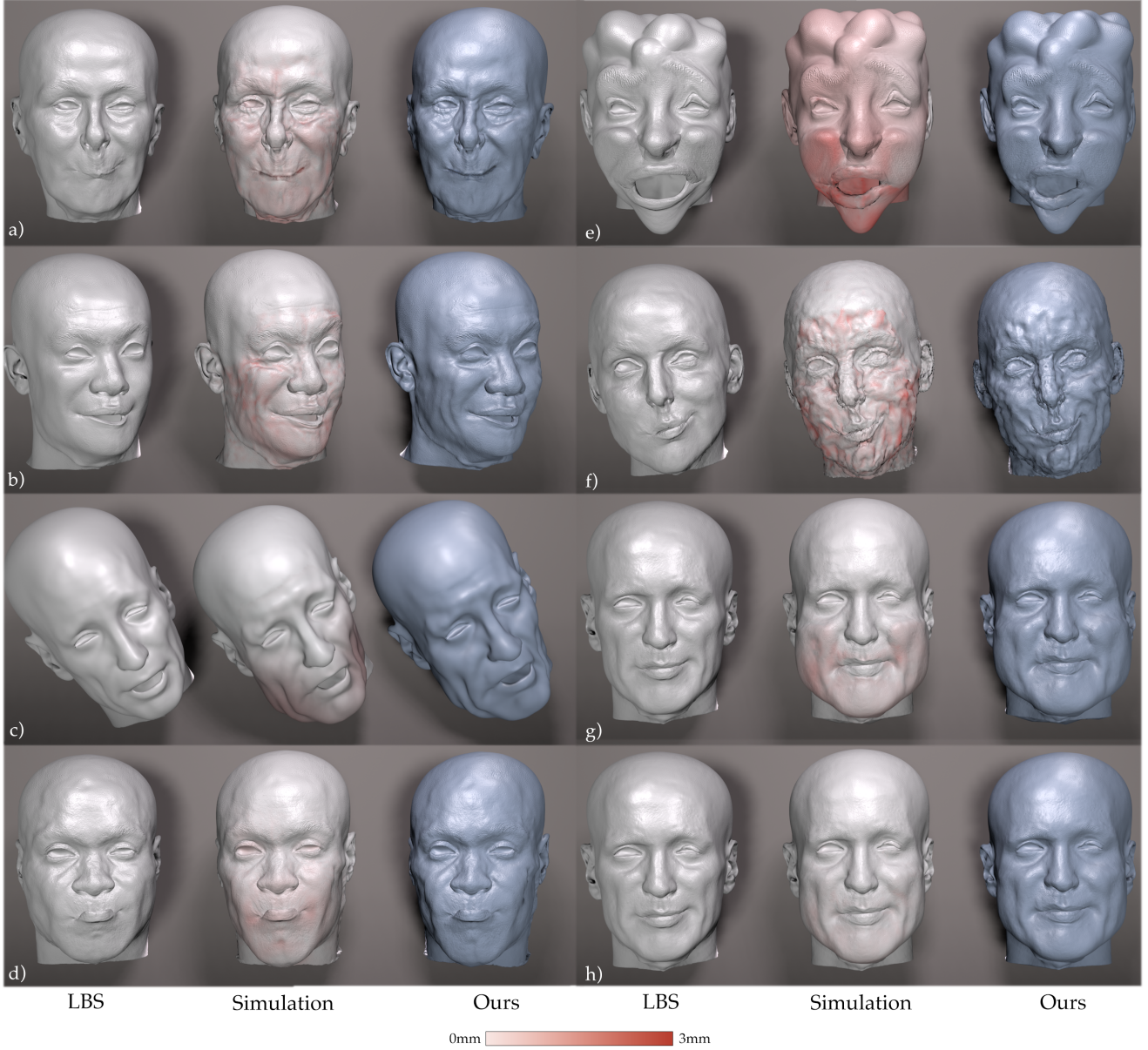


Figure 7: Exemplary results of SoftDECA in comparison to the targeted physics-based facial simulation as well as the inputted linear blendshape expressions. Reconstruction errors are plotted on the simulated expressions.

Table 1: SoftDECA test results in comparison to adapted SNP [Holden et al. 2019] and SoftSMPL [Santesteban et al. 2020] architectures as well as ablations. The runtimes are averages measured on a consumer-grade Intel i5 12600K processor. External refers to the 3Dscanstore dataset. Small and large correspond to the size of the inspected MLP.

Model	Ours			SoftSMPL			SNP	Ablation	
	Static	Dynamic	External	Static (Small)	Static (Large)	Dynamic	Dynamic	Face-wise	Only Vertices
Error in <i>mm</i>	0.23	0.41	0.44	1.67	0.16	0.22	0.14	0.17	0.16
Time in <i>ms</i>	7.45	9.87	7.45	7.62	46.61	47.39	46.61	34.92	0.72

**4.2.2 Quantitative Analysis.** First of all, SoftDECA provides very close approximations in static and dynamic animations with average test reconstruction errors of only  $0.22mm$  and  $0.41mm$ , respectively. Hence, overall, it becomes evident that SoftDECA generalizes across different human identities, facial expressions, and simulation parameters. Nevertheless, the expressions are all obtained from unpersonalized blendshapes which is why we further evaluate on a static external dataset from the 3DScanstore<sup>3</sup>. In this dataset, for each of seven heads, between 20 and 35 scanned facial expressions are available which we convert into personalized ARKit blendshapes using example-based facial rigging [Li et al. 2010]. Starting from there, we create a test dataset as before. Although the 3DScanstore examples are likely not covered by the DECA distribution, the reconstruction error only slightly increases to  $0.44mm$ .

Despite the high approximation quality, SoftDECA needs only  $7.45ms$  to calculate a static frame on average while a runtime of  $9.87ms$  is needed for a dynamic frame. This brief runtime makes SoftDECA appealing even for demanding virtual reality applications. For applications in which unseen personalized blendshape are not desired, we also test a variant of SoftDECA that directly predicts vertex positions. This version achieves an accuracy of  $0.16mm$  and can be accelerated to only  $0.71ms$  per frame.

**4.2.3 Static Comparisons.** For static simulations, SoftDECA can only be compared to SoftSMPL as SNP is solely designed to approximate dynamic effects. Essentially, the difference between the SoftDECA and the SoftSMPL architecture is the difference between our hypernetwork MLP and a standard MLP. SoftSMPL is originally designed for full bodies and has a motion descriptor as input that describes a body and its state. Adapted to our case, these are the blendshape weights, simulation parameters, and the identity code. First, to keep the inference times approximately consistent, we employ the same network dimensions for the standard MLP as in the hypernetwork. As a result, the reconstruction error of the SoftSMPL MLP increases significantly to an average of  $1.67mm$ . Therefore, we additionally investigate a larger MLP which achieves approximately the same reconstruction error as SoftDECA. In turn, however, the runtime increases tremendously to  $46.61ms$ . Another canonical alternative to the hypernetwork is a standard MLP that in the last layer does not map to all DGs simultaneously but calculates the DGs face-wise. The reconstruction error is low with  $0.17mm$ , but the runtime is also high with  $34.92ms$ . Other architectures like CNNs, GNNs, or transformers could not be evaluated in real-time on a consumer-grade CPU with sufficient accuracy.

**4.2.4 Dynamic Comparisons.** For dynamic simulations, SoftDECA can be compared against both SoftSMPL and SNP. Contrary to SoftDECA, SoftSMPL and SNP compute dynamics in a latent space and not directly on vertices. Both differ from one another in that SoftSMPL additionally relies on a recurrent GRU network [Chung et al. 2014], whereas SNP is purely based on a standard MLP. In both cases, we compare solely with the *larger* network design mentioned earlier since we are mainly interested in evaluating the accuracy of our dynamic approximation and not in comparing runtimes. It can be observed that the SoftSMPL as well as the SNP design achieve slightly improved reconstruction errors with  $0.22mm$  and  $0.24mm$ ,

respectively. However, since both do not work vertex-wise, they are not suitable for locally varying simulation parameters.

**4.2.5 Qualitative Analysis.** A visual demonstration of SoftDECA's capabilities is given in Figure 7 where the SoftDECA predictions are contrasted with the targeted physics-based facial simulation. For instance, in a) it can be observed that, although collisions are not guaranteed to be removed, they remain largely dissolved. In b), the triangle strain of the skin is increased locally in the area of the cheeks, leading to the formation of wrinkles in this region. In c), it is demonstrated that external effects can also be included by means of increased gravity. A *surgical manipulation* is shown in d), in which the jaw is lengthened along the vertical axis in the neutral state while the volume of the head is maintained. The representation of a humanoid alien in e) illustrates the robustness of SoftDECA even outside the DECA distribution. This robustness is mainly achieved by transferring DGs instead of directly predicting vertex positions. Our interpretation of zombification is achieved in f) by growing the area of the skin. This effect highlights that SoftDECA is able to closely approximate such excessive high-frequency details, too. Finally, in g-h) we present how different weight additions can be simulated in a non-linear way. For this purpose, we raise the volume of the soft tissue by 20% and 40%. Due to the already comprehensive training domain of SoftDECA, many other effects can be animated in a computationally efficient way that are not displayed in Figure 7. We refer the reader to the supp. material where additional simulations are shown in a video including dynamic effects.

## 5 LIMITATIONS

Although SoftDECA inherits most of the advantages of physics-based facial animations, it lacks the intrinsic handling of interactive effects such as wind or colliding objects. Moreover, although we allow for extensive localized artistic interventions, mixtures of material properties have not been part of the training data. Incorporating such mixtures into the training data is difficult as it is hard to define an adequate mixture distribution. Nonetheless, the smooth material blending of SoftDECA visually appears to be a sufficient approximation.

## 6 CONCLUSION

In this work, we presented SoftDECA, a computationally efficient approximation of physics-based facial simulations even on consumer-grade hardware. With a few exceptions, most simulation capabilities are retained, such as dynamic effects, volume preservation, wrinkle generation, and many more. At this, SoftDECA's runtime is attractive for high-performance applications and low-budget hardware. Moreover, it is lightweight to deploy as it generalizes across different head shapes, facial expressions, and material properties. Finally, the ability to make localized changes after training constitutes an attractive framework for artistic customization.

We aim to improve SoftDECA in at least two directions. On the one hand, with an even more accurate anatomical model that represents e.g. trachea and esophagus more precisely. On the other hand, recent results [Romero et al. 2022] show that contact deformations can also be efficiently learned. Since people touch their faces dozens of times [Spille et al. 2021] a day, adding contact-handling for more realistic gestures may improve immersion significantly.

<sup>3</sup><https://www.3dscanstore.com>

## ACKNOWLEDGMENTS

This research was supported by the German Federal Ministry of Education and Research (BMBF) through the project HiAvA (ID 16SV8785).

## REFERENCES

- Jascha Achenbach, Robert Brylka, Thomas Gietzen, Katja zum Hebel, Elmar Schömer, Ralf Schulze, Mario Botsch, and Ulrich Schwanecke. 2018. A multilinear model for bidirectional craniofacial reconstruction. In *Proceedings of the Eurographics Workshop on Visual Computing for Biology and Medicine*. 67–76.
- Dicko Ali-Hamadi, Tiantian Liu, Benjamin Gilles, Ladislav Kavan, François Faure, Olivier Palombi, and Marie-Paule Cani. 2013. Anatomy transfer. *ACM transactions on graphics (TOG)* 32, 6 (2013), 1–8.
- ShahRukh Athar, Zexiang Xu, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. 2022. RigNeRF: Fully Controllable Neural 3D Portraits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20364–20373.
- Michael Bao, Matthew Cong, Stéphane Grabli, and Ronald Fedkiw. 2019. High-quality face capture using anatomical muscles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10802–10811.
- Vincent Barrielle, Nicolas Stoiber, and Cédric Cagniard. 2016. Blendforces: A dynamic framework for facial animation. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 341–352.
- Thabo Beeler and Derek Bradley. 2014. Rigid stabilization of facial expressions. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–9.
- Bernd Bickel, Manuel Lang, Mario Botsch, Miguel A Otaduy, and Markus H Gross. 2008. Pose-Space Animation and Transfer of Facial Details. In *Symposium on Computer Animation*. 57–66.
- Mario Botsch and Leif Kobbelt. 2005. Real-time shape editing using radial basis functions. In *Computer graphics forum*, Vol. 24. Blackwell Publishing, Inc Oxford, UK and Boston, USA, 611–621.
- Mario Botsch, Robert Sumner, Mark Pauly, and Markus Gross. 2006. Deformation transfer for detail-preserving surface editing. In *Vision, Modeling & Visualization*. Citeseer, 357–364.
- Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective dynamics: Fusing constraint projections for fast simulation. *ACM transactions on graphics (TOG)* 33, 4 (2014), 1–11.
- Derek Bradley, Wolfgang Heidrich, Tiberiu Popa, and Alla Sheffer. 2010. High resolution passive facial performance capture. In *ACM SIGGRAPH 2010 papers*. 1–10.
- Christopher Brandt, Elmar Eiseemann, and Klaus Hildebrandt. 2018. Hyper-reduced projective dynamics. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.
- Chen Cao, Tomas Simon, Jin Kyu Kim, Gabe Schwartz, Michael Zollhoefer, Shun-Suke Saito, Stephen Lombardi, Shih-En Wei, Danielle Belko, Shouo-I Yu, et al. 2022. Authentic volumetric avatars from a phone scan. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–19.
- Dan Casas and Miguel A Otaduy. 2018. Learning nonlinear soft-tissue dynamics for interactive avatars. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 1 (2018), 1–15.
- Byungkuk Choi, Haekwang Eom, Benjamin Mouscadet, Stephen Cullingford, Kurt Ma, Stefanie Gassel, Suzi Kim, Andrew Moffat, Millicent Maier, Marco Revelant, et al. 2022. Anatomy: an Animator-centric, Anatomically Inspired System for 3D Facial Modeling, Animation and Transfer. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.
- Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).
- Matthew Cong and Ronald Fedkiw. 2019. Muscle-based facial retargeting with anatomical constraints. In *ACM SIGGRAPH 2019 Talks*. 1–2.
- Matthew Deying Cong. 2016. *Art-directed muscle simulation for high-end facial animation*. Stanford University.
- Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. 2021. Learning an animatable detailed 3D face model from in-the-wild images. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–13.
- Stephan J Garbin, Marek Kowalski, Virginia Estellers, Stanislaw Szymanowicz, Shideh Rezaeifar, Jingjing Shen, Matthew Johnson, and Julien Valentin. 2022. VolTeMorph: Realtime, Controllable and Generalisable Animation of Volumetric Representations. *arXiv preprint arXiv:2208.00949* (2022).
- Thomas Gietzen, Robert Brylka, Jascha Achenbach, Katja Zum Hebel, Elmar Schömer, Mario Botsch, Ulrich Schwanecke, and Ralf Schulze. 2019. A method for automatic forensic facial reconstruction based on dense statistics of soft tissue thickness. *PLoS one* 14, 1 (2019), e0210257.
- Benjamin Gilles, Lionel Reveret, and Dinesh K Pai. 2010. Creating and animating subject-specific anatomical models. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 2340–2351.
- Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. 2022. Neural head avatars from monocular RGB videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18653–18664.
- David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106* (2016).
- Daniel Holden, Bang Chi Duong, Sayantan Datta, and Derek Nowrouzezahrai. 2019. Subspace neural physics: Fast data-driven interactive simulation. In *Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 1–12.
- Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. 2015. Dynamic 3D avatar creation from hand-held video input. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–14.
- Alexandru-Eugen Ichim, Petr Kadleček, Ladislav Kavan, and Mark Pauly. 2017. Phace: Physics-based face modeling and animation. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–14.
- Alexandru Eugen Ichim, Ladislav Kavan, Merlin Nimier-David, and Mark Pauly. 2016. Building and animating user-specific volumetric face rigs. In *Symposium on Computer Animation*. 107–117.
- Petr Kadleček, Alexandru-Eugen Ichim, Tiantian Liu, Jaroslav Krivánek, and Ladislav Kavan. 2016. Reconstructing personalized anatomical models for physics-based body animation. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–13.
- Petr Kadleček and Ladislav Kavan. 2019. Building accurate physics-based face models from data. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2, 2 (2019), 1–16.
- Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4401–4410.
- Marilyn Keller, Silvia Zuffi, Michael J Black, and Sergi Pujades. 2022. OSSO: Obtaining Skeletal Shape from Outside. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20492–20501.
- Martin Komaritzan and Mario Botsch. 2018. Projective skinning. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 1 (2018), 1–19.
- Martin Komaritzan, Stephan Wenninger, and Mario Botsch. 2021. Inside Humans: Creating a Simple Layered Anatomical Model from Human Surface Scans. *Frontiers in Virtual Reality* 2 (2021), 694244.
- Yeara Kozlov, Derek Bradley, Moritz Bächer, Bernhard Thomaszewski, Thabo Beeler, and Markus Gross. 2017. Enriching facial blendshape rigs with physical simulation. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 75–84.
- Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiosek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*. PMLR, 3744–3753.
- John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H Pighin, and Zhigang Deng. 2014. Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)* 1, 8 (2014), 2.
- John P Lewis, Jonathan Mooser, Zhigang Deng, and Ulrich Neumann. 2005. Reducing blendshape interference by selected motion attenuation. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*. 25–29.
- Hao Li, Thibaut Weise, and Mark Pauly. 2010. Example-based facial rigging. *ACM transactions on graphics (tog)* 29, 4 (2010), 1–6.
- Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.
- Nadia Maalin, Sophie Mohamed, Robin SS Kramer, Piers L Cornelissen, Daniel Martin, and Martin J Tovée. 2021. Beyond BMI for self-estimates of body size and shape: A new method for developing stimuli correctly calibrated for body composition. *Behavior Research Methods* 53, 3 (2021), 1308–1321.
- Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.
- Frederic I Parke. 1991. Control parameterization for facial animation. In *Computer Animation '91*. Springer, 3–14.
- Cristian Romero, Dan Casas, Maurizio M Chiamaronte, and Miguel A Otaduy. 2022. Contact-centric deformation learning. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–11.
- Shunsuke Saito, Zi-Ye Zhou, and Ladislav Kavan. 2015. Computational bodybuilding: Anatomically-based modeling of human bodies. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–12.
- Igor Santesteban, Elena Garces, Miguel A Otaduy, and Dan Casas. 2020. SoftSMPL: Data-driven Modeling of Nonlinear Soft-tissue Dynamics for Parametric Humans. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 65–75.
- Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks* 20, 1 (2008), 61–80.
- Robert Schleicher, Marlies Nitschke, Jana Martschinke, Marc Stamminger, Björn M Eskofier, Jochen Klucken, and Anne D Koelwijn. 2021. BASH: Biomechanical Animated Skinned Human for Visualization of Kinematics and Muscle Activity. In *VISIGRAPP (I: GRAPP)*. 25–36.
- Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. In *ACM*

- SIGGRAPH 2005 Papers*. 417–425.
- Steven L Song, Weiqi Shi, and Michael Reed. 2020. Accurate face rig approximation with deep differential subspace reconstruction. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 34–1.
- Jente L Spille, Martin Grunwald, Sven Martin, and Stephanie M Mueller. 2021. Stop touching your face! A systematic review of triggers, characteristics, regulatory functions and neuro-physiology of facial self touch. *Neuroscience & Biobehavioral Reviews* 128 (2021), 102–116.
- Sangeetha Grama Srinivasan, Qisi Wang, Junior Rojas, Gergely Klár, Ladislav Kavan, and Eftychios Sifakis. 2021. Learning active quasistatic physics-based models from data. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–14.
- Robert W Sumner and Jovan Popović. 2004. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)* 23, 3 (2004), 399–405.
- Lingchen Yang, Byungsoo Kim, Gaspard Zoss, Baran Gözcü, Markus Gross, and Barbara Solenthaler. 2022. Implicit neural representation for physics-driven actuated soft bodies. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–10.
- Li Zhang, Noah Snavely, Brian Curless, and Steven M Seitz. 2008. Spacetime faces: High-resolution capture for modeling and animation. In *Data-Driven 3D Facial Animation*. Springer, 248–276.
- Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Bühler, Xu Chen, Michael J Black, and Otmar Hilliges. 2022. Im avatar: Implicit morphable head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13545–13555.
- Wojciech Zielonka, Timo Bolkart, and Justus Thies. 2023. Instant volumetric head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4574–4584.





## Citation

**SparseSoftDECA: Efficient High-Resolution Physics-Based Facial Animation from Sparse Landmarks**

Nicolas Wagner, Ulrich Schwanecke, and Mario Botsch

Computers and Graphics 119, 2024

DOI: 10.1016/j.cag.2024.103903

## 4.1 METHOD SUMMARY

In the publication *SparseSoftDECA*, we responded to Research Question 2 (Section 1.2.2).

*SparseSoftDECA*'s method expands *SoftDECA* (Chapter 3) but focuses on generating personalized animations directly from sparsely observed facial landmarks; effectively decoupling it from traditional blendshape systems. Distinctions between *SparseSoftDECA* and *SoftDECA* mostly become necessary as landmarks constitute a higher-dimensional and personalized input compared to unpersonalized blendshape weights. In particular, adjustments in both the training data generation pipeline and the physics-based simulation (PBS) are required. Nonetheless, the idea of approximating the PBS via an efficient hypernetwork [15] design remains essentially unchanged. As before, we describe the integral components of *SparseSoftDECA* in more detail in the following.

## PHYSICS-BASED SIMULATION

The simulation of *SparseSoftDECA* is similar to that of *SoftDECA*, however, the simulation is no longer used to correct entire *linear blendshapes* (*LBS*) expressions, but to register a neutral head to sparsely tracked facial landmarks in an anatomically and physically plausible manner.

## NETWORK

Also, the hypernetwork architecture of *SparseSoftDECA* is almost identical to the hypernetwork of *SoftDECA*. Simply, the input of the smaller (animation) component is changed to facial landmarks, which has a negligible influence on the rapid inference speed.

## TRAINING DATA

When generating the training data for *SparseSoftDECA*'s hypernetwork, however, we make two substantial adjustments in comparison to *SoftDECA*. First, we enhance the capabilities of the custom *iOS* app of *SoftDECA* to track 150 facial landmarks using the *ARKit* [4] framework. Here, we primarily target key facial contours such as the mouth, the eyes, and the jawline. Second, unlike blendshape weights, the tracked landmarks are specific to the head shapes of the individuals in the recorded conversations and can not be directly applied to *DECA* heads. Consequently, we employ *deformation transfer* [105] to convert the tracked landmarks between different identities. Besides the adaptation of the tracked landmarks to manifold identities, the training data undergoes extensive augmentations like Gaussian noise to improve the robustness of *SparseSoftDECA*.

## 4.2 DISCUSSION

## RESULTS

Given the structural similarities between *SparseSoftDECA* and *SoftDECA*, it is unsurprising that *SparseSoftDECA* effectively addresses its associated research question, as well. Although the learned approximation exhibits a slightly higher inaccuracy than before, the errors stay within the sub-millimeter range. The model also maintains its broad applicability to diverse head shapes and can be run with nearly 100 frames-per-second on consumer-grade CPUs, too. Additionally, other valuable features of *SoftDECA*, such as the control or manipulation of material properties, are preserved. In an ablation study, we could prove that our data augmentations enhance the generalization capabilities of *SparseSoftDECA* to a considerable extent.

## LIMITATIONS

*SparseSoftDECA* carries over limitations of its predecessor. The primary concern is still the animation of heads outside the *DECA* distribution. To investigate such circumstances, we used *SparseSoftDECA* to animate heads with landmarks retrieved from other individuals. Although there was a slight reduction in the approximation quality, our approach remained robust and did not produce inauthentic results.

Another drawback of *SparseSoftDECA* is that it is tied to a specific facial landmark topology, which requires retraining if it is to be changed. Due to the involved renewal of training data, retraining unfortunately takes several days on a high-end workstation.

## RELATED WORK

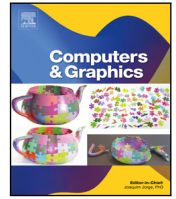
Previous methods for converting sparse facial landmarks into dense facial expressions typically involve the fitting of either 3D morphable models (3DMMs) [118] or *LBS* [56]. To the best of our knowledge, no other approach employs a PBS like *SparseSoftDECA*. Compared to the two classical concepts, our simulation has the advantage that it can be registered flexibly and only adheres to anatomical constraints. 3DMMs or *LBS*, on the other hand, are limited to their original data-driven domain and may not be able to represent details contained by the landmarks. Nonetheless, our approach also holds a significant advantage even when landmarks provide only low-frequency information: as demonstrated in *SoftDECA*, we can authentically add high-frequency features like wrinkles by altering material properties.

## FUTURE WORK

The most intriguing future development we envision for *SparseSoftDECA* is inspired by the current restriction to a predefined landmark topology. Principally, our data generation pipeline already allows us to incorporate various topologies within one training dataset. However, for this feature to be truly beneficial, the hypernetwork [15] must also be able to cope with a varying number of input landmarks. Our current architecture does not support this flexibility and requires a constant number of input dimen-

sions. Neural network architectures like *set transformer* [55] can manage varying input sizes but are renowned for their slow inference speed. An extension that considers the occlusion of landmarks would be as valuable as varying topologies, but would encounter similar challenges. Ultimately, the concept of *SparseSoftDECA* can potentially be enhanced to the extent that the processing of noisy point clouds with random structure becomes feasible. This improvement would be particularly thrilling, as such point clouds can usually be readily generated from 3D multi-view scanners, but the time-consuming tasks of cleaning and standardizing the point clouds would no longer be needed. One starting point for such improvements might be tessellation-agnostic facial rigging [92].

### 4.3 PUBLICATION



Special Section on MIG 2023

# SparseSoftDECA — Efficient high-resolution physics-based facial animation from sparse landmarks

Wagner Nicolas <sup>a,\*</sup>, Schwanecke Ulrich <sup>b</sup>, Botsch Mario <sup>a</sup><sup>a</sup> TU Dortmund University, Otto-Hahn-Str. 16, 44227 Dortmund, Germany<sup>b</sup> University of Applied Sciences RheinMain, Kurt-Schumacher-Ring 18, 65197 Wiesbaden, Germany

## ARTICLE INFO

### Keywords:

Facial animation  
Deep learning  
Physics-based simulation

## ABSTRACT

Facial animation on computationally limited systems still heavily relies on linear blendshape models. Nonetheless, these models exhibit common issues like volume loss, self-collisions, and inaccuracies in soft tissue elasticity. Furthermore, personalizing blendshapes models demands significant effort, but there are limited options for simulating or manipulating physical and anatomical characteristics afterwards. Also, second-order dynamics can only be partially represented.

For many years, physics-based facial simulations have been explored as an alternative to linear blendshapes, however, those remain cumbersome to implement and result in a high computational burden. We present a novel deep learning approach that offers the advantages of physics-based facial animations while being effortless and fast to use on top of linear blendshapes. For this, we design an innovative hypernetwork that efficiently approximates a physics-based facial simulation while generalizing over the extensive DECA model of human identities, facial expressions, and a wide range of material properties that can be locally adjusted without re-training.

In addition to our previous work, we also demonstrate how the hypernetwork can be applied to facial animation from a sparse set of tracked landmarks. Unlike before, we no longer require linear blendshapes as the foundation of our system but directly operate on neutral head representations. This application is also used to complement an existing framework for commodity smartphones that already implements high resolution scanning of neutral faces and expression tracking.

## 1. Introduction

Currently, research in the realm of head avatars and facial animation primarily revolves around achieving photorealistic outcomes using neural networks [1–4]. These approaches require substantial computational resources for operation. However, a significant challenge lies in accommodating less powerful hardware configurations and scenarios where geometry-based processing is necessary. In such cases, various adaptations of linear blendshape models [5] remain the conventional choice for production.

Despite decades of intensive research and refinement of linear facial models, they still exhibit known limitations, including physically implausible distortions, volume loss, anatomically impossible expressions, the absence of volumetric elasticity, and self-intersections. To address these issues, physics-based simulations have been proposed, which mitigate most artifacts associated with linear blendshapes and introduce a range of additional capabilities [6–12]. Researchers have explored applications in fields such as medicine, involving the visual-

ization of weight changes, paralysis, or surgical procedures, as well as visual effects like aging, zombifications, gravity alterations, and second-order effects. Moreover, recent work has demonstrated that simulations incorporating detailed material information result in significantly more realistic facial animations compared to linear models [10].

However, it is important to note that physics-based facial animation models typically impose a substantial computational burden, leading to a considerable body of literature dedicated to acceleration techniques. Much of this research has focused on evaluating simulations within manually constructed subspaces [13] or learned subspaces [14,15] and corrective blendshapes [7]. Among these approaches, learned subspace methods have proven to be more versatile and adaptable [14], which is why they have already found successful application in full-body animations [15]. Nevertheless, there is currently no method that effectively extends these advancements in fast physics-based simulations to facial animations. The principal contribution of this work is closing this gap with a deep learning approach which we call SoftDECA.

\* Corresponding author.

E-mail address: [nicolas.wagner@tu-dortmund.de](mailto:nicolas.wagner@tu-dortmund.de) (W. Nicolas).

SoftDECA introduces an innovative neural network designed to animate facial expressions while closely adhering to a dynamic physics-based model. Our approach possesses universal applicability, as it can accommodate a wide range of physics-based facial animations. However, our specific emphasis lies in approximating a combination of cutting-edge anatomically plausible and volumetric finite element methods (FEM) [6–8,16]. For this, we propose a novel adaption of hypernetworks [17] which yields inference times of about 10 ms on consumer-grade CPUs and has the same programming interface as standard linear blendshapes. More precisely, we train SoftDECA to be applied as an add-on to arbitrary human blendshape rigs that follow the Apple ARKit system.

Furthermore, SoftDECA offers straightforward deployment without the necessity for intricate customizations or retraining efforts due to our extensive compilation of training examples. This comprehensive dataset encompasses a substantial domain of the intended FEM model and amalgamates data from various sources. These sources include CT head scans to capture head anatomy, 3D head reconstructions representing diverse head shapes (utilizing DECA as outlined in [18]), and facial expressions recorded as ARKit blendshape weights from dyadic conversational scenarios. The resulting training dataset ensures SoftDECA's capacity for robust generalization across a spectrum of human identities, facial expressions, and the extensive parameter space of the targeted FEM model. In contrast to earlier methods [14,15], the ability to generalize across simulation parameters makes extensive and efficient artistic interventions possible, with SoftDECA even supporting localized material adjustments.

As an additional contribution, we present a novel layered head model (LHM) that represents all training instances in a standardized way. Unlike fully or partially tetrahedralized volumetric meshes conventionally used for FEM, the LHM has additional enveloping wraps around bones, muscles, and skin. Based on these wraps, we describe a data-driven fitting procedure that positions muscles and bones within a neutral head while avoiding intersections of the various anatomic structures. A characteristic that was mostly not of concern in previous manually crafted physics-based facial animations but can otherwise lead to numerical instabilities in our automated training data generation approach.

This paper is an extension to the previously presented SoftDECA [19]. Here, we additionally introduce the adapted SparseSoftDECA, which maps sparsely observed facial landmarks into plausible facial expressions with respect to the foundational physics-based simulation. Again, SparseSoftDECA is trained to exhibit a high degree of generalization, accommodating a variety of head shapes and landmark positions. As before, we present a pipeline for generating extensive training data that densely samples the input domains.

The animation via facial landmarks offers the advantage of eliminating the need for blendshape generation entirely. All that is required for animating a person's face is SparseSoftDECA and the neutral head shape which can be easily obtained. For instance, Wenniger et al. [20] have demonstrated the quick acquisition of a neutral head shape in just a few minutes solely based on smartphone videos.

Furthermore, SparseSoftDECA inherently supports personalized animations when facial landmarks can be reliably tracked. Achieving this level of personalization, such as through linear blendshapes, typically demands several of additional scans for each individual.

## 2. Related work

### 2.1. Personalized anatomical models

Algorithms for generating personalized anatomical models can be categorized into two main paradigms: *heuristic-based* and *data-driven*. In the realm of heuristic-based approaches, Anatomy Transfer [21] employs a space warp on a template anatomical structure to conform to a target skin surface, deforming the skull and other bones only through

an affine transformation. A similar approach is presented by Gilles et al. [22], incorporating statistical validation of bone shapes derived from artificially deformed bones. In both [7,23], an inverse physics-based simulation is utilized to reconstruct anatomical structures from multiple 3D expression scans. Saito et al. [24] focus on simulating the growth of soft tissue, muscles, and bones. In [25], a complete musculoskeletal biomechanical model is fitted based on sparse observations, however, no qualitative evaluation is conducted.

Primarily, concerns such as data privacy or potential radiation exposure keep the number of data-driven anatomy fitting approaches small. The recent OSSO method [26,27] predicts body skeletons from 2000 DXA images. These images do not contain precise 3D information and bones are placed within the body by predicting solely three anchor points per bone group. Additionally, intersections between skin and bones are not resolved. In [28], skin-bones intersections are addressed and also the musculature is fitted. Instead of fitting anatomical structures directly, encapsulating wraps are placed within a body. However, this approach relies on a BMI regressor rather than accurate medical imaging [29]. Also in [27], skeletons do not intersect but are not placed based on medical imaging either.

A more accurate facial model, developed by Achenbach et al. [30], combines CT scans with optical surface scans using a multilinear model (MLM) that maps between skulls and faces bidirectionally. Despite its accuracy, this model does not prevent self-intersections and solely focuses on fitting bones. Building upon the data from [30] and extending the concept of a layered body model [28], we formulate a statistical layered head model encompassing musculature while mitigating self-intersections.

### 2.2. Physics-based facial animation

Various paradigms for animating faces have been developed in the past [31–34]. Dominating the field are data-driven models [5,7,35], which have witnessed significant advancements with the application of deep learning techniques [1,3,18,36–38]. Linear blendshapes [5] remain prevalent in demanding applications and scenarios lacking computationally rich hardware due to their simplicity and speed. Physics-based simulations, although addressing issues of blendshape models like implausible contortions and self-intersections, are less commonly used due to their inherent complexity and computational demands. Sifakis et al.'s [39] pioneering work represents the first fully physics-based volumetric facial animation, employing a personalized tetrahedron mesh with limited resolution due to an involved dense optimization problem. The Phace system [6] successfully overcame this limitation through an improved simulation. Art-directed physics-based facial animations additionally employ a muscle representation based on B-splines [8,16,40]. Animations can then be controlled via trajectories of spline control points. A solely inverse model for determining physical properties of faces is presented in [41].

Hybrid methodologies incorporate surface-based physics into linear blendshapes to enhance the intricacy of facial expressions [9,11,42,43]. Nevertheless, due to their design, these approaches are unable to represent volumetric effects. The introduction of volumetric blendshapes [7] represents a hybrid solution that amalgamates the structure of linear blendshapes with volumetric physical and anatomical plausibility. However, achieving real-time performance necessitates the utilization of extensive personalized corrective blendshapes.

Considering soft bodies in general, deep learning approaches have been investigated to approximate physics-based simulations. For instance, in [15,44] the SMPL (Skinned Multi-Person Linear Model) proposed in [45] was extended with secondary motion. Recently, [9, 10,12] developed methods to learn the particular physical properties of objects and faces. However, these approaches must be retrained for unseen identities and are slow in inference. A fast and general approach for learning physics-based simulations is introduced in [14]. Unfortunately, they focused on reflecting the dynamics of single objects



Fig. 1. All components of the layered head model template  $\mathcal{T}$ . Skin  $S_{\mathcal{T}}$ , skin wrap  $\hat{S}_{\mathcal{T}}$ , muscles  $M_{\mathcal{T}}$ , muscles wrap  $\hat{M}_{\mathcal{T}}$ , skull  $B_{\mathcal{T}}$ , and the skull wrap  $\hat{B}_{\mathcal{T}}$ .

with limited complexity. We present a real-time capable deep learning approach to physics-based facial animations that does not need to be retrained and maintains the control structure of standard linear blendshapes. Additionally, none of the previously described deep learning methods tackle the challenging creation of facial training data, which we also address in this work.

### 3. Method

The cornerstone of the SoftDECA animation system lies in a novel layered head representation (Section 3.1). Building upon this foundation, we formulate a physics-based facial animation system (Sections 3.2 & 3.3) and illustrate how to distill it into a defining dataset (Section 3.4). Utilizing this dataset, we train a newly devised hypernetwork (Section 3.5) capable of real-time approximation of the animation system. In addition to our previous work [19], we enhance SoftDECA to be directly addressable by sparse landmarks, rendering it entirely independent of linear blendshapes if desired (Section 3.6).

#### 3.1. Layered head model

##### 3.1.1. Structure

We define a head  $\mathcal{H} = \rho_{\mathcal{H}}(\mathcal{T})$  with a neutral expression through a component-wise transformation  $\rho_{\mathcal{H}}$  of a layered head model template

$$\mathcal{T} = (S_{\mathcal{T}}, B_{\mathcal{T}}, M_{\mathcal{T}}, \hat{S}_{\mathcal{T}}, \hat{B}_{\mathcal{T}}, \hat{M}_{\mathcal{T}}), \quad (1)$$

comprising six triangle meshes.  $S_{\mathcal{T}}$  delineates the skin surface, encompassing the eyes, mouth cavity, and tongue.  $B_{\mathcal{T}}$  denotes the surface of all skull bones including the teeth.  $M_{\mathcal{T}}$  represents the surface of all muscles, along with the cartilages of the ears and nose.  $\hat{S}_{\mathcal{T}}$  is the skin wrap, i.e. a closed wrap that envelopes  $S_{\mathcal{T}}$ .  $\hat{B}_{\mathcal{T}}$  is the skull wrap that encloses  $B_{\mathcal{T}}$  and  $\hat{M}_{\mathcal{T}}$  is the muscle wrap that encloses  $M_{\mathcal{T}}$ . For simplicity, other anatomical structures are omitted. The template structures  $S_{\mathcal{T}}$ ,  $B_{\mathcal{T}}$ , and  $M_{\mathcal{T}}$  were artistically designed, while the skin, skull, and muscle wraps  $\hat{S}_{\mathcal{T}}$ ,  $\hat{B}_{\mathcal{T}}$ , and  $\hat{M}_{\mathcal{T}}$  were generated by shrink-wrapping the same sphere as closely as possible to the corresponding surfaces without intersections. The complete template is depicted in Fig. 1.

The shared triangulation among the wraps of the LHM allows to also define a soft tissue tetrahedron mesh  $\mathbb{S}_{\mathcal{T}}$  (between the skin and muscle wraps) and a muscle tissue tetrahedron mesh  $\mathbb{M}_{\mathcal{T}}$  (between the muscle and skull wraps). For this purpose, each triangular prism spanned between corresponding wrap faces is canonically split into three tetrahedra. The complexities of all template components are detailed in the appendix. In the subsequent sections, we denote the number of vertices in a mesh as  $|\cdot|_v$  and the number of faces as  $|\cdot|_f$ .

##### 3.1.2. Fitting

Later on, generating training data involves determining

$$(S, B, M, \hat{S}, \hat{B}, \hat{M}) = \rho_{\mathcal{H}}(\mathcal{T}) \quad (2)$$

when only the skin surface  $S$  of the head  $\mathcal{H}$  is known. For this purpose, we employ a hybrid approach that places the skull wrap in a data-driven manner, while the remaining template components are fitted using heuristics to ensure anatomical plausibility and avoid self-intersections.

Starting with the fitting of the skin wrap, we set

$$\hat{S} = \text{rbf}_{S_{\mathcal{T}} \rightarrow S}(\hat{S}_{\mathcal{T}}). \quad (3)$$

Here, the RBF function denotes a space warp based on triharmonic radial basis functions [46], calculated from the template skin surface  $S_{\mathcal{T}}$  to the target  $S$  and applied to the template skin wrap  $\hat{S}_{\mathcal{T}}$ . Due to the construction of RBFs, the skin wrap undergoes a semantically consistent warp, adhering closely to the targeted skin surface.

Following, we fit the skull wrap  $\hat{B}$  by first evaluating a linear regressor  $D$  that predicts distances from the vertices of  $\hat{S}$  to the corresponding vertices of  $\hat{B}$ . Then, we minimize with projective dynamics [47]

$$\arg \min_X w_{\text{rect}} E_{\text{rect}}(X, \hat{S}_{\mathcal{T}}) + w_{\text{dist}_2} E_{\text{dist}_2}(X, \hat{S}, D(\hat{S})) + w_{\text{curv}} E_{\text{curv}}(X, \hat{B}_{\mathcal{T}}). \quad (4)$$

In this optimization,  $E_{\text{dist}_2}$  ensures the adherence to the predicted distances,  $E_{\text{curv}}$  represents a curvature regularization for the skull wrap, and  $E_{\text{rect}}$  prevents shearing between corresponding faces of the skin and skull wraps. The distances are set to a minimum value if they fall below a threshold, thereby preventing skin–skull intersections. For formal descriptions of the energy components, please refer to the appendix. The optimization is initialized with  $X = \hat{S} - D(S) \cdot n(\hat{S})$ , where  $n(\hat{S})$  denotes area-weighted vertex normals. The linear regressor  $D$  is trained on the dataset from [48] (SKULLS), which correlates CT skull measurements with optical skin surface scans. For a visual illustration of the training process of the linear regressor please refer to Wagner et al. [19].

The muscle wrap  $\hat{M}$  is placed almost at the same absolute distances between corresponding vertices of the skin and skull wraps as in the template. Only ten percent of the relative distance changes compared to the template are incorporated, assuming that the muscle mass in the facial area is only moderately influenced by body weight and skull size.

The skull mesh is placed by setting

$$B = \text{rbf}_{\hat{B}_{\mathcal{T}} \rightarrow \hat{B}}(B_{\mathcal{T}}). \quad (5)$$

The characteristics of the RBF space warp ensure that the skull mesh remains enclosed within the skull wrap, provided the wrap has sufficient resolution. While the muscle mesh could be positioned similarly, it is not utilized further in our pipeline.

Finally, the tetrahedron meshes representing soft and muscle tissue  $\mathbb{S}$  and  $\mathbb{M}$  are constructed as described before. On average, the complete fitting pipeline takes about 500 ms on an AMD Threadripper Pro 3995wx processor. Fig. 2 visualizes the overall fitting process.

#### 3.2. SoftDECA animation system

Building upon the LHM representation, we now introduce the SoftDECA animation system by, first, revisiting the concept of linear blendshapes. Subsequently, we derive the dynamic physics-based facial simulation system, which forms the core of SoftDECA.

In a linear blendshape model,  $n$  surface blendshapes

$$\{S^i\}_{i=1}^n \quad (6)$$

animate a facial expression  $S_i$  as a linear combination

$$S_i = \sum_{i=1}^n w_i^i S^i, \quad (7)$$

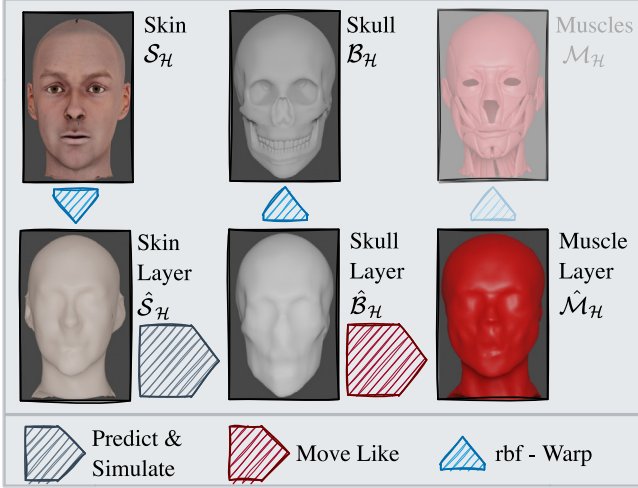


Fig. 2. (a) Procedural overview of the layered head model fitting algorithm.

where the blending weights  $w_i$  determine the contribution of each blendshape to the expression at frame  $t$ .

To achieve the same animation with a physics-based model  $\phi$ , one typically employs either forward or inverse simulations. Without loss of generality, we consider inverse simulations in the following. Here, the expression  $S_t$  is converted into the (in the Euclidean sense) closest  $\phi$ -plausible solution by  $\phi^\dagger$  to

$$T_t = \phi^\dagger(S_t, \mathbf{p}), \quad (8)$$

where  $\mathbf{p}$  is a vector of material and simulation parameters on which  $\phi$  depends. For including second-order effects as well, Eq. (8) expands to

$$T_t = \phi^\dagger(\gamma S_t + 2\alpha T_{t-1} - \beta T_{t-2}, \mathbf{p}). \quad (9)$$

The SoftDECA animation system operates in a similar manner, but the right-hand side of Eq. (9) is approximated by a computationally efficient neural network  $f$ .

Ensuing, we will elucidate our implementation of  $\phi^\dagger$  and the process of generating representative examples. However, please note that SoftDECA is not confined to a specific implementation of  $\phi^\dagger$ .

### 3.3. Physics-based simulations

We implement anatomically plausible inverse physics  $\phi^\dagger$  as a projective dynamics energy  $E_{\phi^\dagger}$ . At this, state-of-the-art FEM models [6,8,41] are merged by applying separate terms for soft tissue, muscle tissue, the skin, the skull, and auxiliary components.

#### 3.3.1. Energy

Considering the soft tissue  $\mathbb{S}$ , we closely follow the model of [6] and impose

$$E_{\mathbb{S}} = w_{\text{vol}} \sum_{t \in \mathbb{S}} E_{\text{vol}}(t) + w_{\text{str}} \sum_{t \in \mathbb{S}} \mathbb{1}_{\sigma_{F(t)} > \epsilon} E_{\text{str}}(t), \quad (10)$$

which for each tetrahedron  $t$  penalizes change of volume and strain, respectively. Strain is only accounted for if the largest eigenvalue  $\sigma_{F(t)}$  of the stretching component of the deformation gradient  $F(t) \in \mathbb{R}^{3 \times 3}$  grows beyond  $\epsilon$ .

To reflect the biological structure of the skin, we additionally formulate a dedicated strain energy

$$E_S = \sum_{t \in \mathbb{S}} E_{\text{str}}(t) \quad (11)$$

on each triangle  $t$  of the skin which, to the best of our knowledge, has not been done before.

For the muscle tetrahedra  $\mathbb{M}$ , we follow Kadleček et al. [41] that capturing fiber directions for tetrahedralized muscles is in general too restrictive. Hence, only a volume-preservation term

$$E_{\mathbb{M}} = w_{\text{vol}} \sum_{t \in \mathbb{M}} E_{\text{vol}}(t) \quad (12)$$

is applied for each tetrahedron in  $\mathbb{M}$ .

The skull is not tetrahedralized as it is assumed to be non-deformable even though it is rigidly movable. The non-deformability of the skull is represented by

$$E_B = \sum_{t \in B} E_{\text{str}}(t) + \sum_{x \in B} E_{\text{curv}}(x, B), \quad (13)$$

i.e. a strain  $E_{\text{str}}$  on the triangles  $t$  and mean curvature regularization on the vertices  $x$  of the skull  $B$ . We do not model the non-deformability as a rigidity constraint due to the significantly higher computational burden.

To connect the muscle tetrahedra as well as the eyes to the skull, connecting tetrahedra are introduced similar to the sliding constraints in [6]. For the muscle tetrahedra, each skull vertex connects to the closest three vertices in  $\mathbb{M}$  to form a connecting tet. For the eyes, connecting tetrahedra are formed by connecting each eye vertex to the three closest vertices in  $B$ . On these connecting tetrahedra, the energy  $E_{\text{con}}$  with the same constraints as in Eq. (10) is imposed. By this design, the jaw and the cranium are moved independently from each other through muscle activations but the eyes remain rigid and move only with the cranium.

Finally, the energy

$$E_{\text{inv}} = \sum_{x \in S} E_{\text{tar}}(x, S_t) \quad (14)$$

of soft Dirichlet constraints is added, attracting the skin surface  $S$  vertices to the targeted expression  $S_t$ .

The weighted sum of the aforementioned energies gives the total energy

$$E_{\phi^\dagger} = w_{\mathbb{S}} E_{\mathbb{S}} + w_{\mathbb{M}} E_{\mathbb{M}} + w_B E_B + w_{\text{mstr}} E_{\text{mstr}} + w_S E_S + w_{\text{con}} E_{\text{con}} + w_{\text{inv}} E_{\text{inv}} \quad (15)$$

of the inverse model  $\phi^\dagger$ . Altogether,  $\phi^\dagger$  results in an expression  $T_t$  that in a Euclidean sense is close to the target  $S_t$ , but is plausible w.r.t. the imposed constraints.

#### 3.3.2. Collisions

Finally, self-intersections are resolved between colliding lips or teeth in a subsequent projective dynamics update as in [49]. The decisive characteristic of this approach is that no gaps can occur after the resolution of self intersections. For example, in the case of a lip collision, the corresponding lower and upper lip points are simulated to the same position.

#### 3.3.3. Parameters

The construction of  $\phi^\dagger$  also implies parts of the parameter vector  $\mathbf{p}$ . As such, the dynamics parameters  $\alpha, \beta, \gamma$ , weights  $w_*$  of all the constraints, but also other attributes of the constraints are considered. For example, the target volume in  $E_{\text{vol}}$  or scaling factors of the skull bones are included. We also add constant external forces like gravity strength and direction into  $\mathbf{p}$ . An overview of all parameters we use and the corresponding value ranges is given in the appendix.

### 3.4. Training data

According to the definition of the animation system in Eq. (9), a comprehensive training dataset  $\mathcal{D}$  should include examples that link various facial expressions generated through linear blendshapes to the corresponding surfaces conforming to  $\phi$ . Moreover, to encompass dynamic effects, the exemplary facial expressions should form coherent



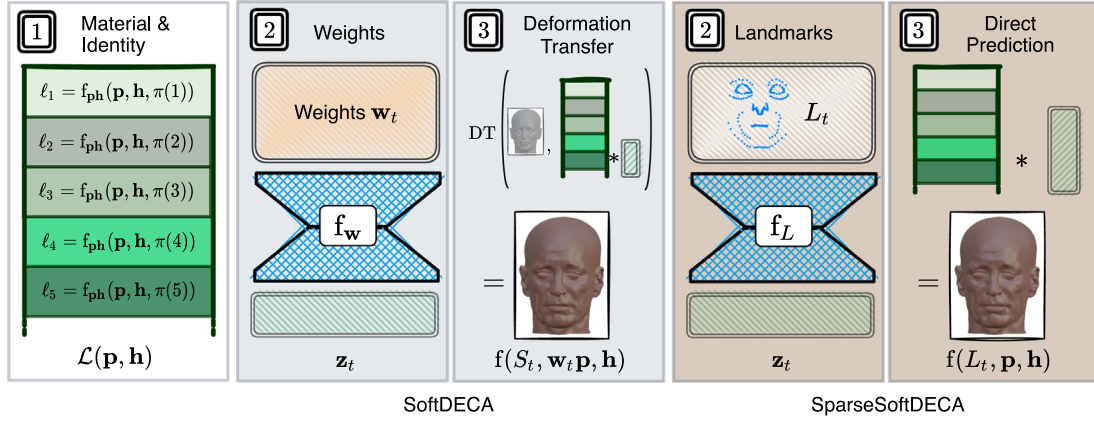


Fig. 3. An overview of SoftDECA and SparseSoftDECA facial animations. In Step (1), for both, the hyper-tensor and the dynamic parameters are determined once for an animation. Subsequently, steps 2–3 are repeatedly evaluated per frame and either map blendshapes weights to deformation gradients (SoftDECA) or landmarks to vertex position (SparseSoftDECA).

sequences. This dataset also needs to encompass a range of diverse head shapes and simulation parameters.

In the following, we describe a pipeline for creating instances of such a dataset, which can be roughly divided into six high-level steps.

1. We commence by randomly selecting a neutral skin surface  $S$  from DECA [18], an extensive high-resolution face model. Specifically, we pick an image at random from the Flickr-Faces-HQ [50] dataset and let DECA determine the corresponding neutral head shape along with a latent representation  $\mathbf{h}$ .
2. The template LHM  $\mathcal{T}$  is aligned with the skin surface  $S$  as described in Section 3.1.
3. Deformation transfer [51] is applied to map ARKit surface-based blendshapes to  $S$ .
4. An expression sequence  $\mathbf{S} = (S_i)_{i=0}^m$  of length  $m + 1$  is generated by applying a sequence of linear blendshape weights  $\mathbf{w} = (\mathbf{w}_i)_{i=0}^m$ . These blendshape weights are derived from 8 approximately 10 min long dyadic conversations recorded using a custom iOS app.
5. As the final step before generating the  $\phi$ -plausible counterpart of  $\mathbf{S}$ , it is necessary to sample simulation parameters within appropriate domains. We expect the user to specify lower and upper bounds for continuous parameter beforehand. Then, for each continuous entry in  $\mathbf{p}$ , a value is independently sampled from a uniform distribution between the specified bounds. Discrete parameters are treated similarly, without specific constraints.
6. Finally,  $\mathbf{T} = (\phi^\dagger(S_i, \mathbf{p}))_{i=0}^m$  is computed and  $(\mathbf{T}, \mathbf{S}, \mathbf{w}, \mathbf{p}, \mathbf{h})$  is appended to  $D$ . Evaluating one time step takes approximately 10 s on an AMD Threadripper Pro 3995wx.

### 3.5. Hypernetwork

#### 3.5.1. Architecture & training

Having training data, we can now design a computationally efficient neural network  $f$  to approximate the physics-based simulation from Eq. (9). Irrespective of a particular architecture, the training goal implied by  $D$  is to optimize on each frame

$$\min_f \sum_{(\mathbf{T}, \mathbf{S}, \mathbf{w}, \mathbf{p}, \mathbf{h}) \in D} \sum_{i=0}^m \|T_i - f(S_i, \mathbf{w}_i, \mathbf{p}, \mathbf{h})\|_2. \quad (16)$$

In words,  $f$  is trained to approximate the  $\phi$ -conformal expressions from the linearly blended expressions  $S_i$ , the blending weights  $\mathbf{w}_i$ , simulation parameters  $\mathbf{p}$ , and the head descriptions  $\mathbf{h}$ . Hence, leaving out dynamic effects to begin with, the probably most naive approach would be to learn  $f$  to directly predict vertex positions. However, this would not allow the usage of personalized blendshapes at inference time that have

not been used in the curation of  $D$ . Therefore, we separate  $f$  into two high-level components

$$f(S_i, \mathbf{w}_i, \mathbf{p}, \mathbf{h}) = \text{DT}(S_i, f_{DG}(\mathbf{w}_i, \mathbf{p}, \mathbf{h})), \quad (17)$$

where  $\text{DT}$  is a deformation transfer function as in [52] that applies  $3 \times 3$  per-face deformation gradients (DGs) predicted by  $f_{DG}(\mathbf{w}_i, \mathbf{p}, \mathbf{h}) \in \mathbb{R}^{|S_i| \times 9}$  to the linearly blended  $S_i$ . By doing so,  $f$  can also be applied to a facial expression  $S_i$  which has been formed by unseen personalized blendshapes while still achieving close approximations of  $\phi^\dagger$ . Fortunately, the evaluation of  $\text{DT}$  is not more than efficiently finding a solution to a pre-factorized linear equation system.

To implement the DG prediction network  $f_{DG}$ , we evaluated multiple network architectures such as set transformers [53], convolutional networks on geometry images, graph neural networks [54], or implicit architectures [55], but all have exhibited substantially slower inference speeds while reaching a similar accuracy as a multi-layer perceptron (MLP). Nevertheless, a plain MLP does not discriminate between inputs that change per frame  $t$  and inputs that have to be computed only once. Therefore, we propose an adaptation of a hypernetwork MLP [17] to implement  $f_{DG}$  in which the conditioning of  $f_{DG}$  with respect to the simulation parameters as well as the DECA identity is done by manipulating network parameters. Formally, we implement

$$f_{DG}(\mathbf{w}_i, \mathbf{p}, \mathbf{h}) = \mathbf{z}_i \mathcal{L}(\mathbf{p}, \mathbf{h}), \quad (18)$$

where  $\mathcal{L}(\mathbf{p}, \mathbf{h}) \in \mathbb{R}^{32 \times |S_i| \times 9}$  returns a tensor that only has to be calculated once for all frames and  $\mathbf{z}_i = f_w(\mathbf{w}_i) \in \mathbb{R}^{32}$  is the result of a small standard MLP that processes the blending weights at every frame  $t$ . Each matrix  $\ell_i \in \mathbb{R}^{32 \times 9}$  in  $\mathcal{L}(\mathbf{p}, \mathbf{h})$  corresponds to a face in  $S$  and the entries are calculated as

$$\ell_i = f_{\text{ph}}(\mathbf{p}, \mathbf{h}, \pi(i)). \quad (19)$$

Again,  $f_{\text{ph}}$  is a small MLP and  $\pi$  is a trainable positional encoding. Please consult the appendix for detailed dimensions of all networks and see Fig. 3 for a structural overview of  $f$ .

#### 3.5.2. Localization

The architecture described above offers extensive possibilities for artistic user interventions at inference time. For instance, different simulation parameters  $\mathbf{p}$ , can be used per triangle  $i$  by changing Eq. (19) to

$$\ell_i = f_{\text{ph}}(\mathbf{p}_i, \mathbf{h}, \pi(i)), \quad (20)$$

which enables a localized application of different material models. The  $\text{DT}$  function ensures that the models are smoothly combined.

### 3.5.3. Dynamics

Given that locally differing simulation parameters are not reflected in the training data, existing approaches to integrate dynamics in deep learning [14,15], cannot be adopted. Therefore, we again use the hypernetwork concept to achieve a piecewise-linear dynamics approximation. More precisely, we recursively extend  $f$  to

$$\begin{aligned} f(S_t, \mathbf{w}_t, \mathbf{p}, \mathbf{h}) &= \gamma \odot \text{DT}(S_t, f_{DG}(\mathbf{w}_t, \mathbf{p}, \mathbf{h})) \\ &+ 2\alpha \odot f(S_{t-1}, \mathbf{w}_{t-1}, \mathbf{p}, \mathbf{h}) \\ &- \beta \odot f(S_{t-2}, \mathbf{w}_{t-2}, \mathbf{p}, \mathbf{h}), \end{aligned} \quad (21)$$

where  $\alpha, \beta, \gamma \in \mathbb{R}^{32 \times |S|_v}$  contain per-vertex dynamics parameters. The first row of Eq. (21) is the same as in Eq. (17) but the second and third rows allow for dependencies on the previous two frames. Each entry of  $\alpha, \beta, \gamma$  is calculated as in Eq. (20) but with dedicated MLPs  $f_\alpha, f_\beta, f_\gamma$ . As a result,  $\alpha, \beta, \gamma$  are again not time-dependent and only have to be calculated once.

### 3.6. Sparse animation control

Previously, we assumed that SoftDECA is supposed to map an expression  $S_t$  generated by linear blendshapes (Eq. (7)) into a  $\phi^\dagger$ -plausible expression  $T_t$  (Eq. (8)). In the following, we now assume that only temporally consistent landmarks  $L_t \in S_t$  can be observed per frame  $t$ . At the same time, we no longer require  $S_t$  to be derived from a specific linear blendshape system for inference. We refer to the *adapted* variant which processes landmarks instead of blendshape weights as SparseSoftDECA. In other words, SparseSoftDECA can create personalized animations from tracked landmarks requiring only a neutral scan as input. In this section, we first explain the adaptation of the physics model to the sparse input. Subsequently, which training data is required for SparseSoftDECA is discussed. Finally, we described changes in the hypernetwork topology of SoftDECA to allow landmarks to be used as input.

#### 3.6.1. Adapted physics-based simulation

The foundation of SparseSoftDECA is a modified physics-based model  $\varphi^\dagger$  which in principle optimizes the same energy as  $\phi^\dagger$ . However, the targeted landmarks are enforced by simultaneously optimizing for

$$E_{\text{lmk}} = \sum_{x \in L} E_{\text{tar}}(x, L_t). \quad (22)$$

In our experiments, it has proven beneficial to keep the previous target energy  $E_{\text{inv}}$  as a regularization term. Otherwise, since  $L_t$  is usually only a sparse observation of  $S_t$ , i.e.  $|L|_v \ll |S|_v$ , solely non-uniformly distributed actuation signals would act in  $\varphi^\dagger$  which would cause distortions.

In summary,  $\varphi^\dagger$  is composed by the overall energy

$$\begin{aligned} E_{\varphi^\dagger} &= w_S E_S + w_M E_M + w_B E_B + w_{\text{mstr}} E_{\text{mstr}} \\ &+ w_S E_S + w_{\text{con}} E_{\text{con}} \\ &+ w_{\text{reg}} E_{\text{inv}} + w_{\text{lmk}} E_{\text{lmk}}, \end{aligned} \quad (23)$$

where  $w_{\text{reg}}$  indicates the strength of the regularization and is included in the parameter vector  $\mathbf{p}$ .

#### 3.6.2. Adapted training data

To generate training data for SparseSoftDECA we, basically follow the same data generation pipeline as described in Section 3.4. Merely the steps 4 and 6 must be adjusted to produce training instances with landmarks rather than blendshape weights.

Concerning step 4, we have extended the custom iOS app such that not only weight vector  $\mathbf{w}$ , but also about 150 corresponding landmarks  $L_t$  are captured by Apple's ARKit. These landmarks mainly represent the contours of a face and are visualized in Fig. 4. Contrary to the blendshape weights, the captured landmarks are tailored to the recorded head.

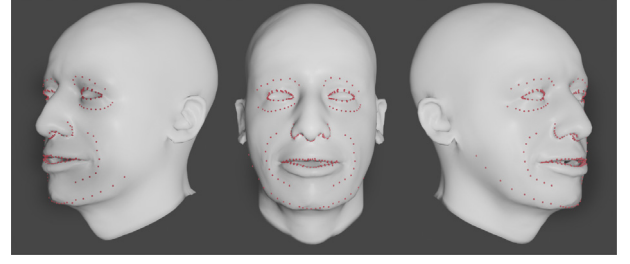


Fig. 4. The set of landmarks used for SparseSoftDECA.

Concerning step 6, a training instance is now formed as  $(\mathbf{T}, \mathbf{S}, \mathbf{L}, \mathbf{p}, \mathbf{h})$  where

$$\begin{aligned} \mathbf{L} &= (\sigma(L_t))_{t=0}^m, \\ \mathbf{T} &= (\varphi^\dagger(\sigma(L_t), S_t, \mathbf{p}))_{t=0}^m. \end{aligned} \quad (24)$$

Here,  $\sigma$  is an augmentation function which serves two purposes. On the one hand, the landmarks must be personalized to account for the difference between the recorded and simulated head shape  $S$  drawn in Step 1 of the data generation pipeline. On the other hand, the notably larger domain as opposed to the blendshape weights requires a denser sampling in the training set, as we will show empirically in Section 4.3. Therefore,  $\sigma$  is composed of a deformation transfer [52] that accomplishes the personalization followed by a coordinate-wise Gaussian noise to achieve a robust domain coverage.

#### 3.6.3. Adapted hypernetwork

For SparseSoftDECA, the efficient hypernetwork topology presented earlier for SoftDECA (Section 3.5) is fundamentally preserved. However, so far, SoftDECA focused on deforming a linear blended surface according to specified material properties. Since SparseSoftDECA is intended to reconstruct a facial expressions without being tied to a particular linear blendshape system, neither the linear blended surface  $S_t$  nor the blendshape weights  $w_t$  can be utilized as input for the adapted hypernetwork. For the same reason, mesh coordinates can be predicted directly without the intermediate step of forming and resolving deformation gradients. Formally, the static hypernetwork  $f$  of SparseSoftDECA is implemented as

$$f(L_t, \mathbf{p}, \mathbf{h}) = f_L(L_t) \mathcal{L}(\mathbf{p}, \mathbf{h}), \quad (25)$$

where  $\mathcal{L}(\mathbf{p}, \mathbf{h}) \in \mathbb{R}^{32 \times |S|_v \times 3}$  returns a tensor that only has to be calculated once for all frames and  $f_L(L_t) \in \mathbb{R}^{32}$  is the result of a small standard MLP that processes the landmarks at every frame  $t$ . The dynamic variant is derived as before in Eq. (21). A structural overview is given in Fig. 3.

### 3.7. Personalized animation from commodity smartphones

We will release SparseSoftDECA trained on the skin topology used in Wenninger et al. [20]. In their work, they demonstrate how to quickly create high-resolution (face) avatars from a single smartphone video. Combining both the high resolution avatars and our models allows for computationally efficient realistic facial animation with real-time tracking even on low budget hardware. Due to the compatibility with ARKit and software based thereon, SoftDECA and SparseSoftDECA can readily be deployed in environments from Apple, Unity, and many more.

## 4. Experiments

Prior to outlining the accuracy and efficiency of SoftDECA (Section 4.2), we first evaluate the precision of the LHM fitting (Section 4.1). Afterwards, we examine both quantitatively and qualitatively SparseSoftDECA (Section 4.3).

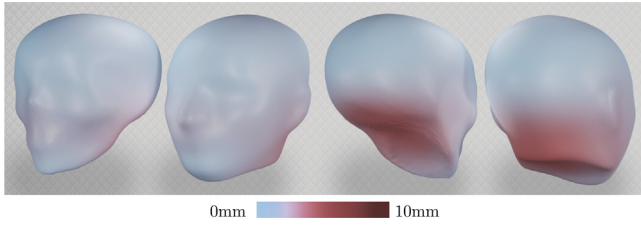


Fig. 5. The per-vertex mean L2-error of the LHM fitting.



Fig. 6. Exemplary fits of the LHM components skull wrap, muscle wrap, and skull.

#### 4.1. LHM fitting

The fitting process of the LHM involves the data-driven positioning of the skull wrap and the subsequent heuristic fitting of the muscle wrap. Our evaluation focuses on the critical fitting of the skull wrap using the CT SKULLS dataset from [48], consisting of 43 instances. To assess precision, a leave-one-out validation is conducted, measuring vertex-wise L2 errors. Prior methods positioning the skull within the head primarily rely on sparse soft tissue statistics derived from a few points on the skull [7,56]. We evaluate our approach against the multilinear model (MLM) proposed by Achenbach et al. [30,48] which demonstrated more robust and precise positioning through the capture of dense soft tissue statistics represented as radii of spheres surrounding the skull.

Both models fall short of achieving medical-grade positioning, exhibiting errors ranging between approximately 2 mm and 4 mm. The MLM demonstrates higher precision with a mean error of 1.98 mm, surpassing our approach, which positions the skull with an average error of 3.83 mm. However, the MLM lacks collision prevention, posing a potential issue for physics-based simulations. Moreover, our fitting algorithm produces significant errors primarily in regions of lesser importance for facial simulations, as depicted in Fig. 5. Notably, errors are concentrated in the back area of the skull, where the rectangular constraints of our fitting procedure may no longer align well with the skin wrap. Fig. 6 provides visual examples of the fitting process.

#### 4.2. SoftDECA

##### 4.2.1. Dataset & training

To train and evaluate  $f$ , we construct a dataset comprising 500k training and test instances using the pipeline detailed in Section 3.4. The parallelized creation of the dataset spanned five days and necessitated one terabyte of storage. To address the disparate sizes of the parameter spaces, 75% of the generated data consists of static

instances where all parameters except the dynamic ones  $\alpha, \beta, \gamma$  are sampled. The remaining 25% of the data is dynamically simulated, resulting in the generation of 6250 dynamic sequences, each with a length of 16 frames. To initiate dynamic sequences with a reasonable velocity, a longer sequence of length 2048 is pre-simulated with fixed dynamics parameters. For each dynamic sequence, a random observed velocity from the long sequence is drawn as the initialization. The dataset is divided into 90% for training and 10% for testing, ensuring no repetition of the same identity, simulation parameters, or facial expression in both sets.

During training, the Adam optimizer executes 200k update steps with a learning rate of 0.0001, linearly decreasing to 0.00005, and a batch size of 128. The training specifications result in an approximate runtime of 8 h on an NVIDIA A6000. The relatively brief training duration can be attributed to the efficient network design and less noisy training data compared to scenarios typically encountered in image-based deep learning.

##### 4.2.2. Quantitative analysis

We quantitatively evaluate SoftDECA based on the L2 reconstruction error with respect to the targeted physics-based simulation and the computational runtimes. Additionally, we compare SoftDECA against Subspace Neural Physics (SNP) [14] and SoftSMPL [15] architectures adapted for facial simulations, recognized as state-of-the-art methods for rapid approximations of physics-based simulations. An overview of all results is provided in Table 1. The reported runtimes represent averages of ten runs measured on a consumer-grade Intel i5 12600 K processor. All implementations rely on PyTorch.<sup>1</sup>

SoftDECA outputs precise approximations for both static and dynamic animations, showcasing average test reconstruction errors of only 0.22 mm and 0.41 mm, respectively. The results underscore SoftDECA's capacity to generalize effectively across diverse human identities, facial expressions, and simulation parameters. However, the test data fully stems from unpersonalized blendshapes, necessitating further assessment using an external dataset obtained from 3DScanstore.<sup>2</sup>

The external data is comprised of 20 to 35 scanned facial expressions for each of seven human identities. We create personalized ARKit blendshapes per head using example-based facial rigging [57]. Subsequently, a test dataset is generated as before. Despite the possibility that the 3DScanstore examples may not align with the DECA distribution, the reconstruction error experiences only a marginal increase to 0.44 mm.

Noteworthy is SoftDECA's swift performance, with an average runtime of 7.45 ms for static frames and 9.87 ms for dynamic frames. This rapid processing makes SoftDECA an appealing choice for resource-demanding applications. Additionally, in scenarios where unseen personalized blendshapes are undesirable, we explored a variant of SoftDECA directly predicting vertex positions. This alternative achieves an accuracy of 0.16 mm and can be executed at an accelerated pace of 0.71 ms per frame.

##### 4.2.3. Static comparisons

In static simulations, SoftDECA is compared with SoftSMPL, as SNP is exclusively tailored for approximating dynamic effects. The key distinction between the SoftDECA and SoftSMPL architectures lies in the choice between our hypernetwork MLP and a conventional MLP. Originally designed for full-body applications, SoftSMPL takes a motion descriptor as input, summarizing a body and its state. In our case, this translates to blendshape weights, simulation parameters, and the identity code. To maintain consistent inference times, we employ identical network dimensions for the standard MLP as those in the hypernetwork. Consequently, the SoftSMPL MLP experiences a notable increase in the reconstruction error, averaging 1.67 mm. We also explore a larger MLP

<sup>1</sup> <https://pytorch.org>

<sup>2</sup> <https://www.3dscanstore.com>

**Table 1**

SoftDECA test results in comparison to adapted SNP [14] and SoftSMPL [15] architectures as well as ablations. The runtimes are averages measured on a consumer-grade Intel i5 12600K processor. External refers to the 3Dscanstore dataset. Small and large correspond to the size of the inspected MLP.

Model	Ours			SoftSMPL			SNP	Ablation	
	Static	Dynamic	External	Static (Small)	Static (Large)	Dynamic	Dynamic	Face-wise	Only Vertices
Error in mm	0.23	0.41	0.44	1.67	0.16	0.22	0.14	0.17	0.16
Time in ms	7.45	9.87	7.45	7.62	46.61	47.39	46.61	34.92	0.72

that achieves a comparable reconstruction error to SoftDECA, however, this results in a substantial increase in runtime to 46.61ms.

Another canonical alternative to the hypernetwork is a standard MLP that does not map to all DGs simultaneously but is evaluated face-wise. This approach yields a low reconstruction error of 0.17 mm, yet it comes with a higher runtime of 34.92 ms. Other architectures like CNNs, GNNs, or transformers could not be evaluated in real-time on a consumer-grade CPU with sufficient accuracy. For CNNs and GNNs, this is due to the fundamental sparse convolutions that are depended on very deep network layers to represent global effects (CNN, GNN). Further, transformer architectures usually require an attention mechanism with quadratic runtime but even optimized set transformer [53] involve significantly more operations than standard MLPs.

#### 4.2.4. Dynamic comparisons

For dynamic simulations, we compare SoftDECA with SoftSMPL and SNP. Unlike SoftDECA, both SoftSMPL and SNP perform dynamic computations in a latent space rather than directly on vertices. Further, SoftSMPL incorporates a recurrent GRU network [58], while SNP relies solely on a standard MLP. For this comparison, we only consider the *larger* network design mentioned earlier, as our primary focus is on evaluating the accuracy of our dynamic approximation rather than comparing runtimes. At this, both SoftSMPL and SNP exhibit slightly improved reconstruction errors at 0.22 mm and 0.24 mm, respectively. However, since both methods do not operate vertex-wise, they are not suitable for handling locally varying parameters of the dynamic simulation.

#### 4.2.5. Qualitative analysis

A visual illustration of SoftDECA's capabilities is given in Fig. 7, presenting a comparison between SoftDECA predictions and the targeted physics-based facial simulation. For example, in (a), it is evident that while collisions are not guaranteed to be entirely eliminated, they are largely mitigated. In (b), a localized increase in triangle strain on the skin around the cheeks results in the formation of wrinkles in that region. The result in (c) demonstrates the incorporation of external effects as heightened gravity. A *surgical manipulation* is shown in (d), where the jaw is lengthened along the vertical axis in the neutral state while maintaining the head's volume. The representation of a humanoid alien in (e) illustrates SoftDECA's robustness even outside the DECA distribution. This robustness is primarily achieved by transferring DGs instead of directly predicting vertex positions. Our interpretation of zombification in (f) is realized by expanding the skin area, highlighting SoftDECA's capability to closely approximate high-frequency details. Lastly, in (g-h), we depict the simulation of different weight additions in a non-linear manner, raising the soft tissue volume by 20% and 40%, respectively. Given the extensive training domain of SoftDECA, many other effects can be animated efficiently which are not displayed in Fig. 7. Additional results, including dynamic effects, are available in the supplementary material video.

### 4.3. SparseSoftDECA

#### 4.3.1. Dataset & training

For the training and assessment of SparseSoftDECA, we create a dataset consisting of 500k training and test examples by following the procedure outlined in Section 3.6.2. Specifically, we simulate 50

**Table 2**

SparseSoftDECA test results using both the same and a different head shape for personalization. Additionally, we investigate the influence of applying noise to the facial landmarks in the training set.

Model	Ours		Ablation	
	Same Identity	Other Identity	With Noise	Without Noise
Error in mm	0.54	0.62	0.55	0.73

distinct sets of facial expressions for each of 10,000 randomly selected identities. The dataset is divided into 90% for training and 10% for testing, ensuring that neither the same identity nor the same facial landmarks appear in both sets. To further rigorously evaluate the robustness of SparseSoftDECA in the face of incorrect and noisy inputs, as well as its generalization capacities, we extend  $\sigma$  in Eq. (24). In contrast to training examples, for test examples the process of personalizing the landmarks applies a separate test identity.

The training process and hyperparameters used are consistent with those described in Section 4.2.1.

#### 4.3.2. Quantitative analysis

SparseSoftDECA demonstrates the ability to closely mimic sparse landmark-guided simulations, as illustrated in Table 2. Whether personalization involves the same individual or a different one appears to be almost irrelevant. The minimal L2-errors of 0.54 mm and 0.62 mm affirm the robustness of SparseSoftDECA in handling erroneous and noisy inputs. We also investigated the influence of training data augmentation with Gaussian noise (standard deviation of 0.01). A slight improvement of the error from 0.73 mm to 0.55 mm can be observed.

In general, the errors observed are greater compared to those of SoftDECA. This can be attributed to the increased complexity of the task. Previously, the learning focus was primarily on changes in simulation properties, whereas now the learning task involves predicting entire facial expressions.

#### 4.3.3. Qualitative analysis

The images depicted in Fig. 8 illustrate landmarks, corresponding simulations, and predictions generated by SparseSoftDECA. In b), skin textures are exhibited aside of the geometry to demonstrate the quality of the final animation result. For the last row of b), Gaussian noise was applied to the landmarks, while all other examples are free of noise. On one hand, the reproduction quality evident from the measured test errors is visually confirmed. On the other hand, the benefits of physics-based simulations are reemphasized, highlighting their capacity to transform even highly noisy landmark inputs into anatomically plausible facial expressions. The principal advantage, however, is that all expressions were generated using only sparse landmarks as input and no underlying blendshapes had to laboriously sculpted. As a side effect, no blendshapes need to be stored, which can greatly reduce the memory footprint depending on the type of animation.

To observe the temporal consistency of SparseSoftDECA we kindly refer the reader to the attached video.

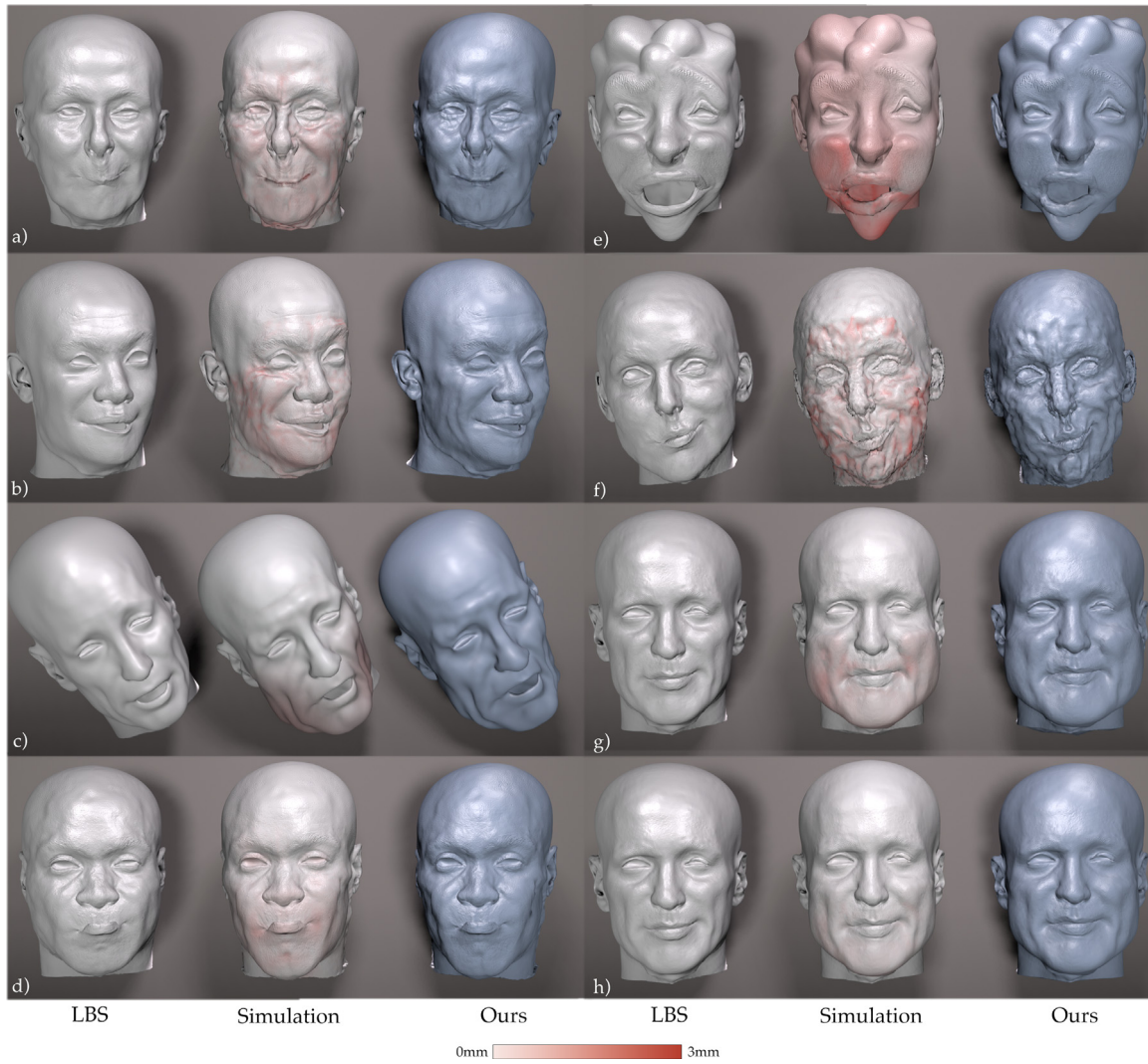


Fig. 7. Exemplary results of SoftDECA in comparison to the targeted physics-based facial simulation as well as the inputted linear blendshape expressions. Reconstruction errors are plotted on the simulated expressions.

## 5. Limitations

Although SoftDECA inherits most of the advantages of physics-based facial animations, it lacks the intrinsic handling of interactive effects such as wind or colliding objects. Moreover, although we allow for extensive localized artistic interventions, mixtures of material properties have not been part of the training data. Incorporating such mixtures into the training data is difficult as it is hard to define an adequate mixture distribution. Nonetheless, the smooth material blending of SoftDECA visually appears to be a sufficient approximation.

Despite SparseSoftDECA differing from SoftDECA in that it is not constrained by a specific set of blendshape weights, it operates on a predefined set of landmarks. However, this limitation could potentially be overcome in future research by implementing a training process that utilizes randomly selected landmark sets. In general, identifying an optimal set of landmarks is left to future work.

## 6. Conclusion

In this work, we have presented SoftDECA, which provides a computationally efficient approximation to physics-based facial simulations, even on consumer-grade hardware. With a few exceptions, most

simulation capabilities are retained, such as dynamic effects, volume preservation, wrinkle generation, and many more. SoftDECA's runtime performance is attractive for high-performance applications and low-cost hardware. In addition, it is versatile as it supports different head shapes, facial expressions, and material properties. The ability to make local adjustments after training makes it a valuable framework for artistic customization.

Our future goals for improving SoftDECA are twofold. On the one hand, we want to refine the anatomical model to achieve an even more accurate representation, especially for structures such as the trachea and esophagus. On the other hand, latest results demonstrate the efficient learning of contact deformations [59]. Given that people often touch their face several times a day, introducing a contact treatment for more realistic gestures could significantly improve immersion.

In continuation of the earlier presentation of SoftDECA [19], this work also includes the introduction of SparseSoftDECA. SparseSoftDECA enables blendshape-free facial animation based on sparse landmarks and exhibits the same generalization characteristics as SoftDECA. SparseSoftDECA seamlessly integrates with the avatar generation pipeline proposed by Wenninger et al. [20], making it straightforward to deploy.



**Fig. 8.** Exemplary results of SparseSoftDECA (right) in comparison to the targeted physics-based facial simulation (left) as well as the inputted landmarks (red dots). Additionally, in (b), the combination of SparseSoftDECA with skin textures is displayed. In the last row of (b), Gaussian noise has been applied to the landmarks.

#### CRediT authorship contribution statement

**Wagner Nicolas:** Writing – original draft, Visualization, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Schwanecke Ulrich:** Writing – review & editing, Conceptualization. **Botsch Mario:** Writing – review & editing, Conceptualization.

#### Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Nicolas Wagner reports financial support was provided by German Federal Ministry of Education and Research through the project HiAvA (ID 16SV8785). If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Data availability

The authors do not have permission to share data.

#### Acknowledgments

This research was supported by the German Federal Ministry of Education and Research (BMBF) through the project HiAvA (ID 16SV8785).

#### Appendix A. Simulation parameters

In the following, we describe all simulation parameters that haven been sampled during the creation of the SoftDECA training data. Moreover, we state the sampling range for each parameter. This list is not complete in the sense that SoftDECA is not committed to it. However,

these parameters already provide a comprehensive test of SoftDECA's capabilities and allow for extensive individualization opportunities.

- **Dynamics** We sample each of the parameters  $\alpha, \beta, \gamma$  that steer the dynamic second order effects in a range from 0 to 2.
- **Constraint Weights** All weights  $w_*$  associated with the constraints of  $\phi^\dagger$  are sampled between 0.001 and 100.
- **Volume** The target determinant in the volume energy  $E_{\text{vol}}$  is sampled from 0.5 to 1.5.
- **Maximum Strain** We allow a varying amount of maximum soft tissue strain by adjusting the  $\epsilon$  from 0.7 to 1.3.
- **Gravity** An additional gravity force is applied in a range from standard earth's gravity up to two times the standard. Further, the gravity direction is sampled.
- **Skull** We incorporate changes in the skull bones by sampling coordinate-wise scaling factors for both the cranium and jaw in the range from 0.5 to 1.5.

## Appendix B. Energies

In the following, we formally state all energies under optimization.

### Volume & Strain

$$E_{\text{vol}}(t) = (\det(F(t)) - 1)^2 \quad (\text{B.1})$$

$$E_{\text{str}}(t) = \min_{R \in SO(3)} \|F(t) - R\|_F^2 \quad (\text{B.2})$$

$F(t)$  denotes the deformation gradient of a tetrahedron  $t$ ,  $R \in SO(3)$  the optimal rotation, and  $\|\cdot\|_F$  the Frobenius norm.

### Bending

$$E_{\text{curv}}(x, B) = A_x \|\Delta x - R \Delta b_x\|^2 \quad (\text{B.3})$$

The matrix  $R \in SO(3)$  denotes the optimal rotation keeping the vertex Laplacian  $\Delta x$  as close as possible to its initial value  $\Delta b_x$ . The vertex Laplacian is discretized using the cotangent weights and the Voronoi areas  $A_x$  [60].

### Soft Dirichlet

$$E_{\text{tar}}(x, S_{\text{exp}}) = \|x - s_x\|^2, \quad (\text{B.4})$$

attracts each vertex  $x$  of the skin surface  $S$  to the corresponding vertex  $s_x$  from the target expression  $S_{\text{exp}}$ .

### Fitting Distances

$$E_{\text{dist}_2}(X, \hat{S}, D(\hat{S})) = \sum_{x \in X} (\|x - s_x\| - d_x)^2 \quad (\text{B.5})$$

ensures that for each vertex  $x \in X$  the predicted distance  $d_x \in D(\hat{S})$  is adhered to.

## Appendix C. Template layered head model

Table C.3 states the cardinality of each component of the layered head model template. By subdividing the wrap meshes or the triangle prisms between the wraps, the resolution of the template tetrahedron meshes can easily be adjusted. We will provide a mapping between the DECA and our topology.

## Appendix D. Network dimensions

See Fig. D.9.

## Appendix E. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.cag.2024.103903>.

Table C.3

Template dimensions.

Mesh	$S_T$	$B_T$	$M_T$	$\hat{S}_T$
#Vertices	35 621	14 572	16 388	7826
#Faces/#Tetrahedrons	71 358	28 856	32 370	15 648
Mesh	$\hat{B}_T$	$M_T$	$S_T$	$M_T$
#Vertices	7826	7826	49 852	
#Faces/#Tetrahedrons	15 648	15 648	123 429	73 681

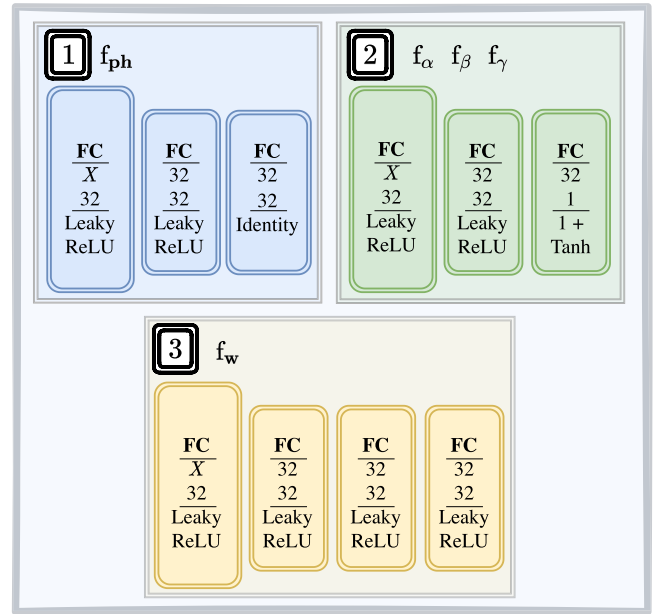


Fig. D.9. Network dimensions. Each fully connected layer (FC) is represented as a box. For each FC, the input and output dimensions are stated as well as the applied activation function.

## References

- [1] Cao C, Simon T, Kim JK, Schwartz G, Zollhoefer M, Saito S-S, et al. Authentic volumetric avatars from a phone scan. *ACM Trans Graph* 2022;41(4):1–19.
- [2] Grassal P-W, Prinzel M, Leistner T, Rother C, Nießner M, Thies J. Neural head avatars from monocular RGB videos. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, p. 18653–64.
- [3] Athar S, Xu Z, Sunkavalli K, Shechtman E, Shu Z. RigNeRF: Fully controllable neural 3D portraits. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, p. 20364–73.
- [4] Zielonka W, Bolkart T, Thies J. Instant volumetric head avatars. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, p. 4574–84.
- [5] Lewis JP, Anjyo K, Rhee T, Zhang M, Pighin FH, Deng Z. Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)* 2014;1(8):2.
- [6] Ichim A-E, Kadlecěk P, Kavan L, Pauly M. Phace: Physics-based face modeling and animation. *ACM Trans Graph* 2017;36(4):1–14.
- [7] Ichim AE, Kavan L, Nimier-David M, Pauly M. Building and animating user-specific volumetric face rigs. In: *Symposium on computer animation*. 2016, p. 107–17.
- [8] Cong MD. Art-directed muscle simulation for high-end facial animation. *Stanford University*; 2016.
- [9] Choi B, Eom H, Mouscadet B, Cullingford S, Ma K, Gassel S, et al. Anatomy: an animator-centric, anatomically inspired system for 3D facial modeling, animation and transfer. In: *SIGGRAPH Asia 2022 conference papers*. 2022, p. 1–9.
- [10] Yang L, Kim B, Zoss G, Gözcü B, Gross M, Solenthaler B. Implicit neural representation for physics-driven actuated soft bodies. *ACM Trans Graph* 2022;41(4):1–10.
- [11] Barrielle V, Stoiber N, Cagniard C. Blendforces: A dynamic framework for facial animation. *Comput Graph Forum* 2016;35(2):341–52.
- [12] Srinivasan SG, Wang Q, Rojas J, Klár G, Kavan L, Sifakis E. Learning active quasistatic physics-based models from data. *ACM Trans Graph* 2021;40(4):1–14.
- [13] Brandt C, Eisemann E, Hildebrandt K. Hyper-reduced projective dynamics. *ACM Trans Graph* 2018;37(4):1–13.

- [14] Holden D, Duong BC, Datta S, Nowrouzehzrai D. Subspace neural physics: Fast data-driven interactive simulation. In: Proceedings of the 18th annual ACM SIGGRAPH/eurographics symposium on computer animation. 2019, p. 1–12.
- [15] Santesteban I, Garces E, Otaduy MA, Casas D. Softsmpl: Data-driven modeling of nonlinear soft-tissue dynamics for parametric humans. *Comput Graph Forum* 2020;39(2):65–75.
- [16] Cong M, Fedkiw R. Muscle-based facial retargeting with anatomical constraints. In: ACM SIGGRAPH 2019 talks. 2019, p. 1–2.
- [17] Ha D, Dai A, Le QV. Hypernetworks. 2016, arXiv preprint arXiv:1609.09106.
- [18] Feng Y, Feng H, Black MJ, Bolkart T. Learning an animatable detailed 3D face model from in-the-wild images. *ACM Trans Graph* 2021;40(4):1–13.
- [19] Wagner N, Botsch M, Schwanecke U. Softdeca: Computationally efficient physics-based facial animations. In: Proceedings of the 16th ACM SIGGRAPH conference on motion, interaction and games. 2023, p. 1–11.
- [20] Wenninger S, Achenbach J, Bartl A, Latoschik ME, Botsch M. Realistic virtual humans from smartphone videos. In: Proceedings of the 26th ACM symposium on virtual reality software and technology. 2020, p. 1–11.
- [21] Ali-Hamadi D, Liu T, Gilles B, Kavan L, Faure F, Palombi O, et al. Anatomy transfer. *ACM Trans Graph* 2013;32(6):1–8.
- [22] Gilles B, Reveret L, Pai DK. Creating and animating subject-specific anatomical models. *Comput Graph Forum* 2010;29(8):2340–51.
- [23] Kadleček P, Ichim A-E, Liu T, Křivánek J, Kavan L. Reconstructing personalized anatomical models for physics-based body animation. *ACM Trans Graph* 2016;35(6):1–13.
- [24] Saito S, Zhou Z-Y, Kavan L. Computational bodybuilding: Anatomically-based modeling of human bodies. *ACM Trans Graph* 2015;34(4):1–12.
- [25] Schleicher R, Nitschke M, Martschinke J, Stamminger M, Eskofier BM, Klucken J, et al. BASH: Biomechanical animated skinned human for visualization of kinematics and muscle activity. In: VISIGraPP (1: GRAPP). 2021, p. 25–36.
- [26] Keller M, Zuffi S, Black MJ, Pujades S. OSSO: Obtaining skeletal shape from outside. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, p. 20492–501.
- [27] Keller M, Werling K, Shin S, Delp S, Pujades S, C. Karen L, et al. From skin to skeleton: Towards biomechanically accurate 3D digital humans. In: ACM TOG, Proc. SIGGRAPH Asia. 2023.
- [28] Komaritzan M, Wenninger S, Botsch M. Inside humans: Creating a simple layered anatomical model from human surface scans. *Front Virtual Real* 2021;2:694244.
- [29] Maalin N, Mohamed S, Kramer RS, Cornelissen PL, Martin D, Tovée MJ. Beyond BMI for self-estimates of body size and shape: A new method for developing stimuli correctly calibrated for body composition. *Behav Res Methods* 2021;53(3):1308–21.
- [30] Achenbach J, Brylka R, Gietzen T, zum Hebel K, Schömer E, Schulze R, et al. A multilinear model for bidirectional craniofacial reconstruction. In: Proceedings of the eurographics workshop on visual computing for biology and medicine. 2018, p. 67–76.
- [31] Ichim AE, Bouaziz S, Pauly M. Dynamic 3D avatar creation from hand-held video input. *ACM Trans Graph* 2015;34(4):1–14.
- [32] Bradley D, Heidrich W, Popa T, Sheffer A. High resolution passive facial performance capture. In: ACM SIGGRAPH 2010 papers. 2010, p. 1–10.
- [33] Zhang L, Snavely N, Curless B, Seitz SM. Spacetime faces: High-resolution capture for modeling and animation. In: Data-driven 3D facial animation. Springer; 2008, p. 248–76.
- [34] Parke FI. Control parameterization for facial animation. In: *Computer animation'91*. 1991, p. 3–14.
- [35] Lewis JP, Mooser J, Deng Z, Neumann U. Reducing blendshape interference by selected motion attenuation. In: Proceedings of the 2005 symposium on interactive 3D graphics and games. 2005, p. 25–9.
- [36] Zheng Y, Abrevaya VF, Bühler MC, Chen X, Black MJ, Hilliges O. Im avatar: Implicit morphable head avatars from videos. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, p. 13545–55.
- [37] Garbin SJ, Kowalski M, Estellers V, Szymanowicz S, Rezaeifar S, Shen J, et al. VolTeMorph: Realtime, controllable and generalisable animation of volumetric representations. 2022, arXiv preprint arXiv:2208.00949.
- [38] Song SL, Shi W, Reed M. Accurate face rig approximation with deep differential subspace reconstruction. *ACM Trans Graph* 2020;39(4):1–12.
- [39] Sifakis E, Neverov I, Fedkiw R. Automatic determination of facial muscle activations from sparse motion capture marker data. In: ACM SIGGRAPH 2005 papers. 2005, p. 417–25.
- [40] Bao M, Cong M, Grabli S, Fedkiw R. High-quality face capture using anatomical muscles. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 10802–11.
- [41] Kadleček P, Kavan L. Building accurate physics-based face models from data. *Proc ACM Comput Graph Interact Techn* 2019;2(2):1–16.
- [42] Bickel B, Lang M, Botsch M, Otaduy MA, Gross MH. Pose-space animation and transfer of facial details. In: Symposium on computer animation. 2008, p. 57–66.
- [43] Kozlov Y, Bradley D, Bächer M, Thomaszewski B, Beeler T, Gross M. Enriching facial blendshape rigs with physical simulation. *Comput Graph Forum* 2017;36(2):75–84.
- [44] Casas D, Otaduy MA. Learning nonlinear soft-tissue dynamics for interactive avatars. *Proc ACM Comput Graph Interact Techn* 2018;1(1):1–15.
- [45] Loper M, Mahmood N, Romero J, Pons-Moll G, Black MJ. SMPL: A skinned multi-person linear model. *ACM Trans Graph (Proc. SIGGRAPH Asia)* 2015;34(6):248:1–248:16.
- [46] Botsch M, Kobbelt L. Real-time shape editing using radial basis functions. *Comput Graph Forum* 2005;24(3):611–21.
- [47] Bouaziz S, Martin S, Liu T, Kavan L, Pauly M. Projective dynamics: Fusing constraint projections for fast simulation. *ACM Trans Graph* 2014;33(4):1–11.
- [48] Gietzen T, Brylka R, Achenbach J, zum Hebel K, Schömer E, Botsch M, et al. A method for automatic forensic facial reconstruction based on dense statistics of soft tissue thickness. *PLoS One* 2019;14(1):e0210257.
- [49] Komaritzan M, Botsch M. Projective skinning. *Proc ACM Comput Graph Interact Techn* 2018;1(1):1–19.
- [50] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 4401–10.
- [51] Botsch M, Sumner R, Pauly M, Gross M. Deformation transfer for detail-preserving surface editing. In: *Vision, modeling & visualization*. 2006, p. 357–64.
- [52] Sumner RW, Popović J. Deformation transfer for triangle meshes. *ACM Trans Graph* 2004;23(3):399–405.
- [53] Lee J, Lee Y, Kim J, Kosiorek A, Choi S, Teh YW. Set transformer: A framework for attention-based permutation-invariant neural networks. In: International conference on machine learning. 2019, p. 3744–53.
- [54] Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. *IEEE Trans Neural Netw* 2008;20(1):61–80.
- [55] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun ACM* 2021;65(1):99–106.
- [56] Beeler T, Bradley D. Rigid stabilization of facial expressions. *ACM Trans Graph* 2014;33(4):1–9.
- [57] Li H, Weise T, Pauly M. Example-based facial rigging. *ACM Trans Graph* 2010;29(4):1–6.
- [58] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. 2014, arXiv preprint arXiv:1412.3555.
- [59] Romero C, Casas D, Chiaramonte MM, Otaduy MA. Contact-centric deformation learning. *ACM Trans Graph* 2022;41(4):1–11.
- [60] Botsch M, Kobbelt L, Pauly M, Alliez P, Lévy B. Polygon mesh processing. CRC Press; 2010.



## Citation

**AnaConDaR: Anatomically-Constrained Data-Adaptive Facial Retargeting**

Nicolas Wagner, Ulrich Schwanecke, and Mario Botsch  
Computers and Graphics 122, 2024  
DOI: [10.1016/j.cag.2024.103988](https://doi.org/10.1016/j.cag.2024.103988)

## 5.1 METHOD SUMMARY

In the publication *AnaConDaR*, we responded to Research Question 3 (Section 1.2.3).

This publication contributes to facial retargeting in two ways. Firstly, an *anatomical deformation transfer* ( $A$ - $DT$ ), which relies on a physics-based simulation (PBS) to transfer a facial expression from a source to a target character, and explicitly endeavors to preserve the perception of expression characteristics. Secondly, a data-driven extension to  $A$ - $DT$  that additionally leverages exemplary facial expressions of the target. In the following, we provide more details on the two contributions and refer to the extension with *AnaConDaR*.

## ANATOMICAL DEFORMATION TRANSFER

The idea of  $A$ - $DT$  is inspired by the original *deformation transfer* ( $DT$ ) [13, 105], please refer to Section 2.1.4 for an algorithmic description. Here, however, we first register the neutral source head to the facial expression to be transferred with an inverse PBS akin to that of *SoftDECA* (Chapter 3). Subsequently, we transfer the identified volumetric anatomical deformations to the neutral target head, also by means of a (forward) PBS.

In the forward simulation, we additionally consider the perception of the transferred expression. This aspect is inherently subjective and challeng-

ing to define precisely. As guidance, we engaged in a thought experiment: when shown a facial expression in a photograph and asked to imitate this expression using a mirror, what characteristics do humans focus on? We concluded that relative changes in facial proportions (e.g., how open the mouth or eyes are) and mouth contours are most vital. Consequently, we integrate these characteristics into the forward simulation through suitable constraints.

#### PATCHWISE LINEAR BLENDSHAPES

For  $A$ - $DT$ , we only need to know the neutral target head; however, if exemplary facial expressions of the target character are available, incorporating them via *example-based facial rigging* ( $EBFR$ ) [58] has proven effective for enhancing the original  $DT$ .  $EBFR$  aims to retarget facial expressions such that the retargeting results can simultaneously reproduce the target examples via weighted linear interpolation. In other words,  $EBFR$  relies on *linear blendshapes* ( $LBS$ ), which suffer from the well-known linearity-related drawbacks as discussed before (Section 2.1.1).

In contrast, *patchwise LBS* [18, 119], which employ varying blendshape weights across different facial patches, offer superior expressiveness due to their inherent nonlinearity. *Patchwise LBS* already demonstrated to be especially beneficial for facial retargeting [18], even though they can lead to unnatural edges between patches. For *AnaConDaR*, we integrate  $A$ - $DT$  with *patchwise LBS* instead of  $EBFR$  and handle implausible edges, as always in this thesis, by leveraging an anatomically plausible PBS. Unlike  $EBFR$ , we incorporate target examples only when they offer valuable information for retargeting. If they do not contribute to the expression transfer, we locally fall back to the  $A$ - $DT$  retargeting. Due to this adaptive blending scheme, *AnaConDaR* is devoid of artifacts that might result from an insufficient data foundation.

## 5.2 DISCUSSION

### RESULTS

Evaluating facial retargeting poses a significant challenge due to the subjective perception of the results, making it difficult to quantitatively compare the performance of different concepts. Therefore, we conducted two

comprehensive user studies involving approximately 30 participants. The first study directly compared *AnaConDaR* with a state-of-the-art peer group, while the second study validated the effectiveness of individual components of our approach in an ablation study. The former study additionally investigated how the number of available target examples affects the user perception. Our findings indicate that users consistently perceive *AnaConDaR* as more authentic across all scenarios and that all method components make a valid contribution to the overall results. Notably, our method becomes even more convincing when fewer target examples are provided. This is a crucial observation since 3D scanning and manual editing are very time-consuming, a problem that we already faced several times in this thesis.

Through visual results, we demonstrate that *AnaConDaR* is temporally consistent and illustrate the intuition of the local blending scheme. Moreover, the publication entails visually convincing examples that *A-DT* preserves facial characteristics better than *DT* and that *AnaConDaR* generally generates more expressivity from exemplary target data than *EBFR*.

Finally, *AnaConDaR* offers numerous straightforward artistic intervention possibilities, such as adjusting patchwise blendshape weights and manipulating facial characteristics like the mouth contour. Similar to *Soft-DECA*, a user can tweak simulation and material parameters, enabling effects such as body fat changes with minimal effort. Unlike our previously discussed approaches, *AnaConDaR* does not involve neural networks, which preserves *all* the advantageous features of PBSs. For instance, dynamic external effects such as fluctuating winds can be directly incorporated into the retargeting process.

#### LIMITATIONS

The most significant limitation of *AnaConDaR* stems from its initial conception, where we design facial characteristics on theoretical assumptions instead of an empirical validation. Although our ablation studies demonstrated the benefits of these characteristics, they did not verify the foundational thought experiment that guided their selection. Conducting preliminary studies to identify what users genuinely focus on when assessing retargeting quality could have led us to choose more effective characteris-

tics and potentially improved outcomes. Unfortunately, we overlooked the opportunity to explore this question within our user studies, too.

Another limitation of our work is that the current implementation of *AnaConDaR* does not support real-time processing; it operates at less than one frame per second when utilizing all features. However, for the overarching framework of this thesis, the impact of this limitation is rather small. This is because we primarily intend to use *A-DT* to algorithmically generate blendshapes once, which can then be animated in real-time using our own *SoftDECA*. In many other scenarios, such as studio productions or game cutscenes, online (real-time) retargeting is not necessary either. Nonetheless, it is worth noting that our results rely on a pure CPU-based implementation, utilizing widely adopted frameworks for which numerous GPU alternatives exist. Therefore, *AnaConDaR* has the potential to achieve real-time execution speed with appropriate optimization and hardware acceleration.

Lastly, *AnaConDaR* currently adheres to a predetermined topology, necessitating that both the source expressions to be retargeted and the target examples conform to this structure. Otherwise, the patch and facial characteristic definitions, which are based on vertex selections, would no longer be valid. While several algorithmic methods exist for finding mappings between different topologies [100], achieving a sufficient accuracy usually requires manual intervention. We mitigate this issue by employing a standard template topology across all our contributions, which ensures direct compatibility between methods like *A-DT* and *SoftDECA*. In this context, we also want to point out that we have not evaluated the effects of a change to the patch configuration. Nevertheless, other research on *patchwise LBS* indicates that these effects are generally minimal [18].

#### RELATED WORK

The related work on *AnaConDaR* divides into two main areas: previous efforts concerning the “data-free” *A-DT*, and data-driven facial retargeting as implemented by *AnaConDaR* in total.

Between the original *DT* [105] and *A-DT*, only a few enhancements specifically related to faces emerged, likely due to the original’s fundamentally compelling and reliable results. Xu et al. [122] use a customized *DT* for edges focusing on lip and eye contours, Bhat et. al. [9] show how to transfer lip contours to humanoid aliens, and, most closely related to our

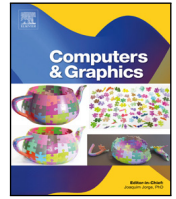
$A$ - $DT$ , Onizuka et al. [84] propose a  $DT$  with locally scaled vertex targets to keep facial contours consistent. None of the existing approaches considers facial characteristics to the extent our method does, nor do they incorporate them within a volumetric simulation of the anatomy.

In addition to the discussion of reusing more general, data-driven facial animation methods for retargeting as described in Chapter 2, we want to highlight how *AnaConDaR* differs from the previous state-of-the-art *Anatomical Local Model* [18]. This earlier approach also employs *patchwise LBS* but relies on a simplified surface description of anatomical properties instead of a volumetric one. This simplification limits artistic flexibility and fails to resolve issues like self-collisions. Furthermore, the *Anatomical Local Model* requires the target examples to be semantically related to the source expression – a considerable limitation that *AnaConDaR* does not share.

#### FUTURE WORK

We anticipate that future developments most relevant to this thesis will primarily be related to  $A$ - $DT$  and, therefore, automatic blendshape generation. As can currently be observed with many heuristic facial animation methods, a ceiling for improvements becomes very close, and data-driven approaches promise breakthroughs [70, 90]. An attempt to generate personalized blendshapes without target examples using neural networks has already been made [59]. However, this network is trained on a small training corpus and is only partially convincing in terms of the level of detail compared to  $DT$ , for example. Hence, considerable development potential exists, especially for hybrid methods that predict deformations and then robustly simulate them with  $A$ - $DT$ . Such approaches could effectively combine the strengths of both worlds: leveraging neural networks’ predictive power alongside the constraining capabilities of PBSs.

#### 5.3 PUBLICATION



## Technical Section

## AnaConDaR: Anatomically-Constrained Data-Adaptive Facial Retargeting

Nicolas Wagner<sup>a,\*</sup>, Ulrich Schwanecke<sup>b</sup>, Mario Botsch<sup>a</sup><sup>a</sup> TU Dortmund University, Otto-Hahn-Str. 16, 44227 Dortmund, Germany<sup>b</sup> University of Applied Sciences RheinMain, Kurt-Schumacher-Ring 18, 65197 Wiesbaden, Germany

## ARTICLE INFO

## Keywords:

Facial animation  
 Offline performance retargeting  
 Physics-based simulation

## ABSTRACT

Offline facial retargeting, i.e., transferring facial expressions from a source to a target character, is a common production task that still regularly leads to considerable algorithmic challenges. This task can be roughly dissected into the transfer of sequential facial animations and non-sequential blendshape personalization. Both problems are typically solved by data-driven methods that require an extensive corpus of costly target examples. Other than that, geometrically motivated approaches do not require intensive data collection but cannot account for character-specific deformations and are known to cause manifold visual artifacts.

We present AnaConDaR, a novel method for offline facial retargeting, as a hybrid of data-driven and geometry-driven methods that incorporates anatomical constraints through a physics-based simulation. As a result, our approach combines the advantages of both paradigms while balancing out the respective disadvantages. In contrast to other recent concepts, AnaConDaR achieves substantially individualized results even when only a handful of target examples are available. At the same time, we do not make the common assumption that for each target example a matching source expression must be known. Instead, AnaConDaR establishes correspondences between the source and the target character by a data-driven embedding of the target examples in the source domain. We evaluate our offline facial retargeting algorithm visually, quantitatively, and in two user studies.

## 1. Introduction

Creating high-fidelity facial expressions for human or humanoid characters is one of the most challenging problems in computer graphics applications. To that end, it is common practice to record a source actor with high-resolution motion capture technology and subsequently transfer the scanned expressions to the targeted character either frame-by-frame or via blendshapes [1]. A comprehensive corpus of research focuses on the latter step, the so-called offline facial performance retargeting. While deep learning predominates in various facial animation tasks, here, *more traditional* approaches retain distinct advantages and are commonly used in production [2]. Particularly, due to the still limited availability of high-resolution facial expression meshes for training, the risk of generalization gaps is ubiquitous [3]. The reliance on implicit representations within current neural telepresence applications [4,5] underscores the lack of suitable training data.

Two main streams of work can be identified within which most of the current non-learning methodologies can be categorized. On the one hand, there are data-driven methods that have access to numerous exemplary facial expressions of the target character and form new expressions by combining these [2,6]. On the other hand, there are geometry-driven methods that try to transfer the geometric deformations of the source actor's face to the target character [7–9]. Both

methodologies offer complementary advantages and disadvantages. For instance, data-driven methods can consider anatomy-specific differences between the source and the target, whereas geometry-driven methods force deformations regardless of the structure of the respective heads. In return, geometry-driven methods do not rely on elaborately recorded or artistically sculpted examples of the target character and are, therefore, usually more efficient than data-driven methods.

Generally, there is a trade-off between the cost and complexity of data acquisition and retargeting quality. When time and effort are not a constraint, establishing extensive corresponding linear blendshape (LBS) systems [1] between the source and target character can be the most reasonable approach to facial retargeting. As such situations rarely occur in reality, the current state-of-the-art Anatomical Local Model (ALM) [2] has been developed. ALM requires a significantly reduced amount of blendshapes due to replacing plain LBS with more expressive patchwise LBS (PLBS). However, the authors point out that insufficiently comprehensive PLBS nonetheless result in severe retargeting artifacts and recognize the limitation that non-corresponding source and target blendshapes are not supported. Similar shortcomings in LBS can partially be overcome by employing example-based facial rigging (EBFR) [6], which supplements the data-driven retargeting with

\* Corresponding author.

E-mail address: [nicolas.wagner@tu-dortmund.de](mailto:nicolas.wagner@tu-dortmund.de) (N. Wagner).

a geometry-driven deformation transfer [7]. Unfortunately, there has not been an adaption to ALM so far.

In this work, we improve on ALM and fill this very gap by introducing AnaConDaR, an anatomically-constrained and data-adaptive facial retargeting. Here, corresponding PLBS systems are derived from the available target examples and used for an initial retargeting in a data-driven manner. The parts that are not explainable by PLBS are retargeted by a novel anatomical deformation transfer (ADT). In a final step, both the PLBS and ADT results are added together and a physics-based simulation ensures anatomical plausibility, also with combined retargeting. Moreover, this simulation enables artistic interventions on material properties, can incorporate external forces, and preserves expression-specific characteristics.

We evaluate AnaConDaR in two user studies and a quantitative comparison. In one user study, we asked the participants to benchmark the state-of-the-art peer group against AnaConDaR, while the other focused on the necessity of individual algorithmic components. Quantitative comparisons of facial retargeting algorithms are generally challenging, as the subjective nature of perceiving facial expressions makes it difficult to establish a definitive ground truth. Therefore, we quantitatively showcase the advantages of AnaConDaR over ALM in a particularly construed retargeting scenario.

The *key novelties and contributions* we present in this paper can be summarized as follows:

- A novel hybrid approach for offline facial performance retargeting that can leverage a small number of target examples.
- A new, fully volumetric deformation transfer for faces, which respects anatomical and physical constraints. During the deformation transfer, expression-specific characteristics are retained.
- Two user studies, a quantitative analysis, and various visual examples that evaluate and showcase AnaConDaR.

## 2. Related work

### 2.1. Facial retargeting in general

Besides offline performance targeting, there are several other variants of facial retargeting, which are all related but can also be clearly distinguished.

First, the 2D variant in which so-called deep fakes [10–18] swap faces directly in images almost entirely independent of the underlying geometry [16,19]. While these works can generate outstanding results, they are hardly artist-controllable, cannot integrate physics-based effects, and lose mesh-based advantages like shading adjustments. Our approach offers all of the features mentioned above.

Second, online performance retargeting algorithms that animate characters in real time. Usually, such methods are either of low quality [1,20] or need time consuming training on extensive datasets [21–27]. Our approach can handle high resolutions, is applicable without training, and only requires a handful of expression examples.

Third, more general (neural) face models [3,28–32] that capture both human identities and facial expressions in latent spaces. Unfortunately, their generalization capabilities usually do not meet the quality requirements of sophisticated CGI productions [30,32]. Moreover, many models can only perform the facial retargeting task for low-resolution geometries [28,29,33]. Starting from a reversed perspective, the neural physics-based facial animation of Yang et al. [26] has recently been extended into a more comprehensive face model [3]. Nonetheless, this model is severely limited to only a handful of identities and adding a novel identity requires five days of retraining [26]. Further, they expect access to 30 s of performance capture per identity while the captured expressions must be semantically aligned. The likewise neural approach AnimateMy [25] faces similar problems. Neither of the latter two algorithms [3,25] was evaluated concerning facial retargeting.

Finally, image-based face avatars primarily work on low-resolution geometries [4,5] and, hence, do not meet production requirements, as well. Overall, we follow the recent assessment of Chandran et al. [2] that deep learning for facial retargeting still cannot fully compete with *more traditional* techniques.

### 2.2. Offline facial performance retargeting

As the introduction notes, offline facial performance retargeting without learning can be divided mainly into data-driven and geometry-driven methods. For data-driven methods, linear blendshapes [1] are still the gold standard due to their simplicity and computational speed. Since the nonlinear aspects of facial expressions have a significant influence, a variety of extensions [34–36] have been developed over the years. Nonetheless, only minor improvements have been achieved, and it remains common practice to model or scan a large number of linear blendshapes to account for nonlinearity. In an effort to reduce costs, methods have been developed that generate extensive blendshape rigs from just a few exemplary expressions [6,33]. Often, however, these only exhibit weak personalization. Recently, Chandran et al. [2] demonstrated how to gain more expressiveness from expression samples using piecewise linear blendshapes. To the best of our knowledge, none of the aforementioned data-driven techniques deals with missing information due to insufficient training data. The method we present in this work addresses this problem by combining piecewise linear blendshapes with a geometry-driven approach.

The most widely used geometry-driven facial retargeting approach is deformation transfer [7–9]. This approach extracts deformation gradients from a source expression and applies them to the neutral target face. Closely related is delta transfer, which transfers deformations in the form of (scaled) per-vertex displacements. However, neither deformation nor delta transfer can prevent the retargeting of character-specific details. Further, many known artifacts arise, such as loss of volume, self-collisions, and incorrectly transmitted deformation amplitudes. A body of related work is therefore concerned with explicitly distinguishing expression-specific from character-specific details [9,37,38]. For instance, Onizuka et al. [9] propose a locally scaled deformation transfer to keep facial contours consistent, Xu et al. [37] use an adapted deformation transfer for edges to focus on lip and eye contours, and Bhat et al. [38] show how to transfer lip contours to humanoid aliens. In contrast to previous work, we design facial features that aim to retain not only contours but also other facial proportions. Furthermore, we use a fully volumetric approach to avoid artifacts like volume loss and self-collisions.

## 3. Method

### 3.1. Problem statement & method overview

The input to *offline facial performance retargeting* is a facial animation of a source character captured as a set  $S = \{S_i\}_{i=0}^N$  of  $N + 1$  surface meshes with identical tessellation. The overall goal is to curate a corresponding set of surface meshes  $\mathcal{T} = \{T_i\}_{i=0}^N$  for a different target character, such that each expression  $T_i$  exhibits the same characteristics as  $S_i$ . These characteristics are primarily rooted in human perception and, therefore, difficult to capture through formal means.

To achieve this goal, we present AnaConDaR (Section 3.2), a mainly data-driven approach to facial retargeting, which is supplemented by a geometry-driven component (Section 3.3) whenever the available data is not sufficiently expressive. Moreover, anatomical plausibility and expression characteristics are ensured through a quasi-static physics-based simulation (Section 3.4).

In the ensuing formal derivation of AnaConDaR, we follow a top-down scheme in which we first explain the fundamental functionality of our approach (Section 3.2). Afterward, individual constituents are explained in more detail (Sections 3.3, 3.4, and 3.5). To ease the reading flow, Table 1 gives a summary of the notation. We slightly abuse the notation by denoting a surface mesh and the corresponding vector of stacked vertex positions with the same symbol.

**Table 1**  
An overview of the notation of AnaConDaR.

Notation	Description
$M$	Surface mesh and stacked vertex positions
$S, \mathcal{T}$	Source and retargeted animation
$S_{\mathcal{E}}, \mathcal{T}_{\mathcal{E}}$	Source and target examples
$S, T$	Neutral head surfaces
$S_i, T_i$	Source expression and AnaConDaR retargeting
$S_i^L, T_i^L$	Reconstruction and retargeting of $S_i$ with LBS
$S_i^P, T_i^P$	Reconstruction and retargeting of $S_i$ with PLBS
$w_i^L, w_i^P$	Optimal LBS and PLBS reconstruction weights
$\hat{S}_i^M, \hat{T}_i^M$	Missing delta blendshapes
$S_i^M, T_i^M$	Missing blendshapes
$\mathbb{S}, \mathbb{M}$	Template soft and muscle tissue tetrahedra meshes
$H_S, H_T$	Source and target heads
$F_i$	Facial characteristics



**Fig. 1.** The patch layout (80 patches) we use has been automatically determined with METIS [39].

## 3.2. Anatomically-constrained data-adaptive facial retargeting

### 3.2.1. Data-driven component

For the derivation of the data-driven component of AnaConDaR, we initially assume to have access to a set of target examples  $\mathcal{T}_{\mathcal{E}}$  with corresponding expressions  $S_{\mathcal{E}} \subset S$ . This assumption will be lifted in Section 3.5. Further, we expect the neutral head surfaces  $S$  and  $T$  of both characters to be known. In such situations, a variety of blendshape concepts can be applied for data-driven facial retargeting. For example, plain linear blendshapes (LBS) [1] first *approximate* each source expression  $S_i \in S$  by a linear combination

$$S_i^L = S + \sum_{S_j \in S_{\mathcal{E}}} w_{ij}^L (S_j - S) \quad (1)$$

of the source examples  $S_{\mathcal{E}}$ . The optimal blending weights  $w_i^L = (\dots, w_{ij}^L, \dots)$  are the solution of the linear least squares problem

$$w_i^L = \arg \min_{w_i} \left\| S + \sum_{S_j \in S_{\mathcal{E}}} w_{ij} (S_j - S) - S_i \right\|^2 + \lambda_{reg} \|w_i\|^2, \quad (2)$$

where the first term draws the blended surface  $S_i^L$  to the targeted expression  $S_i$ . Since this reconstruction is underconstrained, the second term adds the squared norm  $\|w_i\|^2$  of the blending weights to regularize them to be close to zero. The factor  $\lambda_{reg} \in \mathbb{R}$  controls the strength of the regularization. Subsequently, the LBS *retargeting*

$$T_i^L = T + \sum_{T_j \in \mathcal{T}_{\mathcal{E}}} w_{ij}^L (T_j - T) \quad (3)$$

is obtained by simply applying the optimized weights  $w_i^L$  to the target examples  $\mathcal{T}_{\mathcal{E}}$ .

*Patchwise* linear blendshapes (PLBS) outperform the classical LBS in efficiency and expressiveness [2,40]. Our implementation partitions all vertices consistently into a set of small (non-overlapping)

patches (Fig. 1) and performs the LBS retargeting defined in Eqs. (1)–(3) independently for each patch. We refer to the resulting source approximation of PLBS as  $S_i^P$  and to the retargeted expression as  $T_i^P$ .

The PLBS retargeting  $T_i^P$  is the *data-driven component* of AnaConDaR.

### 3.2.2. Geometry-driven component

Although variants of PLBS are the foundation of the current state-of-the-art in facial retargeting [2], errors in the source approximation

$$\hat{S}_i^M = S_i - S_i^P \quad (4)$$

are inevitably retargeted, as well. Seen from a different perspective,  $\hat{S}_i^M$  is a missing delta blendshape for which no corresponding blendshape  $T_i^M = \hat{T}_i^M + T$  is known. We approximate

$$T_i^M = \text{adt}(S_i^M, S, T) \quad (5)$$

with a novel (geometry-driven) deformation transfer  $\text{adt}$  (Section 3.3), which transfers the deformations of the missing blendshape  $S_i^M = \hat{S}_i^M + S$  from the source to the target character. As opposed to the original deformation transfer [7],  $\text{adt}$  is physics-based, volumetric, and anatomically-constrained. Moreover,  $\text{adt}$  preserves expression-specific characteristics from  $S_i^M$  in  $T_i^M$ .

The retargeted missing delta blendshape  $\hat{T}_i^M = T_i^M - T$  is the *geometry-driven component* of AnaConDaR.

### 3.2.3. Assembling the components

AnaConDaR processes the sum of both the actual patchwise blendshapes  $T_i^P$  (data-driven component) and the missing delta blendshape  $\hat{T}_i^M$  (geometry-driven component) with the physics-based simulation  $\text{anacon}$  (Section 3.4) to form the final retargeting

$$T_i = \text{anacon}(T_i^P + \hat{T}_i^M, S_i, S, T). \quad (6)$$

Conceptually,  $\text{anacon}$  is similar to  $\text{adt}$  and also enhances the retargeting plausibility through anatomical constraints as well as expression-specific characteristics. Additionally, visible patch boundaries are eliminated, which can occur in the PLBS result  $T_i^P$ .

Summarized in words, AnaConDaR retargets as extensively as possible through exemplary data but does not lose valuable information due to source approximation errors, since these are corrected with the geometry-driven component. The overview of AnaConDaR described so far is also visualized in steps 2 and 3 of Fig. 2.

Next, we will depict  $\text{adt}$  and  $\text{anacon}$  in more detail. As both only differ slightly, we will explain them using the example of  $\text{adt}$  (Section 3.3) and then discuss the differences to  $\text{anacon}$  (Section 3.4). Finally, we will resolve the initial assumption of corresponding source and target examples  $S_{\mathcal{E}}$  and  $\mathcal{T}_{\mathcal{E}}$  (Section 3.5).

## 3.3. Anatomical deformation transfer

### 3.3.1. Overview

Given the neutral head surfaces  $S$  and  $T$  of the source and target character,  $\text{adt}$  executes four fundamental functions for retargeting the missing blendshape  $S_i^M$  to  $T_i^M$  as outlined in Algorithm 1. To facilitate the introduction of  $\text{adt}$ , we again follow a top-down scheme and first give a brief overview of every function in this section. The subsequent Section 3.3.2 provides the corresponding detailed descriptions, each of which can be found in an identically named paragraph.

*Template fitting.* As a first step, the function  $\text{fitHead}$  creates volumetric head representations for the source and target character by fitting a template head  $H = (\mathbb{S}, \mathbb{M}, B)$  to the neutral surfaces  $S$  and  $T$ . The template comprises a soft tissue tetrahedra mesh  $\mathbb{S}$ , a muscle tissue tetrahedra mesh  $\mathbb{M}$ , and a skull surface mesh  $B$ . Please refer to Fig. 3 for a visualization of the corresponding surfaces and more details. The resulting heads

$$H_S = (\mathbb{S}_S, \mathbb{M}_S, B_S) = \text{fitHead}(S, H) \quad (7)$$

$$H_T = (\mathbb{S}_T, \mathbb{M}_T, B_T) = \text{fitHead}(T, H)$$

consist of the fitted components.



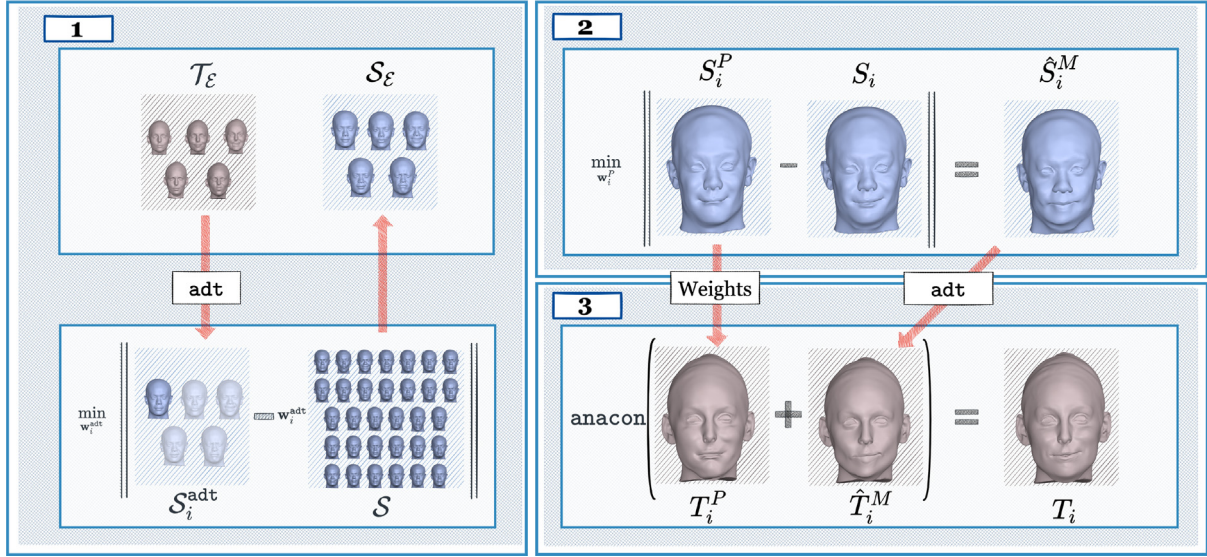


Fig. 2. An overview of AnaConDaR (Section 3.2). In the first step, the target examples  $\mathcal{T}_\epsilon$  are mapped into the source domain (Section 3.5). In the second step, the embedded expressions  $\mathcal{S}_\epsilon$  are used to form a PLBS approximation  $S_i^P$  of the targeted expression  $S_i$  by optimizing the patchwise blending weights  $w_i^P$  (Section 3.2.1). In step three, the evaluation of the same patchwise weights in the target domain  $T_i^P$  is supplemented with the adt result (Section 3.3) of the missing blendshape  $T_i^M$  (Section 3.2.2). Lastly, anacon ensures anatomical plausibility of the final AnaConDaR retargeting  $T_i$  (Section 3.4).

#### Algorithm 1 Anatomical Deformation Transfer

##### Input

$S_i^M$  The missing blendshape  
 $S, T$  The neutral head surfaces

##### Function $\text{adt}(S_i^M, S, T)$

```
// Section 3.3.2 Template Fitting.
 $H_S = \text{fitHead}(S, H), H_T = \text{fitHead}(T, H)$ 
// Section 3.3.2 Inverse Simulation.
 $(\nabla S, \nabla M, \nabla B) = \text{invSim}(S_i^M, H_S)$ 
// Section 3.3.2 Facial Characteristics.
 $F_i = \text{fc}(S_i^M, S, T)$ 
// Section 3.3.2 Forward Simulation.
 $T_i^M = \text{fwdSim}((\nabla S, \nabla M, \nabla B), F_i, H_T)$ 
// Return the retargeted missing blendshape.
return  $T_i^M$ 
end Function
```

*Inverse simulation.* After fitting the template, the inverse physics-based simulation

$$(\nabla S, \nabla M, \nabla B) = \text{invSim}(S_i^M, H_S) \quad (8)$$

identifies volumetric changes of the source head  $H_S$  to form the targeted missing blendshape  $S_i^M$  while respecting bio-mechanical and physical properties. Here,  $\nabla S$  and  $\nabla M$  are stacked per tetrahedron  $3 \times 3$  deformation gradients that capture changes in soft and muscle tissue  $\mathbb{S}_S$  and  $\mathbb{M}_S$ , respectively. For the jaw and cranium parts of  $B_S$ , rigid movements are individually captured by  $\nabla B$ .

*Facial characteristics.* Alongside the volumetric changes, the function  $\text{fc}$  identifies expression-specific facial characteristics

$$F_i = \text{fc}(S_i^M, S, T) \quad (9)$$

in the missing blendshape  $S_i^M$  and adapts them to the target character. These characteristics are our answer to the following thought experiment:

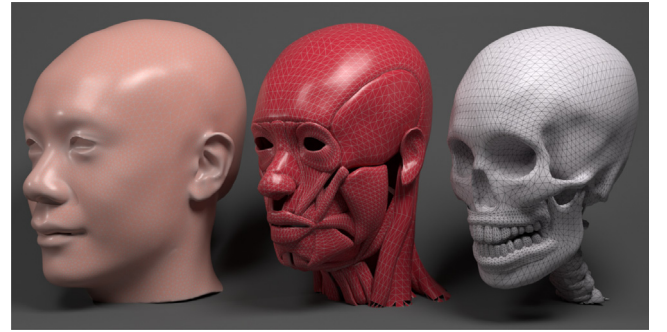


Fig. 3. The surfaces of all anatomical structures that are part of the volumetric head template we use for AnaConDaR. From left to right, the surface of the soft tissue, the surface of the muscle tissue, and the skull surface. The soft tissue includes the neutral head surface, the muscle tissue is connected to the skull as well as the soft tissue, and the skull is separated into jaw and cranium.

“If you are given a picture of an expression to mimic and a mirror to look at yourself, what do you use as guidance?”

We assume that human perception is guided by relative changes of face openings and facial contours which can be influenced through muscle activation. More specifically, we assume that the eyes in the missing and the retargeted blendshape should open and close by almost the same relative proportions while the skin around the eye sockets is assumed to move in a consistent manner. Furthermore, we expect the lips to form similar contours in both, since these can be manipulated by humans with a great degree of control.

*Forward simulation.* Finally, the forward physics-based simulation  $\text{fwdSim}$  generates the retargeted missing blendshape

$$T_i^M = \text{fwdSim}((\nabla S, \nabla M, \nabla B), F_i, H_T) \quad (10)$$

by applying the previously calculated volumetric changes  $(\nabla S, \nabla M, \nabla B)$  and facial characteristics  $F_i$  to the target head  $H_T$ .

### 3.3.2. Constituents

In the remainder of this section the four adt functions  $\text{fitHead}$ ,  $\text{invSim}$ ,  $\text{fc}$ , and  $\text{fwdSim}$  are precisely described.

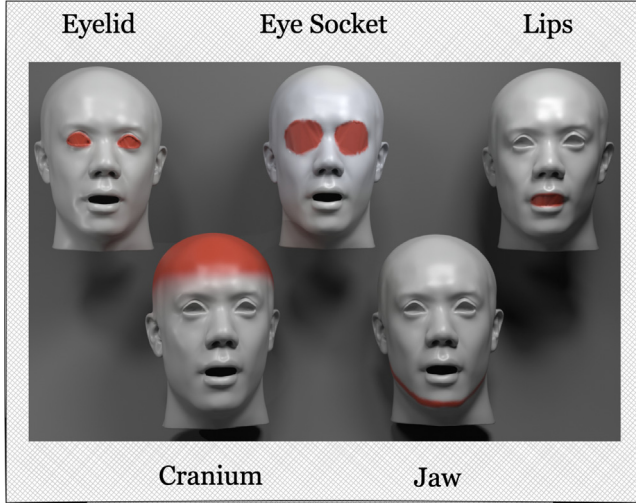


Fig. 4. The supplementary meshes that are used to determine the position of the jaw and cranium bones, as well as the expression-specific characteristics.

**Template fitting.** The template fitting  $\text{fitHead}$ , which fits the volumetric head template  $H = (\mathbb{S}, \mathbb{M}, B)$  to the neutral source and target head surfaces  $S$  and  $T$ , performs two steps.

1. The skull  $B$  is placed by a dense linear model trained on the computed tomography dataset of Achenbach et al. [41]. This model maps from the vertex positions of the head surface to the vertex positions of the skull surface.
2. Soft and muscle tissue  $\mathbb{S}, \mathbb{M}$  are positioned by a radial basis function (RBF) space warp [42] calculated from the template to the targeted head and skull surfaces. By the construction of RBFs, the vertices of  $\mathbb{S}$  and  $\mathbb{M}$  are warped to a similar semantic position as in the template.

**Inverse simulation.** The inverse simulation  $\text{invSim}$ , which aligns the source head  $H_S = (\mathbb{S}_S, \mathbb{M}_S, B_S)$  with the targeted blendshape  $S_i^M$ , is composed of two steps.

1. Both skull parts of  $B_S$ , cranium and jaw, are each directly positioned by independent rigid transformations  $\nabla B$  that are calculated between respective subsets of  $S$  and  $S_i^M$ . The subsets are visualized in Fig. 4 Cranium and Jaw.
2. Soft tissue and muscle tissue are deformed by minimizing the following energies that reflect anatomical properties. The energy for soft tissue is defined as

$$E_{\mathbb{S}}(\mathbb{S}_S) = \sum_{t \in \mathbb{S}_S} \left( \min_{\mathbf{R} \in SO(3)} \left\| \nabla(t, \mathbb{S}_S) - \mathbf{R} \right\|_F^2 + (\det(\nabla(t, \mathbb{S}_S)) - 1)^2 \right), \quad (11)$$

which for each soft tissue tetrahedron  $t$  penalizes changes in volume and strain. Here,  $\mathbf{R} \in SO(3)$  denotes the optimal rotation,  $\nabla(t, \mathbb{S}_S) \in \mathbb{R}^{3 \times 3}$  the deformation gradient of  $t$ , and  $\|\cdot\|_F$  the Frobenius norm.

For the muscle tetrahedra, only a volume-preservation term

$$E_{\mathbb{M}}(\mathbb{M}_S) = \sum_{t \in \mathbb{M}_S} (\det(\nabla(t, \mathbb{M}_S)) - 1)^2 \quad (12)$$

is applied to allow for muscle contractions.

Finally, the source head surface  $S \subset \mathbb{S}_S$  is drawn to the targeted missing blendshape  $S_i^M$  via

$$E_{\text{tar}}(\mathbb{S}_S, S_i^M) = \|S - S_i^M\|^2. \quad (13)$$

In total, we minimize the weighted energy

$$E_{\text{inv}}(\mathbb{S}_S, \mathbb{M}_S, S_i^M) = w_{\mathbb{S}} E_{\mathbb{S}}(\mathbb{S}_S) + w_{\mathbb{M}} E_{\mathbb{M}}(\mathbb{M}_S) + w_{\text{tar}} E_{\text{tar}}(\mathbb{S}_S, S_i^M) \quad (14)$$

with respect to the vertex positions of the soft and muscle tissue meshes  $\mathbb{S}_S, \mathbb{M}_S$  in the projective dynamics framework [43]. The values of all weights can be found in Table 3.

Paired with the rigid transformations of the skull  $\nabla B$ , the deformations caused by the simulation are passed on to the forward simulation  $\text{fwdSim}$  in the form of stacked per tetrahedron deformation gradients  $\nabla \mathbb{S}$  (soft tissue),  $\nabla \mathbb{M}$  (muscle tissue).

**Facial characteristics.** In correspondence to the facial characteristics described above,  $f_c$  is composed of three methods

$$f_c = (f_{c_{\text{eo}}}, f_{c_{\text{es}}}, f_{c_{\text{lc}}}) \quad (15)$$

which specify objectives for the eye opening ( $f_{c_{\text{eo}}}$ ), the eye sockets ( $f_{c_{\text{es}}}$ ), and the lip contour ( $f_{c_{\text{lc}}}$ ) in the forward simulation  $\text{fwdSim}$ .

To capture the eye characteristics with  $f_{c_{\text{eo}}}$  and  $f_{c_{\text{es}}}$ , we add supplementary triangles between the upper and lower eyelids (Eyelid) and between the upper and lower boundaries of the eye sockets (Eye Socket) as visualized in Fig. 4. Hereafter, we refer to these triangles as EO and ES, respectively. Since the eye characteristics are intended to transfer relative movements, we define them such that the scaling of the surface area of the previously added triangles is identical in both the targeted and the retargeted blendshape. More formally, the characteristic  $F_{\text{eo}} = f_{c_{\text{eo}}}(S_i^M, S, T)$  is a vector which contains the surface area of each EO triangle in  $T$ , scaled by the ratio of the corresponding triangle areas in the targeted  $S_i^M$  and the neutral  $S$ .  $F_{\text{es}} = f_{c_{\text{es}}}(S_i^M, S, T)$  is defined accordingly.

We define the characteristic of the lip contour  $F_{\text{lc}}$  on a set of vertices LC as visualized in Fig. 4 Lips. Here, we intend to transfer the vertex positions of the contour from the targeted  $S_i^M$  to the retargeted blendshape  $T_i^M$  as similar as possible. To that end, we first apply the original deformation transfer  $\text{dt}$  [7] to determine the coarse shape and position of the targeted lip contour in the retargeting result. Afterward, we correct  $\text{dt}$  by finding an optimal similarity mapping. Formally, we define

$$F_{\text{lc}} = f_{c_{\text{lc}}}(S_i^M, S, T) = s \mathbf{R} (S_i^M)^{\text{LC}} + \mathbf{t}, \quad (16)$$

where  $s \in \mathbb{R}$  (scaling),  $\mathbf{R} \in SO(3)$  (rotation),  $\mathbf{t} \in \mathbb{R}^3$  (translation) represent the optimal similarity mapping regarding

$$\min_{s, \mathbf{R}, \mathbf{t}} \left\| \text{dt}(S_i^M, S, T)^{\text{LC}} - s \mathbf{R} (S_i^M)^{\text{LC}} - \mathbf{t} \right\|^2 \quad (17)$$

and  $(\cdot)^{\text{LC}}$  selects the vertices of the lip contour.

**Forward simulation.** The forward simulation  $\text{fwdSim}$ , which applies the previously identified deformations ( $\nabla \mathbb{S}$ ,  $\nabla \mathbb{M}$ ,  $\nabla B$ ) and expression-specific facial characteristics  $F_i$  to the target head  $H_T = (\mathbb{S}_T, \mathbb{M}_T, B_T)$ , consists of three steps.

1. As for  $\text{invSim}$ , the skull  $B_T$  is directly positioned by applying the rigid transformations  $\nabla B$ . However, to align the range of motions, we scale the translational components by  $\frac{\text{BB}(T)}{\text{BB}(S)}$ , where BB calculates diameters of the respective bounding boxes.
2. The weighted energy

$$E_{\text{fwd}}(\mathbb{S}_T, \mathbb{M}_T, \nabla \mathbb{S}, \nabla \mathbb{M}, F_i) = w_{\nabla \mathbb{S}} E_{\nabla \mathbb{S}}(\mathbb{S}_T, \nabla \mathbb{S}) + w_{\nabla \mathbb{M}} E_{\nabla \mathbb{M}}(\mathbb{M}_T, \nabla \mathbb{M}) + w_{\text{F}} E_{\text{F}}(\mathbb{S}, F_i) \quad (18)$$

is minimized, which applies the deformation gradients  $\nabla \mathbb{S}, \nabla \mathbb{M}$  to the respective tissue while adhering to the facial characteristics  $F_i$ . All energies in Eq. (18) act similar to Eq. (13) and are formally defined in the Appendix. Again, we rely on projective dynamics for solving the minimization problem.

3. Finally, we resolve self-collisions between lips similar to Komaritzan et al. [44]. Here, each collided lower lip point and the closest upper lip point in vertical direction on the head surface are resolved to the average position of both. The average position is enforced in an additional run of the second step.

After both optimizations, the retargeted missing blendshape  $T_i^M \subset S_T$  can be extracted.

### 3.4. Anatomical plausibility

Based on the functions of `adt`, we can now implement `anacon`, the final physics-based simulation of AnaConDaR (Eq. (6)). By setting

$$\begin{aligned} T_i &= \text{anacon}(T_i^P + \hat{T}_i^M, S_i, S, T) \\ &= \text{fwdSim}(\text{invSim}(T_i^P + \hat{T}_i^M, H_T), F_i, H_T), \end{aligned} \quad (19)$$

the anatomical constraints involved in `invSim` (Section 3.3.2 Inverse Simulation) improve the anatomical plausibility of the combined retargeting  $T_i^P + \hat{T}_i^M$  while preventing visible patch boundaries. Moreover, expression-specific facial characteristics  $F_i = f_c(S_i, S, T)$  (Section 3.3.2 Facial Characteristics) derived from the targeted expression  $S_i$  are also reflected in the final AnaConDaR result  $T_i$ .

### 3.5. Target example embedding

Although all components are now specified, AnaConDaR is still unable to handle situations where the target examples  $\mathcal{T}_\mathcal{E}$  lack corresponding expressions in the source animation  $S$ , a common limitation of other data-driven facial retargeting approaches [1,2]. We remove this initial assumption (Section 3.2.1) by embedding the target examples in the source domain.

To that end, we first retarget  $\mathcal{T}_\mathcal{E}$  with `adt` to create an initial embedding

$$S_\mathcal{E}^{\text{adt}} = \{\text{adt}(T_j, T, S)\}_{T_j \in \mathcal{T}_\mathcal{E}}. \quad (20)$$

As `adt` is geometry-driven,  $S_\mathcal{E}^{\text{adt}}$  might still exhibit character-specific details of the target character. In a second step, we therefore exploit the observation that, in most cases, the source animation  $S$  is extensive and expressive in linear combinations.

More precisely, we reconstruct each  $S_i^{\text{adt}} \in S_\mathcal{E}^{\text{adt}}$  by solving linear least squares problems as in Eqs. (1)–(3). This time, however, all source expressions  $S_j \in S$  act as blendshapes. The resulting optimal blending weights  $\mathbf{w}_i^{\text{adt}} = (\dots, w_{ij}^{\text{adt}}, \dots)$  are then used to form the data-driven embedding

$$S_\mathcal{E} = \left\{ S + \sum_{S_j \in S} w_{ij}^{\text{adt}} (S_j - S) \right\}_{S_i^{\text{adt}} \in S_\mathcal{E}^{\text{adt}}}. \quad (21)$$

By construction,  $S_\mathcal{E}$  is fully embedded in the source domain and no longer includes details of the target character. The embedding process is also illustrated in step 1 of Fig. 2, which completes the visual overview of AnaConDaR.

## 4. Experiments

Before visually demonstrating AnaConDaR's capabilities for offline facial performance retargeting in Section 4.2, we discuss implementation details and runtimes in Section 4.1. Thereafter, in Section 4.3, a user study investigates the human perception of AnaConDaR in comparison to the most relevant peers. In Section 4.4, a quantitative analysis demonstrates the advantages of AnaConDaR over the state-of-the-art ALM [2] algorithm. However, we also elaborate on why quantitative evaluations only have limited meaningfulness for facial retargeting. Section 4.5 focuses on an extensive ablation study, while Section 4.6 showcases selected AnaConDaR features in more detail.

### 4.1. Implementation & runtimes

We implement all projective dynamics simulations with the CPU-based ShapeOp framework [45] and exploit parallelism wherever applicable. Table 2 gives the dimensions of all template components, and Table 3 states the weights of all experiments. All runtimes were determined on an AMD Ryzen Threadripper PRO 3995WX processor.

Overall, once the simulations are initialized ( $\approx 9$  s), AnaConDaR can be run at either approximately 10 fps (without collision resolving) or 0.3 fps (with collision resolving). There are many GPU-based solvers available (e.g., <http://suitesparse.com>) that can optimize the runtime in general. Collision resolution could also be accelerated, as most of the time spent on collision resolution is due to the refactorisation of the projective dynamics solver. In Wang et al. [46], for instance, an efficient alternative is proposed. However, as our focus has been on methodological improvements, not on inference speed, we leave computationally more efficient implementations as future work.

### 4.2. Qualitative evaluation

Fig. 5 and Fig. 6 display representative retargeting results of AnaConDaR. All shown 3D models are part of the commercial [3Dscanstore.com](http://3Dscanstore.com) database and have been acquired with a high-resolution optical multi-view scanner. We manually established a common topology using [faceform.com](http://faceform.com). The retargeted expressions are either facial movements like *cheek puffer* and *mouth stretch* or emotions like *sad*, *happy*, and *surprise*.

Fig. 5 displays results obtained from a  $\mathcal{T}_\mathcal{E}$  composed of only 5 target examples, whereas for the results from Fig. 6, an extensive set of 30 examples has been available. For each retargeting result, a different  $\mathcal{T}_\mathcal{E}$  has been randomly drawn. Please refer to the attached video for a demonstration of the temporal consistency of AnaConDaR.

#### 4.2.1. Peer group

We compare AnaConDaR to Example-Based Facial Rigging [6] (EBFR), Anatomical Local Models [2] (ALM), Deformation Transfer [9] (DT), Linear Blendshapes [1] (LBS), and our own Anatomical Deformation Transfer (ADT).<sup>1</sup> Generally, ALM and LBS require expressions in the source animation  $S$  that correspond to the target examples  $\mathcal{T}_\mathcal{E}$ . Therefore, we follow the suggestion by the authors of ALM to use EBFR as preprocessing if this requirement is not fulfilled.

#### 4.2.2. Discussion

The subsequent discussion of the presented outcomes follows along the structural varieties of all compared algorithms. For easier traceability of our analysis, Fig. 7 provides a visual overview.

- The DT implementation we investigate [9] is the most recent adaption to faces. Here, locally adapted delta transfers for predefined landmarks are additionally incorporated. Previous findings [2] already indicated minimal distinctions between delta transfer and DT. Our results consistently demonstrate that also this DT variant transfers character-specific details and not only deformations related to the targeted expressions.
- EBFR seeks a target animation  $\mathcal{T}$  such that a linear combination of  $\mathcal{T} \setminus \mathcal{T}_\mathcal{E}$  can approximate the target examples  $\mathcal{T}_\mathcal{E}$ . Since this is a strongly underdetermined optimization problem, the DT results are used for regularization. As a consequence, depending on the solver, either only a few expressions of  $\mathcal{T}$  contribute to the explanation of each example in  $\mathcal{T}_\mathcal{E}$  or all contribute only a small fraction to the explanation. In any case, this leads to only minor personalization beyond DT, especially when only 5 examples are available.

<sup>1</sup> We compare to our own implementations of the peer group.



Fig. 5. AnaConDaR in comparison to the state-of-the-art peer group EBFR [6], ALM [2], DT [9], LBS [1], and to our ADT. Furthermore, the source reconstruction after applying anatomical constraints (i.e., *anacon* w/o facial characteristics) is shown for a reasonable comparison. Plotted on the reconstruction is the PLBS reconstruction error (in centimeters). The difference between ADT and AnaConDaR is plotted on the ADT expression. All results have been achieved with **five randomly drawn examples** of the target character. Especially in this setting, with only a few target examples, AnaConDaR leads to considerable improvements.

- The state-of-the-art ALM approach [2] is closely related to LBS [1]. After establishing a corresponding set of source examples  $S_{\mathcal{E}}$  to the target examples  $\mathcal{T}_{\mathcal{E}}$  with EBFR, both form the target animation  $\mathcal{T}$  by blending  $\mathcal{T}_{\mathcal{E}}$ . The blending weights are found by rebuilding the source animation  $S$  with  $S_{\mathcal{E}}$ . ALM mainly differs from LBS in that the blending is conducted on small patches and not on complete meshes (please refer to Section 3.2.1 for more details).

In our experiments, LBS suffers from a strong bias which prevents an adequate reconstruction of source expressions, leading to the retargeting of *different* expressions. Put differently, the poor results primarily stem from the *missing* blendshape as described in Section 3.2.2. The PLBS of ALM significantly mitigate this issue, especially when having access to 30 target examples  $\mathcal{T}_{\mathcal{E}}$ . Nonetheless, notable reconstruction errors remain.



Fig. 6. The same experiment as in Fig. 5, however, 30 examples of the target character have been available. In principle, reconstruction errors decrease with more target examples, reducing the influence of the missing blendshape. Nevertheless, considerable benefits of AnaConDaR can also be recognized in this setting, as even with more examples, a complete reconstruction is not guaranteed. Moreover, other advantages, such as the fully volumetric simulation or the preservation of facial features, weigh in.

- Our AnaConDaR approach exhibits a high degree of personalization even with only a few examples from the target character and can still achieve more appealing results than ALM, LBS, EBFR, DT, or ADT if the number of examples is high. In contrast to DT, the data-driven AnaConDaR component abstains from transferring character-specific details wherever feasible. Unlike ALM and LBS, additionally retargeting the missing

blendshape ensures that AnaConDaR does not lose information due to informational gaps of the exemplary target data. Lastly, different from EBFR, in our approach the target examples  $\mathcal{T}_E$  explain each expression to be transferred in  $S$ , and not all expressions to be transferred explain the target examples. This strategy effectively avoids the *explanation problem* associated with EBFR, as discussed before.

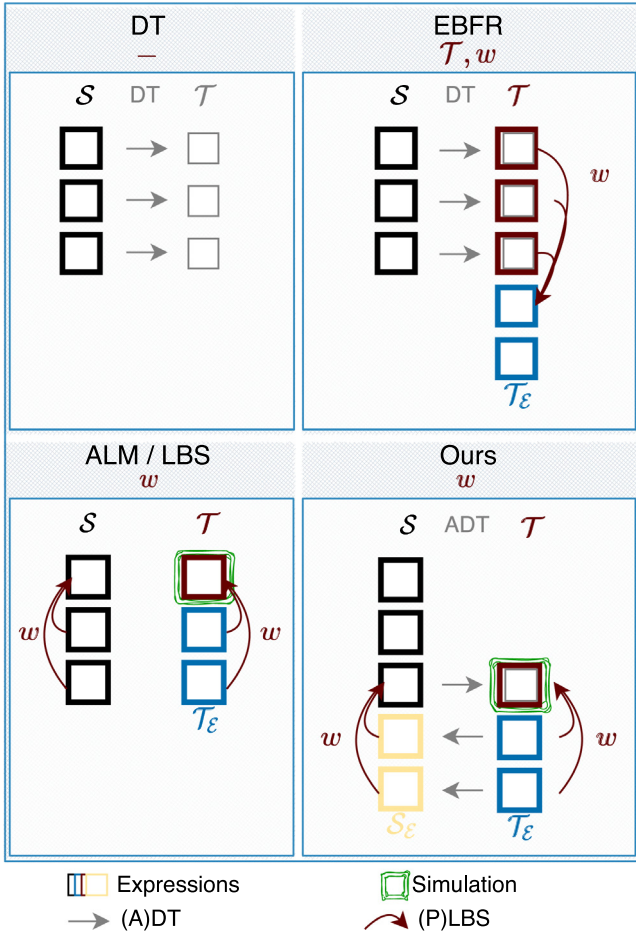


Fig. 7. A structural overview of AnaConDaR and other state-of-the-art facial retargeting approaches. Below each algorithm name, the variables under direct optimization are stated. Source and target expressions correspond only if depicted in the same row.

In summary, AnaConDaR achieves convincing visual results by compensating the conceptual weaknesses of other algorithms while adopting their respective advantages.

### 4.3. User study

In a user study, we presented the following task.

“Please rank the images according to how natural the transfer of the expression seems to you from best to worst.”

The study involves 15 randomly selected retargeting instances, with five each produced using 5, 15, and 30 examples of the target character. Participants ranked the results of AnaConDaR, DT, EBFR, and ALM, respectively. The design of the study is aligned with Chandran et al. [2], an illustration can be found in Appendix B. To ensure independent documentation, we used [survio.com](https://www.surveymonkey.com) for the technical implementation.

The outcome shown in Fig. 8 summarizes 33 responses by university members and computer graphics students from two universities who were not familiar with facial retargeting algorithms. We performed Wilcoxon Signed-Rank tests to inspect if the tendency of the AnaConDaR mean rank in comparison to the other peers is significant. We can confirm this hypothesis for all peers on a significance level of 0.05.

The user study emphasizes that AnaConDaR is perceived as a more natural facial retargeting. Nonetheless, perception variations are

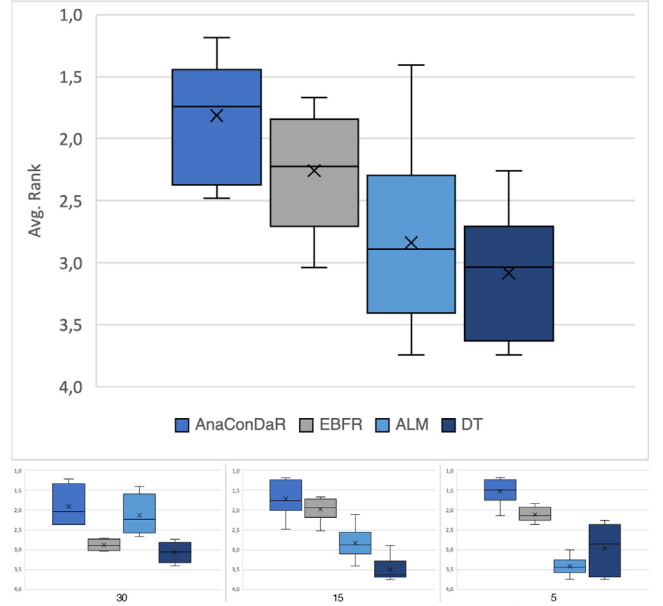


Fig. 8. A user study among university members and computer graphics students from two universities supports that AnaConDaR is perceived as a more natural facial retargeting. The combined results (top) as well as the results per number of target examples (bottom) are shown.

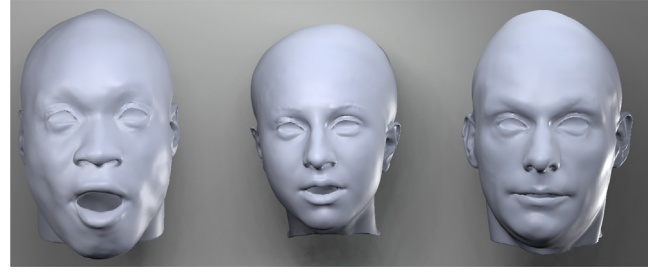


Fig. 9. An example of the diverse ways in which individuals interpret the same expression (here, *Surprise*). For more examples please refer to Wu et al. [40].

evident from the ranking variances shown in Fig. 8. Probably by construction, the *data-infused* EBFR outperforms the solely geometry-based DT. Interestingly, EBFR also outperforms ALM, while ALM exhibits the highest variance. This was to be expected to some extent, as ALM is the only method lacking a geometry-based component and its retargeting quality, therefore, heavily depends on the number of target examples. The latter observation is further supported by the separated representation of the user study in Fig. 8.

### 4.4. Quantitative evaluation

In previous work, quantitative evaluations have only been conducted in cherry-picked individual cases but not in empirically comprehensive investigations [2,6]. This is mostly due to ambiguities in facial expressions (see Fig. 9 and Wu et al. [40] for examples) as well as varying human perception. Our user study (Section 4.3) underscores the latter issue. Although the perceived qualities of individual retargeting methods differ significantly, the variances are not negligible. Sometimes cyclic errors, i.e., mapping from the source to the target and back, are considered as a suitable evaluation protocol. Nonetheless, they only validate how well geometric transformations are preserved in the cycle. By construction, deformation transfer [7] would be unsurpassed in this

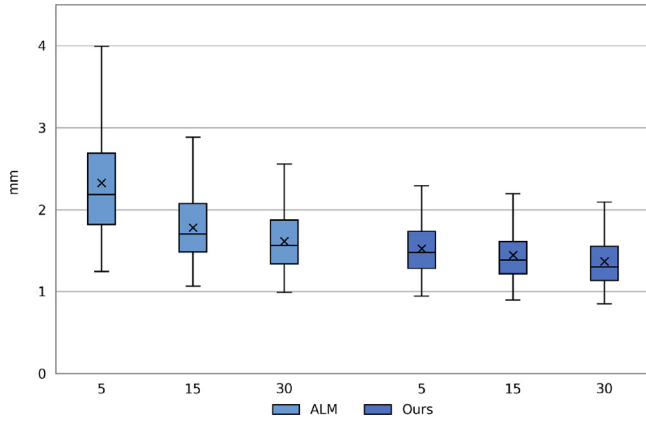


Fig. 10. A quantitative comparison of AnaConDaR against ALM [2] based on synthesized ground truth (Section 4.4). The results are grouped by the number of available target examples and reported in terms of the L2 error (mm). AnaConDaR outperforms ALM in each scenario.

Table 2

The dimensions of all template components in our experiments.

Mesh	$S$	$B$	$S$	$M$
# Vertices	29 826	14 572		61 875
# Faces/Tetrahedra	59 648	28 727	126 612	107 437

Table 3

The weights of the physics-based simulations and the PLBS reconstruction.

$w_{VS}$	$w_{VM}$	$w_F$	$w_S$	$w_M$	$w_{tar}$	$\lambda_{reg}$
1.0	1.0	10.0	1.0	1.0	100.0	0.01

evaluation, while the flaws of geometry-driven approaches are well known.

To nonetheless quantitatively compare AnaConDaR to ALM, we first synthesize an appropriate dataset. More precisely, we use EBFR [6] to create personalized ARKit<sup>2</sup> blendshape rigs for each identity of the 3Dscanstore.com database. Subsequently, we create the same 250 random facial expressions for all identities through linear blending of the ARKit rigs with blending weights recorded in dyadic conversations [47]. In the resulting dataset, corresponding facial expressions exhibit reduced ambiguities and, hence, can rather be regarded as ground truth.

Therefore, we conduct the following experiment on this dataset. To begin with, five source expressions  $S$  as well as either 5, 15, or 30 target examples  $T_e$  are randomly drawn for all source-target identity combinations. Afterward, we run AnaConDaR and ALM for each source expression and measure the average of the vertex-wise L2 differences to the ground truth in mm. The findings of this experiment, reported in Fig. 10, indicate that AnaConDaR outperforms ALM, especially when only a few target examples are available. A moderate improvement can still be recognized when many target examples are available. This quantitative evaluation ignores human perception but is nonetheless consistent with the previously discussed user study (Section 4.3).

#### 4.5. Ablation study

We examine the main components of AnaConDaR in another user study, which is summarized in Fig. 11 and visualized in Fig. 12. Particularly, we compare the regular AnaConDaR to AnaConDaR without expression-specific facial characteristics, without the missing blendshape, and with the standard deformation transfer dt [7] instead of our adt. The design of the user study is mostly as described in Section 4.3.

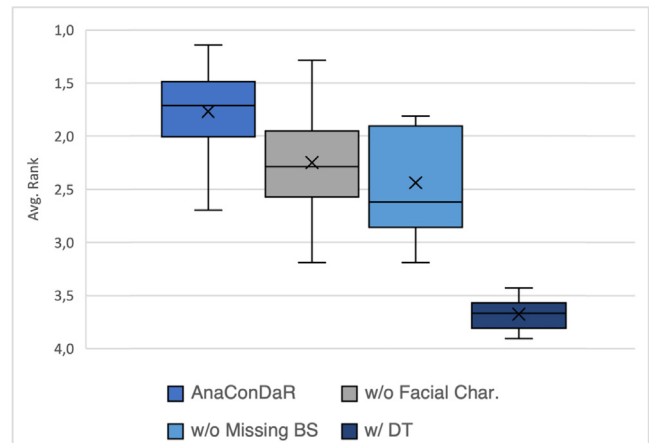


Fig. 11. A user study among university members and computer graphics students from two universities proves the benefits of each AnaConDaR component.

However, no similar example has been provided to the 29 participants. Please refer to Appendix B for an exemplary question from this study.

**Deformation transfer.** The most noticeable visual differences arise in the setting in which dt is used rather than adt. Here, the artifacts caused by PLBS patch boundaries are transferred by dt, and the strain constraint in anacon does not provide a sufficient countermeasure. An increased strain weight  $w_S$  could potentially compensate for this but would also remove high-frequency details. Since adt, unlike dt, also applies anatomical constraints, similar artifacts do not occur in the regular AnaConDaR results. The user study supports this visual observation as the dt variant is ranked last.

Instead of an amplified strain, another option would be to eliminate the patch boundaries directly in the source estimation  $S_i^P$  before calculating the missing blendshape  $S_i^M$ . For this, there are at least two obvious solutions. The first solution is to set up anatomical models as described in ALM [2,40]. However, this adds considerable unnecessary complexity, mainly due to additional optimization steps. Since these models only use data-driven anatomical surface constraints, they also cannot be used as an alternative to anacon. Particularly, they are not applicable to unseen expressions, cannot enforce facial characteristics, and cannot resolve collisions. The second solution is to apply anacon to the source estimation  $S_i^P$ . Essentially, this means applying the same physics-based simulations as in adt to a different input. Nevertheless, we decided to favor adt for theoretical reasons. Particularly, as adt applies anatomical constraints and expression-specific facial characteristics during the retargeting and not before. Neither of the two variants was visually superior in our experiments.

**Facial characteristics & missing blendshape.** The AnaConDaR modifications without facial characteristics and missing blendshape demonstrate that both components are essential, although their importance varies depending on the retargeting scenario. For instance, in the first row of Fig. 12, the expression-specific facial characteristics are especially important, whereas in the second row, the missing blendshape has a strong impact. Again, the user study confirms this visual observation, in which AnaConDaR is ranked ahead of both modifications.

The influence of the facial characteristics and the missing blendshape can also be observed in Fig. 13, in which each retargeting is performed once with 5 and once with 30 target examples. Although the relevance of both components is most evident when only a few target examples are available, the effects of both are still not negligible, even when there are many available target examples.

<sup>2</sup> <https://developer.apple.com/augmented-reality/arkit/>.



Fig. 12. A visual ablation study that illustrates the individual components of AnaConDaR. In particular, the effects of enforcing expression-specific facial characteristics, adding the missing blendshape, and using ADT over DT become apparent. Additionally, the PLBS result and the missing blendshape are depicted. Please note, that we show the PLBS results after applying anatomical constraints (i.e., anacon w/o facial characteristics) for a reasonable comparison.



Fig. 13. AnaConDaR retargetings with 5 and 30 available target examples. For each instance, the PLBS component (after imposing anatomical constraints) and the missing blendshape are shown.



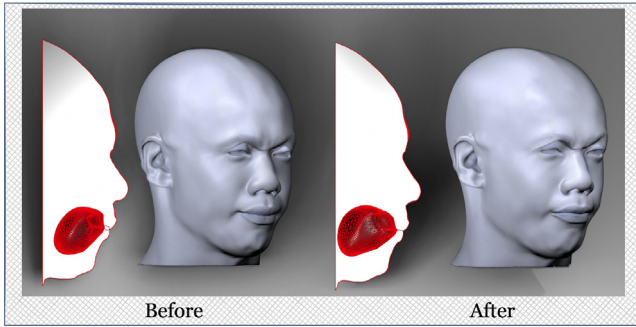


Fig. 14. An example of our method for resolving collisions. The lips get disentangled and the arising forces propagate through the soft tissue.

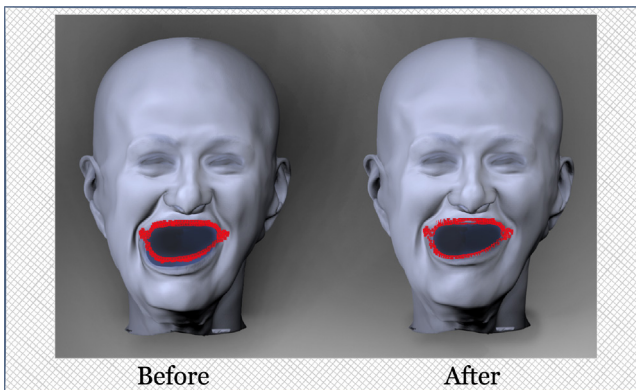


Fig. 15. An example of an artistic intervention into anacon. The targeted contour (red) is realized and the surrounding tissue is moved appropriately.

#### 4.6. Collisions and artistic control

In this paragraph, we will briefly highlight two features that become feasible through the physics-based simulations involved in AnaConDaR.

First, Fig. 14 displays our approach to resolving lip collisions. Not only do the upper and lower lip get disentangled, but the final volumetric simulation anacon of AnaConDaR (Section 3.4) propagates the displacements through the soft tissue.

Second, Fig. 15 shows an example of artistic intervention into anacon. To that end, we manually modify vertices of the lip contour and add corresponding soft Dirichlet constraints to the forward component fwdSim. For streamlining the process, we move only a few control points and govern the remaining lip contour points through an RBF space warp [42].

This example only serves as one illustration of applicable artistic interventions. For instance, material properties, the weight of a character, or external forces, like varying gravity directions [47], can also be manipulated. Furthermore, artistic interventions into the patchwise blending weights of PLBS are inherited from ALM. For a more detailed description, please refer to [2].

#### 5. Limitations

We assume that no global rigid motion occurs in facial expressions. Effective methods to achieve this prerequisite are available [48]. Moreover, AnaConDaR requires a shared mesh/patch topology of all source and target expressions. If this is not the case, a mapping can be found with unsupervised approaches [49,50] or manually, for instance, using [faceform.com](https://faceform.com). Concerning the physics-based simulations, we focus on the projective dynamics simulator [43] and do not add dynamic effects to obtain temporal independence. We chose projective dynamics because of its simplicity and sufficient efficiency, but other simulators can be used as drop-in replacements. Finally, we only handle self-collisions of the lips, while lip-teeth and eyelid collisions might also occur.

#### 6. Conclusion

In this work, we introduced AnaConDaR, a method that integrates data-driven and geometry-driven facial retargeting algorithms. More precisely, the geometry-driven approach bridges informational gaps resulting from insufficiently expressive target examples within the data-driven approach. As a result, we enhance the current state-of-the-art ALM [2] to attain superior retargeting outcomes, particularly in situations where only a minimal number of target examples is available.

Due to the usage of patchwise linear blendshapes and the volumetric head representation, the user can readily guide and tailor AnaConDaR. The presented visually convincing qualitative examples of our approach are supported by two user studies and a quantitative analysis.

Promising future directions for improving AnaConDaR are to employ even more anatomically precise physics-based simulations and fully volumetric blendshapes [51]. Also, a more in-depth user study that queries rationales may facilitate targeted improvements. Finally, an accelerated GPU implementation of AnaConDaR could potentially achieve real-time operating speeds.

#### CRedit authorship contribution statement

**Nicolas Wagner:** Writing – original draft, Methodology, Data curation, Conceptualization. **Ulrich Schwanecke:** Writing – review & editing, Supervision. **Mario Botsch:** Writing – review & editing, Supervision, Project administration.

#### Data availability

The authors do not have permission to share data.

#### Acknowledgment

We want to thank the German Federal Ministry of Education and Research (BMBF) that supported this research through the project HiAvA (ID 16SV8785). We also want to express our appreciation to the reviewers who have improved this work with their helpful comments.

#### Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Nicolas Wagner reports financial support was provided by German Federal Ministry of Education and Research (BMBF). If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Appendix A. Energies

In the following, we formally state the individual energies of the forward simulation simFwd (Eq. (18)).

##### A.1. Facial characteristics

The energy for the facial characteristics

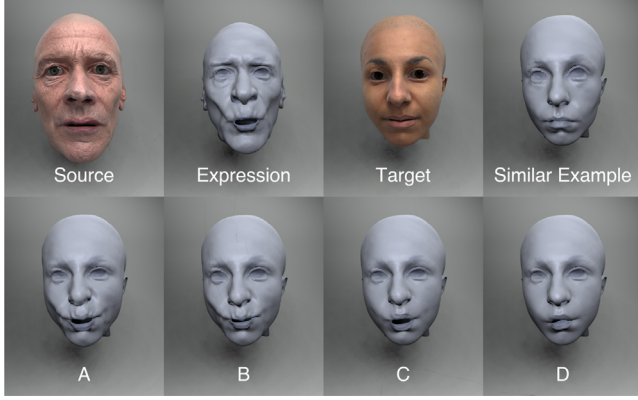
$$E_F(\mathbb{S}_T, F_i) = E_{eo}(\mathbb{S}_T, F_{eo}) + E_{es}(\mathbb{S}_T, F_{es}) + E_{lc}(\mathbb{S}_T, F_{lc}) \quad (A.1)$$

is composed of terms for the eye openings, the eye sockets, and the lip contour.

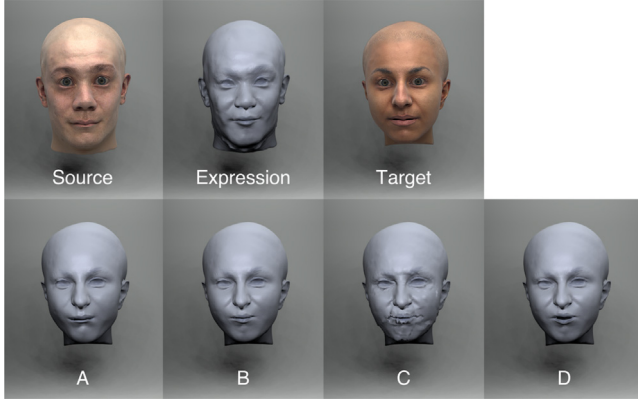
The energy for eye openings

$$E_{eo}(\mathbb{S}_T, F_{eo}) = \sum_{f \in EO} (A(\mathbb{S}_T, f) - a_{eo}(f))^2 \quad (A.2)$$

penalizes for each triangular face  $f \in EO$  deviations in the surface area  $A(\mathbb{S}_T, f)$  from the corresponding targeted surface area  $a_{eo}(f) \in F_{eo}$ .



**Fig. B.16.** An instance of the user study, wherein 33 participants ranked the peer group of AnaConDaR. Consistent with the user study conducted by Chandran et al. [2], a real target example supported the participants in ranking. In this illustration, A-D are the results of EBFR, DT, AnaConDaR, and ALM. Generally, the results were placed in a random order.



**Fig. B.17.** An instance of the user study, wherein 29 participants ranked individual components of AnaConDaR. In this illustration, A-D are the results of AnaConDaR without the missing blendshape, AnaConDaR, AnaConDaR with DT, and AnaConDaR without facial features. Generally, the components were placed in a random order.

The energy for the eye sockets

$$E_{\text{es}}(\mathbb{S}_T, F_{\text{es}}) = \sum_{f \in \text{ES}} (A(f) - a_{\text{es}}(f))^2 \quad (\text{A.3})$$

penalizes for each triangular face  $f \in \text{ES}$  deviations in the surface area  $A(\mathbb{S}_T, f)$  from the corresponding targeted surface area  $a_{\text{es}}(f) \in F_{\text{es}}$ .

The energy for the lip contours

$$E_{\text{lc}}(\mathbb{S}_T, F_{\text{lc}}) = \left\| (\mathbb{S}_T)^{\text{LC}} - F_{\text{lc}} \right\|^2 \quad (\text{A.4})$$

draws the vertices  $(\mathbb{S}_T)^{\text{LC}} \in \mathbb{S}_T$  to the corresponding vertices  $F_{\text{lc}}$ .

### A.2. Tissue deformations

The energy for the soft tissue

$$E_{\text{vs}}(\mathbb{S}_T, \nabla \mathbb{S}) = \sum_{t \in \mathbb{S}_T} \left\| \nabla(t, \mathbb{S}_T) - \text{DG}_{\nabla \mathbb{S}}(t) \right\|_F^2 \quad (\text{A.5})$$

penalizes for each tetrahedron  $t \in \mathbb{S}_T$  deviations from the deformation gradient  $\nabla(t, \mathbb{S})$  to the corresponding targeted deformation gradient  $\text{DG}_{\nabla \mathbb{S}}(t) \in \nabla \mathbb{S}$ .

The energy for the muscle tissue

$$E_{\text{vm}}(\mathbb{M}_T, \nabla \mathbb{M}) = \sum_{t \in \mathbb{M}_T} \left\| \nabla(t, \mathbb{M}_T) - \text{DG}_{\nabla \mathbb{M}}(t) \right\|_F^2 \quad (\text{A.6})$$

penalizes for each tetrahedron  $t \in \mathbb{M}_T$  deviations from the deformation gradient  $\nabla(t, \mathbb{M})$  to the corresponding targeted deformation gradient  $\text{DG}_{\nabla \mathbb{M}}(t) \in \nabla \mathbb{M}_T$ .

### Appendix B. User studies

See Figs. B.16 and B.17 for examples of the user studies.

### Appendix C. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.cag.2024.103988>.

### References

- [1] Lewis JP, Anjyo K, Rhee T, Zhang M, Pighin FH, Deng Z. Practice and theory of blendshape facial models. *Eurographics (State Art Rep)* 2014;1(8):2.
- [2] Chandran P, Ciccone L, Gross M, Bradley D. Local anatomically-constrained facial performance retargeting. *ACM Trans Graph (ToG)* 2022;41(4):1–14.
- [3] Yang L, Zoss G, Chandran P, Gotardo P, Gross M, Solenthaler B, Sifakis D. An implicit physical face model driven by expression and style-supplemental. 2023.
- [4] Zielonka W, Bolkart T, Thies J. Instant volumetric head avatars. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, p. 4574–84.
- [5] Qian S, Kirschstein T, Schoneveld L, Davoli D, Giebenhain S, Nießner M. GaussianAvatars: Photorealistic head avatars with rigged 3D Gaussians. 2023, arXiv preprint arXiv:2312.02069.
- [6] Li H, Weise T, Pauly M. Example-based facial rigging. *Acm Trans Graph (ToG)* 2010;29(4):1–6.
- [7] Sumner RW, Popović J. Deformation transfer for triangle meshes. *Acm Trans Graph (ToG)* 2004;23(3):399–405.
- [8] Botsch M, Sumner R, Pauly M, Gross M. Deformation transfer for detail-preserving surface editing. In: *Vision, modeling & visualization*. Citeseer; 2006, p. 357–64.
- [9] Onizuka H, Thomas D, Uchiyama H, Taniguchi R-i. Landmark-guided deformation transfer of template facial expressions for automatic generation of avatar blendshapes. In: *Proceedings of the IEEE/CVF international conference on computer vision workshops*. 2019.
- [10] Chen R, Chen X, Ni B, Ge Y. Simswap: An efficient framework for high fidelity face swapping. In: *Proceedings of the 28th ACM international conference on multimedia*. 2020, p. 2003–11.
- [11] Garrido P, Valgaerts L, Rehmsen O, Thormahlen T, Perez P, Theobalt C. Automatic face reenactment. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, p. 4217–24.
- [12] Kim H, Elgharib M, Zollhöfer M, Seidel H-P, Beeler T, Richardt C, Theobalt C. Neural style-preserving visual dubbing. *Acm Trans Graph (ToG)* 2019;38(6):1–13.
- [13] Nirkin Y, Keller Y, Hassner T. Fsgan: Subject agnostic face swapping and reenactment. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, p. 7184–93.
- [14] Perov I, Gao D, Chervoniy N, Liu K, Marangonda S, Umé C, Dpfs M, Facenheim CS, RP L, Jiang J, et al. DeepFaceLab: Integrated, flexible and extensible face-swapping framework. 2020, arXiv preprint arXiv:2005.05535.
- [15] Ren Y, Li G, Chen Y, Li TH, Liu S. Pirenderer: Controllable portrait image generation via semantic neural rendering. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, p. 13759–68.
- [16] Wang Y, Chen X, Zhu J, Chu W, Tai Y, Wang C, Li J, Wu Y, Huang F, Ji R. Hiface: 3d shape and semantic prior guided high fidelity face swapping. 2021, arXiv preprint arXiv:2106.09965.
- [17] Zhang J, Zeng X, Wang M, Pan Y, Liu L, Liu Y, Ding Y, Fan C. Freenet: Multi-identity face reenactment. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, p. 5326–35.
- [18] Moser L, Chien C, Williams M, Serra J, Hendler D, Roble D. Semi-supervised video-driven facial animation transfer for production. *Acm Trans Graph (ToG)* 2021;40(6):1–18.
- [19] Hong Y, Peng B, Xiao H, Liu L, Zhang J. Headnerf: A real-time nerf-based parametric head model. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, p. 20374–84.
- [20] Bouaziz S, Wang Y, Pauly M. Online modeling for real-time facial animation. *Acm Trans Graph (ToG)* 2013;32(4):1–10.
- [21] Chen L, Cao C, De la Torre F, Saragih J, Xu C, Sheikh Y. High-fidelity face tracking for ar/vr via deep lighting adaptation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, p. 13059–69.
- [22] Lombardi S, Saragih J, Simon T, Sheikh Y. Deep appearance models for face rendering. *Acm Trans Graph (ToG)* 2018;37(4):1–13.
- [23] Cao C, Simon T, Kim JK, Schwartz G, Zollhoefer M, Saito S-S, Lombardi S, Wei S-E, Belko D, Yu S-I, et al. Authentic volumetric avatars from a phone scan. *Acm Trans Graph (ToG)* 2022;41(4):1–19.

- [24] Garbin SJ, Kowalski M, Estellers V, Szymanowicz S, Rezaeifar S, Shen J, Johnson M, Valentin J. VolTeMorph: Realtime, controllable and generalisable animation of volumetric representations. 2022, arXiv preprint arXiv:2208.00949.
- [25] Choi B, Eom H, Mouscadet B, Cullingford S, Ma K, Gassel S, Kim S, Moffat A, Maier M, Revelant M, et al. Anatomomy: an animator-centric, anatomically inspired system for 3D facial modeling, animation and transfer. In: SIGGRAPH Asia 2022 conference papers. 2022, p. 1–9.
- [26] Yang L, Kim B, Zoss G, Gözcü B, Gross M, Solenthaler B. Implicit neural representation for physics-driven actuated soft bodies. *Acm Trans Graph (ToG)* 2022;41(4):1–10.
- [27] Kim S, Jung S, Seo K, i Ribera RB, Noh J. Deep learning-based unsupervised human facial retargeting. In: *Computer graphics forum*. vol. 40, Wiley Online Library; 2021, p. 45–55.
- [28] Feng Y, Feng H, Black MJ, Bolkart T. Learning an animatable detailed 3D face model from in-the-wild images. *Acm Trans Graph (ToG)* 2021;40(4):1–13.
- [29] Li T, Bolkart T, Black MJ, Li H, Romero J. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics* 2017;36(6):1–17.
- [30] Chandran P, Bradley D, Gross M, Beeler T. Semantic deep face models. In: 2020 international conference on 3D vision. 3DV, IEEE; 2020, p. 345–54.
- [31] Zhang J, Chen K, Zheng J. Facial expression retargeting from human to avatar made easy. *IEEE Trans Vis Comput Graphics* 2020;28(2):1274–87.
- [32] Yang H, Zhu H, Wang Y, Huang M, Shen Q, Yang R, Cao X. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, p. 601–10.
- [33] Li J, Kuang Z, Zhao Y, He M, Bladin K, Li H. Dynamic facial asset and rig generation from a single scan. *ACM Transactions on Graphics* 2020;39:1–18.
- [34] Kim PH, Seol Y, Song J, Noh J. Facial retargeting by adding supplemental blendshapes. In: *PG (short papers)*. 2011.
- [35] Song J, Choi B, Seol Y, Noh J. Characteristic facial retargeting. *Comput Animat Virtual Worlds* 2011;22(2–3):187–94.
- [36] Ribera RBI, Zell E, Lewis JP, Noh J, Botsch M. Facial retargeting with automatic range of motion alignment. *Acm Trans Graph (ToG)* 2017;36(4):1–12.
- [37] Xu F, Chai J, Liu Y, Tong X. Controllable high-fidelity facial performance transfer. *Acm Trans Graph (ToG)* 2014;33(4):1–11.
- [38] Bhat KS, Goldenthal R, Ye Y, Mallet R, Koperwas M. High fidelity facial animation capture and retargeting with contours. In: *Proceedings of the 12th ACM SIGGRAPH/eurographics symposium on computer animation*. 2013, p. 7–14.
- [39] Karypis G, Kumar V. METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. 1997.
- [40] Wu C, Bradley D, Gross M, Beeler T. An anatomically-constrained local deformation model for monocular face capture. *Acm Trans Graph (ToG)* 2016;35(4):1–12.
- [41] Achenbach J, Brylka R, Gietzen T, zum Hebel K, Schömer E, Schulze R, Botsch M, Schwanecke U. A multilinear model for bidirectional craniofacial reconstruction. In: *Proceedings of the eurographics workshop on visual computing for biology and medicine*. 2018, p. 67–76.
- [42] Botsch M, Kobbelt L. Real-time shape editing using radial basis functions. In: *Computer graphics forum*. vol. 24, Blackwell Publishing, Inc Oxford, UK and Boston, USA; 2005, p. 611–21.
- [43] Bouaziz S, Martin S, Liu T, Kavan L, Pauly M. Projective dynamics: Fusing constraint projections for fast simulation. *Acm Trans Graph (ToG)* 2014;33(4):1–11.
- [44] Komaritzan M, Botsch M. Projective skinning. *Proc ACM Comput Graph Interact Tech* 2018;1(1):1–19.
- [45] Deuss M, Deleuran AH, Bouaziz S, Deng B, Piker D, Pauly M. ShapeOp—a robust and extensible geometric modelling paradigm. In: *Modelling behaviour*. Springer; 2015, p. 505–15.
- [46] Wang Q, Tao Y, Brandt E, Cutting C, Sifakis E. Optimized processing of localized collisions in projective dynamics. In: *Computer graphics forum*. vol. 40, Wiley Online Library; 2021, p. 382–93.
- [47] Wagner N, Botsch M, Schwanecke U. Softdeca: Computationally efficient physics-based facial animations. In: *Proceedings of the 16th ACM SIGGRAPH conference on motion, interaction and games*. 2023, p. 1–11.
- [48] Beeler T, Bradley D. Rigid stabilization of facial expressions. *Acm Trans Graph (ToG)* 2014;33(4):1–9.
- [49] Achenbach J, Zell E, Botsch M. Accurate face reconstruction through anisotropic fitting and eye correction. In: *VMV*. 2015, p. 1–8.
- [50] Schmidt P, Pieper D, Kobbelt L. Surface maps via adaptive triangulations. In: *Computer graphics forum*. vol. 42, Wiley Online Library; 2023, p. 103–17.
- [51] Ichim AE, Kavan L, Nimier-David M, Pauly M. Building and animating user-specific volumetric face rigs. In: *Symposium on computer animation*. 2016, p. 107–17.



## Citation

**NePHIM: A Neural Physics-Based Head-Hand Interaction Model**

Nicolas Wagner, Ulrich Schwanecke, and Mario Botsch

Computer Graphics Forum 44, 2025

DOI: [10.1111/cgf.70045](https://doi.org/10.1111/cgf.70045)

## 6.1 METHOD SUMMARY

In the publication *NePHIM*, we responded to Research Question 4 (Section 1.2.4).

*NePHIM* aims at the efficient animation of head–hand interactions and uses a concept similar to that of *SoftDECA* (Chapter 3), albeit implemented in a completely different manner. Here, too, an efficient neural network is employed to approximate a heuristically defined interaction physics–based simulation (PBS). However, training such a network presents different challenges than for our previous contributions, primarily due to the lack of an extensive data foundation like provided by *DECA* [32]. Moreover, the interaction simulation is inherently more complex, as long–term dependencies and friction must be considered to animate push and pull interactions authentically. The simulation requirements, in turn, necessitate the neural approximation to account for long–term dependencies, too, something for which the hypernetworks [15] from *(Sparse)SoftDECA* are not equipped. As in the previous chapters, the most important components of *NePHIM* are described in more detail below.

## PHYSICS-BASED SIMULATION

The *NePHIM* PBS is again based on *projective dynamics* [14] and relies on similar constraints that establish physical and anatomical plausibility

as in our previous works. The actual simulation of head–hand interactions stems from novel heuristics we introduce for both push and pull contacts that recognize collisions, respect temporal dependencies, and render sophisticated friction calculations unnecessary.

#### NETWORK

*NePHIM* relies on the well-established method *subspace neural physics* (*SNP*) [39] to accelerate the previously described simulation through a neural approximation. *SNP* first maps simulated heads and hands into respective compact subspaces and learns the simulation within these. This reduction of complexity enables a small and therefore very efficient neural network to be employed, which can effectively account for long-term dependencies due to a clever training procedure.

#### TRAINING DATA

To generate training data for *NePHIM*'s neural network, we assembled a complex multi-view rig that operates 16 video cameras synchronously to capture a person performing exemplary head–hand interactions. Established multi-view reconstruction methods subsequently transform the video recordings into per-frame tracked head and hand surface meshes. These meshes serve as the training input. Ground truth is obtained from the simulation of the tracked meshes, whereby the temporal sequence of the recordings is preserved. Our dataset comprises up to ten sequences for each of eight individuals. All sequences last about 30 seconds with 20 frames-per-second.

## 6.2 DISCUSSION

#### RESULTS

The evaluation of *NePHIM* mainly centers around the perception of the simulated head–hand interactions. In the PBSs of our previous works, we were able to rely on state-of-the-art components that had been tried and tested several times in real-world applications. However, for our new interaction heuristics, it was uncertain if these components would produce authentic animations. Consequently, we conducted an extensive user study involving approximately 50 participants to evaluate the realism of

our approach compared to related work and non-simulated interactions. For every example shown in the user study, *NepHIM* was judged as significantly more “natural” than its peers. We support these findings with visual examples that evidently demonstrate the extensive range of interactions *NepHIM* can animate. For instance, it is able to simulate convoluted interactions such as pulling the nose, pushing down a lip with one finger, or lifting the cheeks with flat hands. Moreover, a visual ablation study verifies the benefits of the individual aspects of the newly developed simulation heuristics.

The difficult nature of interaction handling and long-term temporal dependencies also raise doubts about the ability of the fast yet relatively simple *SNP* approach to learn our simulation sufficiently. To address these concerns, we conducted a range of quantitative analyses. Our observations indicate that *SNP* can approximate our simulation with an accuracy of less than one millimeter on test data, irrespective of whether the training involves a single individual or all eight from our recordings. At the same time, the approximation is remarkably efficient and can be executed on consumer-grade CPUs and GPUs with 50 and 200 frames-per-second, respectively.

Although *SNP* is fast and effective, it comes with the disadvantage that it cannot be controlled intuitively in the underlying latent spaces. In an ideal world, *NePHIM* should be steerable via tracked 2D landmarks from a single camera. To that end, we conducted another experiment in which we train a small multilayer perceptron to map 2D facial and hand landmarks [126] captured by the most frontal camera of our multi-view rig into the latent spaces of *SNP*. Our findings reveal that while the approximation accuracy remains nearly unaffected, the runtimes are slightly slower. However, this is primarily due to the tracking of 2D landmarks on images rather than the newly added multilayer perceptron itself. Nonetheless, we still achieve a performance of 66 frames-per-second on the GPU.

#### LIMITATIONS

In general, we successfully address the associated research question with *NePHIM*, although some limitations persist across various aspects. For instance, despite our PBS being convincing in the user study, the anatomical model we employ limits realism. Above all, we do not specifically model ear or nose cartilages, making these components appear overly flexible

during interactions, and neglect self-collisions of the lips or the lips with the teeth.

Another apparent limitation is the generalization capability of *NePHIM*, which was not our focus but is desirable for ease of use. In general, we can learn a personalized *SNP* network for any individual within five hours on a high-end workstation, if they are willing to be scanned for a few minutes in our 3D scanner. To train *NePHIM* to be more broadly applicable, the key challenge lies in recruiting an adequate number of participants willing to undergo the scanning procedures. However, the data processing of so many participants would introduce additional issues. For instance, recording 30 seconds in our scanner already generates approximately 100GB of data.

The last limitation we want to name extends beyond the scope of this thesis. While we can realistically simulate many different kinds of head-hand interactions, these simulations fundamentally rely on accurate tracking of heads and hands. To that end, we rely on the publicly available *Mediapipe* models [126], which generally exhibit high accuracy. However, they occasionally produce significant discrepancies depending on the viewing angle and output erroneous predictions due to occlusions. Particularly, errors can occur in the multi-view 3D reconstruction step and then lead to contaminated training data. Errors can also manifest when deploying the trained models in production applications. We safeguard *NePHIM* against tracking errors, for example, by not resolving interaction collisions when a hand penetrates too deeply into the skull. Thereby, the rigidity of the skull is maintained, yielding more realistic animations.

#### RELATED WORK

As mentioned in Chapter 2, there is only one previous work in the research field of *NePHIM*. This method, labeled *Decaf* [101], relies on a strongly simplified *position-based dynamics* [78] simulation of the face surface rather than on a volumetric representation of the head anatomy. It also does not account for long-term dependencies, limiting its ability to represent authentic interactions, and entirely neglects skin-pulling. Moreover, *Decaf* takes several seconds to simulate one frame and is designed for only a few predetermined interactions. For our dataset, we did not instruct participants on how to interact with the head. *Decaf* holds an advantage over our approach as it generalizes across the *FLAME* head model [61] and, hence, is more broadly applicable. Nevertheless, the question can be



raised as to what extent this is desirable, as *FLAME* typically depicts facial expressions with strong smoothness and minimal detail compared to more involved head models like *DECA* [32]. In *NePHIM*, we represent facial expressions through blendshape rigs of significantly higher quality.

Concurrently with *NePHIM*, *DICE* [120] was developed. However, *DICE* builds upon the *Decaf* simulation and complements our work as it solely improves on the visual tracking of interactions.




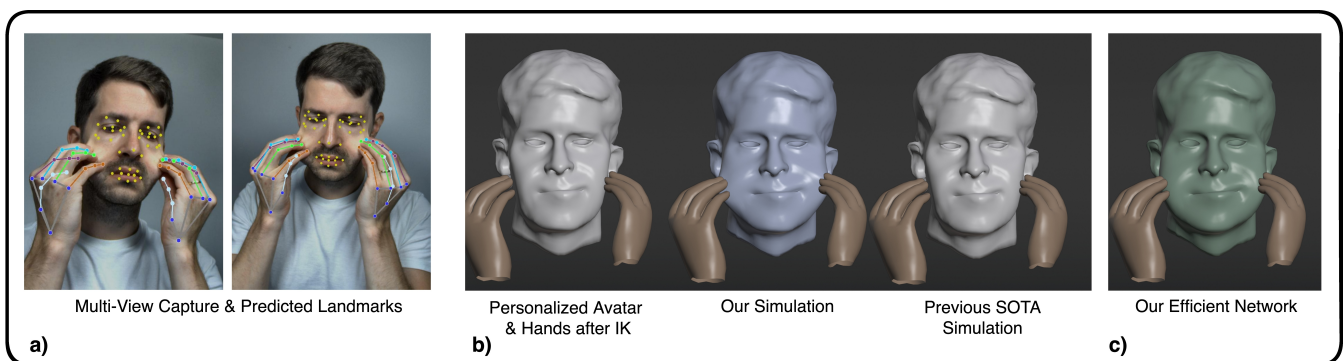
#### FUTURE WORK

We can think of at least two directions for further developing *NePHIM*. First, adapting the trend of photorealistic *Gaussian splatting* [51] – which is currently driving numerous innovations and improvements in facial animations [90, 70] – could benefit our use case. For instance, similar to *Gaussian Avatars* [90], attaching Gaussian splats to the simulated head meshes and training them with a photogrammetric loss might correct oversimplified or incorrect simulation assumptions when rendering.

The second direction is to improve the *NePHIM* PBS. Although enhancing the interaction heuristics, the anatomical representation, or the material parameters could yield better outcomes, manually modeling simulations of intricate natural processes remains challenging. Therefore, recent efforts in facial animation have focused on learning such simulations using deep learning [124, 123]. Unfortunately, an extensive and detailed ground truth dataset is a prerequisite for these approaches. As always, creating such a dataset requires enormous manual effort to clean up recorded 3D multi-view reconstructions and to convert them into standardized hand and head meshes.

#### 6.3 PUBLICATION

# NePHIM: A Neural Physics-Based Head-Hand Interaction Model

Nicolas Wagner<sup>1,2</sup> , Ulrich Schwanecke<sup>2</sup> , and Mario Botsch<sup>1,3</sup> <sup>1</sup>TU Dortmund University, Germany<sup>2</sup>RheinMain University of Applied Sciences, Germany<sup>3</sup>Lamarr Institute for Machine Learning and Artificial Intelligence, Germany

**Figure 1:** The different steps of NePHIM by means of a single frame: a) Two of the 16 views of our multi-camera rig used to capture head-hand interactions and corresponding landmarks. b) Our proposed simulation for head-hand interactions in comparison to the tracked input (after fitting template surfaces to the landmarks (IK)) and the simulation used in the previous state-of-the-art [SGPT23]. c) Prediction of the efficient neural network trained to approximate our simulation.

## Abstract

Due to the increasing use of virtual avatars, the animation of head-hand interactions has recently gained attention. To this end, we present a novel volumetric and physics-based interaction simulation. In contrast to previous work, our simulation incorporates temporal effects such as collision paths, respects anatomical constraints, and can detect and simulate skin pulling. As a result, we can achieve more natural-looking interaction animations and take a step towards greater realism. However, like most complex and computationally expensive simulations, ours is not real-time capable even on high-end machines. Therefore, we train small and efficient neural networks as accurate approximations that achieve about 200 FPS on consumer GPUs, about 50 FPS on CPUs, and are learned in less than four hours for one person. In general, our focus is not to generalize the approximation networks to low-resolution head models but to adapt them to more detailed personalized avatars. Nevertheless, we show that these networks can learn to approximate our head-hand interaction model for multiple identities while maintaining computational efficiency.

Since the quality of the simulations can only be judged subjectively, we conducted a comprehensive user study which confirms the improved realism of our approach. In addition, we provide extensive visual results and inspect the neural approximations quantitatively. All data used in this work has been recorded with a multi-view camera rig. Code and data are available at [https://gitlab.cs.hs-rm.de/cvml\\_releases/HeadHand](https://gitlab.cs.hs-rm.de/cvml_releases/HeadHand).

## 1 Introduction

How many times per hour do you think you touch your face? Probably more often than you are aware of. Although the answer to this question varies in scientific studies, it can be said

that, on average, people touch their heads several dozen times an hour [KGM15, RMF20, MMG19]. There are many ways to interact, such as touching, stroking, scratching, rubbing, pulling, tugging, squeezing, and caressing, to name but a few. Behavioral sciences do not conclusively answer why people touch their faces,

yet the implications even extend to computer graphics. Due to the frequency and expressiveness of head-hand interactions, simulating them in facial animations would considerably improve user perception. Especially with the focus on photo-realistic avatars these days [QKS\*24, MWSZ24, ZBT23, AXS\*22, GKE\*22], the relevance of authentic facial animations is further accentuated.

Only recently, attempts to incorporate head-hand interactions into facial animations have been proposed [SGPT23, WDX\*24]. In particular, these approaches address two main challenges. Naturally, the simulation of interactions is the main emphasis, but three-dimensional tracking of the head and hands is also a prerequisite for realistic animations. Shimada et al. [SGPT23] and Wu et al. [WDX\*24] are impressive in determining simulated 3D head and hand surfaces from a single monocular image. However, both neglect the fidelity of the interaction animation as they are based on the same rather coarse physics-based *surface* simulation. More sophisticated, detailed, and anatomically accurate *volumetric* physics-based simulations of heads have been explored in other contexts [SNF05, IKKP17, Con16, CZ24].

This work introduces a substantially improved physics-based simulation of head-hand interactions and designs more realistic interaction mechanisms. For instance, in contrast to the previous methods, we consider pulling interactions and the influence of the skull. Since this simulation is not real-time capable, we also learn a personalized neural network as an approximation. Both our simulation and the network process tracked head and hand surfaces and thus remain compatible with the tracking concepts of previous approaches [SGPT23, WDX\*24]. Another contribution of this work is the creation of a dataset of real head-hand interactions. To this end, we built a multi-view rig with 16 high-resolution and synchronized video cameras with which we recorded several subjects. Unlike the only other comparable dataset available [SGPT23], we did not instruct the participants which head-hand interactions to perform. We simply asked the participants to perform arbitrary interactions and can, therefore, reproduce an even wider range of hand movements in our data. Among other things, we also capture skin pulling, which was previously ignored. Figure 1 is an exemplary illustration that shows a recorded *pulling* frame, the associated simulation, and the approximation by our neural network.

We evaluate our approach qualitatively using visual examples and the accompanying video of dynamic head-hand interaction animations. Furthermore, we conducted an extensive user study that confirms that our approach is perceived more naturally than previous ones. Quantitative experiments demonstrate that the neural approximation can be created in just a few hours and adapted to multiple human identities simultaneously. The trained network achieves around 50 frames-per-second (FPS) even on slower CPUs.

## 2 Related Work

In this section, we discuss three literature fields related to our approach. First, Section 2.1 presents physics-based facial animations in general. Next, Section 2.2 addresses recent developments focusing specifically on animated head-hand interactions. Finally, Section 2.3 examines work in which neural networks approximate physics-based simulations.

### 2.1 Physics-Based Facial Animations

Heuristic physics-based facial simulations have been developed for a long time and principally intend to compensate for shortcomings of *simpler* but popular facial animation methods like linear blendshapes [LAR\*14]. For instance, artifacts like implausible contortions and self-intersections can be avoided by including volumetric and anatomical constraints. The pioneering work of Sifakis et al. [SNF05] is a volumetric physics-based facial simulation that runs on a personalized tetrahedral mesh. Unfortunately, the tetrahedral mesh can only be of limited resolution due to an associated dense optimization problem. With Phace [IKKP17, IKNDP16], an improved simulation concept has been introduced, which is also defined on a tetrahedral mesh but can handle higher resolutions and considers anatomical structures more precisely. In addition to a tetrahedral mesh, the art-directed muscle models [CF19, BCGF19, Con16] represent muscles as B-splines that steer facial expressions via trajectories of spline control points. A solely inverse model for determining the physical properties of faces was proposed in [KK19].

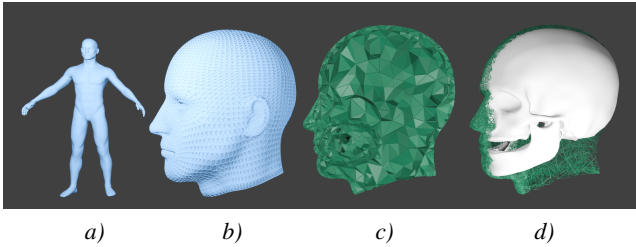
Thanks to increased computing capabilities, data-driven physics-based facial simulations have also become appealing recently. An example is the model of Yang et al. [YKZ\*22] that learns to volumetrically animate a person's face from multi-view videos with the help of differentiable physics [DWM\*21]. Although Yang et al. [YZC\*23] extend the model to cover several identities, adding a novel identity requires five days of retraining and the inference of one frame runs multiple seconds. While faster alternatives exist [WBS23], generally, heuristic as well as data-driven physics-based simulations are not commonly used in real applications due to their complexity and computational effort. Other data-driven approaches include Animatomy [CEM\*22], which represents muscles as curves, and the implicit model of Chandran et al. [CZ24]. The aforementioned data-driven simulations are not designed to handle collisions and external interactions.

### 2.2 Head-Hand Interactions

All previously discussed simulations have in common that they are primarily aimed at facial animation, facial retargeting, or face reconstruction, but not at the simulation of external influences such as hands. Although models like Phace [IKKP17] are theoretically applicable in such scenarios, the non-trivial practical implementation of interactions has not happened until lately. Shimada et al. [SGPT23] propose the first head-hand interaction simulation, Decaf, and demonstrate how a neural network can learn the simulation while generalizing over the FLAME head model [LBB\*17] and the MANO hand model [RTB17]. Decaf focuses on mapping a single RGB image to interaction deformations, using only a surface-based simulation that, in terms of quality and realism, falls short of the volumetric simulations discussed in Section 2.1. Also, the low resolution and the sometimes too smooth representation of heads in FLAME are often insufficient for demanding applications. Wu et al. [WDX\*24] advance Decaf by an extended generalization to in-the-wild images. Unfortunately, the underlying simulation remains the same. Consequently, we focus on a more realistic simulation for personalized and more detailed head avatars.

Variable	Description
$\mathbb{S}, \mathbb{J}, \mathbb{C}$	Tetrahedral meshes of soft tissue, jaw, and cranium
$H, L, R, J, C$	Surface meshes of head, left hand, right hand, jaw, and cranium
$E_*$	Energies
$w_*, s_*$	Scalar weights
$C_*$	Vertex targets of hand interactions
$I_*$	Set or dictionary of vertex indices
$*_t$	Indicates the time step $t$ of a variable
$*_{\text{src}}$	Indicates the source $\text{src}$ of a variable
$\mathcal{X}_T$	Sequence $(X_t)_{t=1}^T$ of $T$ surface meshes $X_t$
$\tilde{X}$	Projection into PCA space of surface mesh $X$
$v, c, t$	A geom. element like a vertex $v$ , a cylinder $c$ , or a tetrahedron $t$
$\text{func}$	Denotes a function

**Table 1:** The notation of the main concepts of Section 3.



**Figure 2:** a) Full-body template which includes the head template surface  $H$  shown in b). c) Cross section of the connected tetrahedral meshes  $\mathbb{S}, \mathbb{J}, \mathbb{C}$ . d) The template jaw and cranium surfaces  $J, C$  embedded in the tetrahedral meshes.

### 2.3 Approximating Physics-Based Simulations

As we accelerate our approach with efficient neural networks, we also give a brief overview of the literature on neural approximations of physics-based simulations. On the one hand, there are general methodologies [SWR\*21] that also explicitly deal with interactions of two or more objects [RCCO22, RCPO21]. On the other hand, there are methods with a focus on bodies [SGOC20, CO18] or heads [WBS23]. For NePHIM, we adopt the general method of subspace neural physics [HDDN19] that is, in particular, computationally efficient for approximating simulations of interacting objects.

## 3 Method

This section first outlines the objectives of our approach (Section 3.1) and then presents the formal implementation (Sections 3.2–3.5). To support the reading flow, we slightly misuse the notation in the following derivations by denoting a mesh and the corresponding vector of stacked vertex positions with the same symbol. Table 1 gives a summary of the notation.

### 3.1 Objectives & Method Overview

We consider an animation at time  $T$  with tracked surfaces for the left hand  $L_T^{\text{tra}}$ , the right hand  $R_T^{\text{tra}}$ , and the head  $H_T^{\text{tra}}$  of a person. Given the corresponding neutral head surface mesh  $H$  as well as tracked sequences (consisting of all previous frames up to  $T$ ) for the left hand  $\mathcal{L}_T = (L_t^{\text{tra}})_{t=1}^T$ , right hand  $\mathcal{R}_T = (R_t^{\text{tra}})_{t=1}^T$ , and head  $\mathcal{H}_T = (H_t^{\text{tra}})_{t=1}^T$ , our first objective is to deform the tracked head surface mesh at time  $T$ ,  $H_T^{\text{tra}}$ , to

$$H_T^{\text{phy}} = \text{phy}(\mathcal{R}_T, \mathcal{L}_T, \mathcal{H}_T, H), \quad (1)$$

such that head-hand interactions are resolved *realistically* through a physics-based simulation  $\text{phy}$ . Previous methods [SGPT23, WDX\*24] determine deformations through a simple surface-based simulation [MHHR07] incorporating only constraints for the skin surface and (static) pushing hand interactions. We improve realism by implementing  $\text{phy}$  (Section 3.3) as a *volumetric* simulation that additionally respects

- long-term collision *paths* of pushing interactions,
- pulling hand interactions,
- and volumetric anatomical constraints.

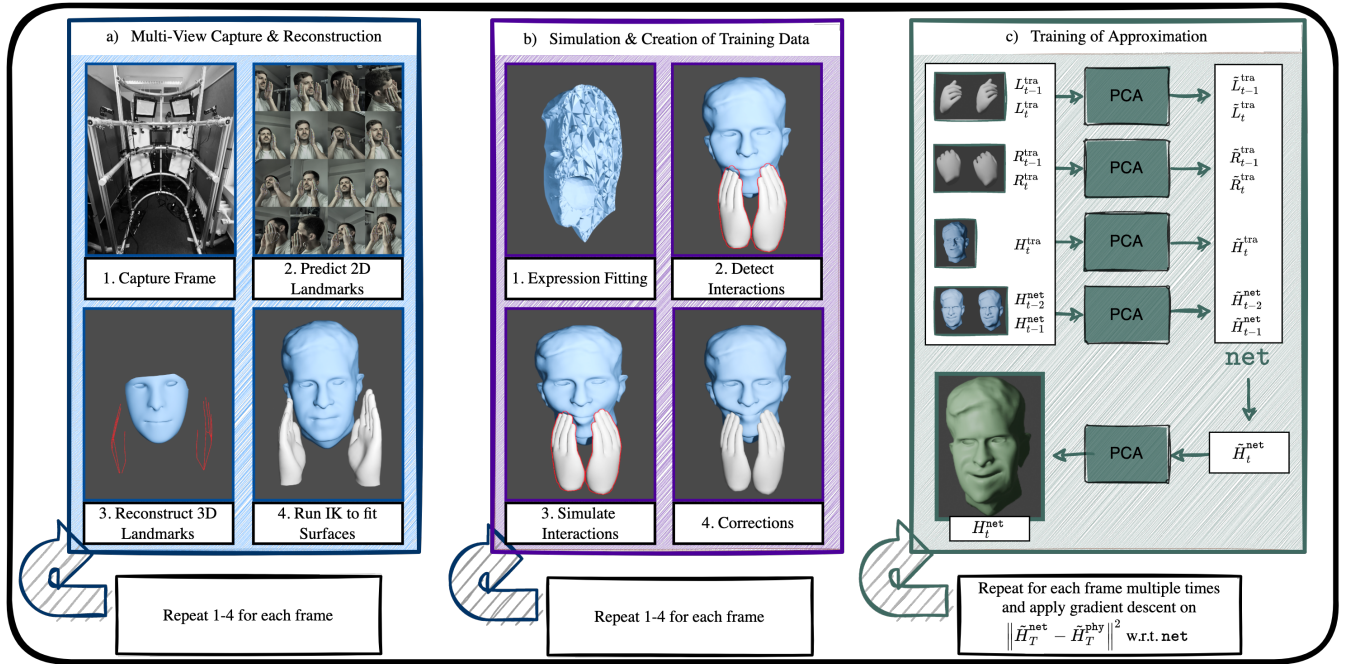
Although the resulting  $H_T^{\text{phy}}$  appears more natural (Section 4.4), our simulation  $\text{phy}$  is not real-time capable and, hence, potential applications are restricted. Therefore, our second objective is to train a neural network  $\text{net}$  (Section 3.5) that approximates  $H_T^{\text{phy}}$  while being real-time capable even on CPUs.

### 3.2 Volumetric Template

In the remainder of this section, we will precisely state  $\text{phy}$  and  $\text{net}$ . However, as our approach is intended to reflect volumetric constraints, we first introduce a head template  $(H, J, C, \mathbb{S}, \mathbb{J}, \mathbb{C})$  as the foundation of  $\text{phy}$ . The template includes the neutral head surface mesh  $H \subset \mathbb{S}$  that encloses a soft tissue tetrahedral mesh  $\mathbb{S}$ . The two template surface meshes  $J, C$  form the corresponding skull as jaw and cranium and are filled with respective tetrahedral meshes  $\mathbb{J}$  and  $\mathbb{C}$ . All tetrahedral meshes are connected, and the surface vertices of  $H$  can be addressed in  $\mathbb{S}$  with the same indices. An experienced digital artist designed the template surfaces while the tetrahedral meshes were created with TetGen [Han15].

Figure 2 b–d visualize all template components; all dimensions can be found in Appendix A. The tessellation of  $H$  is aligned with a full-body avatar (Figure 2a), which is part of the code release to easily integrate NePHIM into other applications.

To register the volumetric template to a tracked person, we expect the neutral head surface  $H$  of this person to be known. Then, we reposition the skull components by a dense linear model that we trained on the computed tomography dataset of Achenbach et al. [ABG\*18]. Formally, this model maps from the vertex positions of  $H$  to the vertex positions of the jaw  $J$  and the cranium  $C$ . The vertices of each tetrahedral mesh  $\mathbb{S}, \mathbb{J}, \mathbb{C}$  are placed by radial basis function space warps [BK05] calculated *from* the respective enclosing surfaces in the template *to* the corresponding surfaces of the tracked person.



**Figure 3:** Overview of the three stages of our approach. a) Data capturing as described in Section 4.1. b) All steps of our physics-based simulation  $\text{phy}$  as explained in Section 3.3. c) Efficient neural approximation  $\text{net}$  of  $\text{phy}$  as explained in Section 3.5.

### 3.3 Simulation

Building on the volumetric template, we can now continue with the detailed introduction of our physics-based simulation  $\text{phy}$ . As Algorithm 1 outlines,  $\text{phy}$  conducts four steps that are described in separate subsections from Section 3.3.1 to Section 3.3.4. Figure 3b visualizes an exemplary cycle of all steps. Since we want to take long-term effects such as collision paths and skin pulling into account, it is not sufficient to consider only the last time step  $T$  to determine  $H_T^{\text{phy}}$ . Instead, we start at the beginning of the tracked sequences and run all four simulation steps consecutively for each time step  $t$ .

#### 3.3.1 Expression Fitting

As the initial simulation step, we deform the neutral volumetric tetrahedral meshes  $\mathbb{S}$ ,  $\mathbb{J}$ , and  $\mathbb{C}$  in an anatomically plausible manner to fit the tracked surface  $H_t^{\text{tra}}$  instead of the neutral surface  $H$ . To this end, we minimize a constraint-based energy in the projective dynamics (PD) simulation framework [BML\*14]. The first objective

$$E_{\text{target}}(H, H_t^{\text{tra}}) = \|H - H_t^{\text{tra}}\|^2 \quad (2)$$

attracts the surface vertices  $H \subset \mathbb{S}$  of the soft tissue tetrahedral mesh towards the tracked head surface. The second objective

$$E_{\text{strain}}(\mathbb{S}) = \sum_{\mathbf{t} \in \mathbb{S}} s_{\mathbf{t}} \min_{\mathbf{R} \in SO(3)} \|\nabla(\mathbf{t}) - \mathbf{R}\|_F^2 \quad (3)$$

models strain for each soft tissue tetrahedron  $\mathbf{t} \in \mathbb{S}$ . Here,  $\mathbf{R} \in SO(3)$  denotes the optimal rotation,  $\nabla(\mathbf{t}) \in \mathbb{R}^{3 \times 3}$  the deformation gradient of  $\mathbf{t}$  (w.r.t. the neutral rest shape),  $\|\cdot\|_F$  the Frobenius norm, and  $s_{\mathbf{t}} \in (0, 1]$  is a stiffness value calculated as in [SGPT23]. In intuitive

and simplified terms, the stiffness decreases the further a tetrahedron is located from the skull. Analogous to the soft tissue strain, we also add strain energies for the jaw  $E_{\text{strain}}(\mathbb{J})$  and the cranium  $E_{\text{strain}}(\mathbb{C})$ . Overall, the weighted energy

$$E_{\text{tracked}}(H_t^{\text{tra}}, \mathbb{S}, \mathbb{J}, \mathbb{C}) = w_{\text{tar}} E_{\text{target}}(H, H_t^{\text{tra}}) + w_{\mathbb{S}} E_{\text{strain}}(\mathbb{S}) + w_{\mathbb{J}} E_{\text{strain}}(\mathbb{J}) + w_{\mathbb{C}} E_{\text{strain}}(\mathbb{C}) \quad (4)$$

is minimized. To reflect that both jaw and cranium are rigid, we set the weights  $w_{\mathbb{J}}$  and  $w_{\mathbb{C}}$  to a high value compared to  $w_{\text{tar}}$  and  $w_{\mathbb{S}}$  and apply a constant stiffness of one. The values of all weights and other simulation parameters can be found in Appendix B. The outputs of the optimization are the tracked tetrahedral meshes

$$(\mathbb{S}^{\text{tra}}, \mathbb{J}^{\text{tra}}, \mathbb{C}^{\text{tra}}) = \underset{\mathbb{S}, \mathbb{J}, \mathbb{C}}{\text{argmin}} E_{\text{tracked}}(H_t^{\text{tra}}, \mathbb{S}, \mathbb{J}, \mathbb{C}). \quad (5)$$

Please note that although there are more detailed simulation methods than PD [LFS\*20, IKKP17, YKZ\*22], these are often more complex and cannot outweigh the efficiency of PD in our use case.

#### 3.3.2 Detect Interactions

Subsequently, we detect pushing and pulling head-hand interactions and translate them into target positions of the head vertices  $C_{\text{push}}$  and  $C_{\text{pull}}$ , which we will simulate in the next step (Section 3.3.3).

**Pushing Interactions** We first explain how we handle pushing in  $\text{phy}$ . Previous approaches [SGPT23, WDX\*24] would simply iterate over the vertices of the head surface  $H_t^{\text{tra}}$  and if a vertex enters either the left hand  $L_t^{\text{tra}}$  or the right hand  $R_t^{\text{tra}}$ , it is moved in the

**Algorithm 1** Volumetric Physics-Based Simulation `phy`


---

```

Function phy( $\mathcal{R}_T, \mathcal{L}_T, \mathcal{H}_T, H$ )
  // Section 3.2 Register Template to Neutral
  Register volumetric template (Figure 2) to obtain the tracked person's volumetric head description ( $H, J, C, \mathbb{S}, \mathbb{J}, \mathbb{C}$ ).

   $t = 1$ 
  while  $t \leq T$  do
    // Section 3.3.1 Expression Fitting
    Step 1 Determine the tracked tetrahedral meshes  $\mathbb{S}^{\text{tra}}, \mathbb{J}^{\text{tra}}, \mathbb{C}^{\text{tra}}$  by aligning  $\mathbb{S}, \mathbb{J}, \mathbb{C}$  with the tracked head surface  $H_t^{\text{tra}}$  as described in Equation (5).

    // Section 3.3.2 Detect Interactions
    Step 2 Determine the push and pull target positions  $C_{\text{push}}, C_{\text{pull}}$  as described in Algorithm 2 and Algorithm 3, respectively.

    // Section 3.3.3 Simulate Interactions
    Step 3 Determine the interaction tetrahedral meshes  $\mathbb{S}^{\text{int}}, \mathbb{J}^{\text{int}}, \mathbb{C}^{\text{int}}$  by applying the push and pull targets  $C_{\text{push}}, C_{\text{pull}}$  to  $H_t^{\text{tra}}$  as described in Equation (9).

    // Section 3.3.4 Corrections
    Step 4 Determine the corrected tetrahedral meshes  $\mathbb{S}^{\text{cor}}, \mathbb{J}^{\text{cor}}, \mathbb{C}^{\text{cor}}$  by resolving remaining collisions  $I_{\text{corr}}$  as described in Equation (11). Extract  $H_t^{\text{phy}}$  from  $\mathbb{S}^{\text{cor}}$ .

     $t = t + 1$ 

  // Return the simulated head surface
  return  $H_T^{\text{phy}}$ 

```

---

direction of the corresponding inverted normal until the collision is resolved. Unfortunately, this strategy largely ignores temporal dependencies, and the normal direction only provides an imprecise collision resolution.

For this reason, we rely on the linear movements between  $H_{t-1}^{\text{phy}}$  and  $H_t^{\text{tra}}, L_{t-1}^{\text{tra}}$  and  $L_t^{\text{tra}}$ , as well as  $R_{t-1}^{\text{tra}}$  and  $R_t^{\text{tra}}$ , i.e., between the previous simulated frame and the current tracked frame, as formally described in Algorithm 2. Expressed in words, we check at short intervals  $\epsilon$  between the time steps  $t-1$  and  $t$  whether one of the two hands touches a vertex of the head. If so, the head vertex is dragged from the initial point of contact with the hand at  $\epsilon$  to the same point on the hand at time  $t$ . Please see Figure 4a for a visual explanation. Our way of resolving hand pushing is more *natural* and incorporates long-term effects per construction. Although there are more involved and time-consuming forms of continuous collision detection, these did not yield substantially better results in our experiments.

**Pulling Interactions** Pulling is considerably more challenging and has not been addressed in prior approaches. We present a heuristic in Algorithm 3 that does not require cumbersome friction calculations but, unfortunately, still has an elaborated notation. Yet, the foundational idea of our heuristics can easily be put into words. First, we form cylinders with radius  $r$  between the fingertips of all fingers (index, middle, ring, little) and the thumb as illustrated in Figure 4b. Then, for each cylinder, we determine whether it grabs, i.e., has shortened in length from time step  $t-1$  to time step  $t$ . If so, and if the length falls below a minimum  $l_{\text{min}}$ , all head vertices inside the cylinder at time  $t$  are marked as *pulled*. We maintain a dictionary  $I_{\text{pull}}$  over time that stores a set of the *pulled* vertices for each finger.

**Algorithm 2** Pushing Interaction

---

```

Function push( $H_{t-1}^{\text{phy}}, H_t^{\text{tra}}, L_{t-1}^{\text{tra}}, L_t^{\text{tra}}, R_{t-1}^{\text{tra}}, R_t^{\text{tra}}$ )
  // Initialize linear movement directions
   $H_{\text{dir}} = H_t^{\text{tra}} - H_{t-1}^{\text{phy}}$ 
   $L_{\text{dir}} = L_t^{\text{tra}} - L_{t-1}^{\text{tra}}$ 
   $R_{\text{dir}} = R_t^{\text{tra}} - R_{t-1}^{\text{tra}}$ 

  // Initialize push targets
   $C_{\text{push}} = \{\}$ 
   $I = \{\}$ 

  // Iterate over linear movements
  for  $\epsilon = 0; \epsilon \leq 1; \epsilon += \Delta\epsilon$  do
    // Iterate over head surface vertices
    for  $v_i^H \in (H_{t-1}^{\text{phy}} + \epsilon \cdot H_{\text{dir}})$  do
      // Find collisions with left hand
      if  $v_i^H$  collides with  $(L_{t-1}^{\text{tra}} + \epsilon \cdot L_{\text{dir}})$  and  $i \notin I$  then
        // Find nearest neighbor in current left hand
         $v_{j,\epsilon}^L = \text{nn}(v_i^H, L_{t-1}^{\text{tra}} + \epsilon \cdot L_{\text{dir}})$ 
        // Add same vertex of final left hand as target position
        Add  $(v_{j,\epsilon}^L, i)$  to  $C_{\text{push}}$ 
        Add  $i$  to  $I$ 
      Repeat the same if-clause for the right hand

  // Return the push targets
  return  $C_{\text{push}}$ 

```

---

The target positions of the *pulled* vertices  $C_{\text{pull}}$  are calculated so that they form smooth ridges within the cylinders (see Figure 4b). The shape of the ridges imitates the skin's natural deformation due to pinching. A *pulled* vertex is unmarked once the corresponding cylinder exceeds  $l_{\text{min}}$ , i.e., the finger no longer grabs.

**3.3.3 Simulate Interactions**

For applying the previously determined push and pull targets  $C_{\text{push}}$  and  $C_{\text{pull}}$  to the tracked head  $H_t^{\text{tra}}$ , we again make use of a PD simulation on the fitted tetrahedral meshes  $\mathbb{S}^{\text{tra}}, \mathbb{J}^{\text{tra}}$ , and  $\mathbb{C}^{\text{tra}}$  (Section 3.3.1). Here, we establish anatomical plausibility similar as before by adding strain constraints  $E_{\text{strain}}(\mathbb{S}^{\text{tra}})$ ,  $E_{\text{strain}}(\mathbb{J}^{\text{tra}})$ , and  $E_{\text{strain}}(\mathbb{C}^{\text{tra}})$  to the simulation. Also as before, we add

$$E_{\text{target}}(H^{\text{tra}}, H_t^{\text{tra}}) = \left\| H^{\text{tra}} - \left( H_t^{\text{tra}} + \frac{\alpha}{s} (H_{t-1}^{\text{phy}} - H_{t-2}^{\text{phy}}) \right) \right\|^2 \quad (6)$$

to draw the surface vertices  $H^{\text{tra}} \subset \mathbb{S}^{\text{tra}}$  of the soft tissue to the tracked surface. This time, however, including damped velocities of the head, where  $s$  denotes the size of a time step. A low damping factor  $\alpha$  adds natural-looking dynamic effects to the interactions.

New to the simulation are the target constraints

$$E_{\text{push}}(H^{\text{tra}}, C_{\text{push}}) = \sum_{(p,i) \in C_{\text{push}}} \|p - v_i\|^2, \quad (7)$$

$$E_{\text{pull}}(H^{\text{tra}}, C_{\text{pull}}) = \sum_{(p,i) \in C_{\text{pull}}} \|p - v_i\|^2,$$

which draw interacting vertices  $v_i \in H^{\text{tra}}$  to their precalculated target

**Algorithm 3** Pulling Interaction

---

*Notation*  
 $c_t^{L,f}$  Cylinder of finger  $f$  of the left hand  $L$  at timestep  $t$   
 $len$  Length of a cylinder  
 $I_{pull}[L, f]$  Dictionary entry of key  $L, f$ , i.e., a set

**Function**  $pull(H_t^{tra}, L_{t-1}^{tra}, L_t^{tra}, R_{t-1}^{tra}, R_t^{tra}, I_{pull})$

// Initialize pull targets  
 $C_{pull} = \{\}$

// Check if new vertices are pulled per cylinder  
**for**  $f = 1; f \leq 4; f += 1$  **do**  
  // Pull only if cylinder gets smaller and is small enough  
  **if**  $len(c_t^{L,f}) < len(c_{t-1}^{L,f})$  and  $len(c_t^{L,f}) < l_{min}$  **then**  
    // Check for each head vertex if inside cylinder  
    **for**  $v_i \in H_t^{tra}$  **do**  
      **if**  $v_i$  inside  $c_t^{L,f}$  **then**  
        Append  $i$  to  $I_{pull}[L, f]$

// Check if vertices are no longer pulled per cylinder  
**for**  $f = 1; f \leq 4; f += 1$  **do**  
  **if**  $len(c_t^{L,f}) \geq l_{min}$  **then**  
     $I_{pull}[L, f] = \emptyset$

// Calculate target positions of pulled vertices per cylinder by  
// creating a ridge per cylinder as defined in Appendix C  
**for**  $f = 1; f \leq 4; f += 1$  **do**  
  Append  $ridge(I_{pull}[L, f], H_t^{tra}, c_t^{L,f})$  to  $C_{pull}$

**Repeat** same procedure for the right hand

// Return the pull targets  
**return**  $C_{pull}$

---

position  $p$ . Overall, the weighted energy

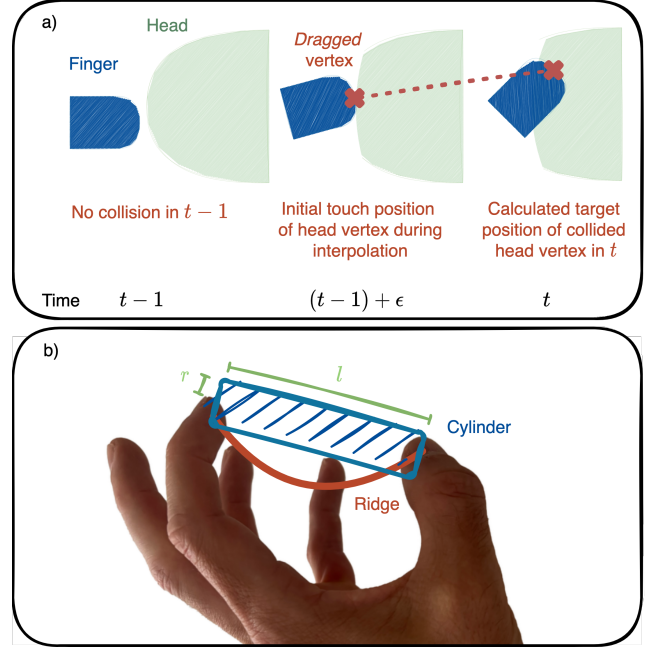
$$\begin{aligned}
 E_{inter}(C_{push}, C_{pull}, H_t^{tra}, S^{tra}, J^{tra}, C^{tra}) = & \\
 & w_{push} E_{push}(H_t^{tra}, C_{push}) + \\
 & w_{pull} E_{pull}(H_t^{tra}, C_{pull}) + \\
 & w_{tar} E_{target}(H_t^{tra}, H_t^{tra}) + \\
 & w_S E_{strain}(S^{tra}) + \\
 & w_J E_{strain}(J^{tra}) + \\
 & w_C E_{strain}(C^{tra})
 \end{aligned} \quad (8)$$

is minimized, where we again set the weights  $w_J$  and  $w_C$  to a high value for approximating a rigid skull. Likewise, the weights  $w_{push}$  and  $w_{pull}$  are set to a high value to enforce the target positions, but lower as  $w_J, w_C$ . By balancing the previously mentioned weights, we achieve a more natural simulation since the bones do not bend in the case of tracking errors and too deeply penetrating hands. The outputs of the optimization are the interaction tetrahedral meshes

$$\begin{aligned}
 (S^{int}, J^{int}, C^{int}) = \operatorname{argmin}_{S^{tra}, J^{tra}, C^{tra}} E_{inter}(C_{push}, C_{pull}, \\
 H_t^{tra}, S^{tra}, J^{tra}, C^{tra}).
 \end{aligned} \quad (9)$$

### 3.3.4 Corrections

The preceding steps of  $phy$  do not fully resolve all head-hand collisions. For instance, the last step in Section 3.3.3 allows soft



**Figure 4:** a) Visualization of pushing as described in Section 3.3.2 and Algorithm 2. Here,  $\epsilon$  is a substep between the time steps  $t - 1$  and  $t$ . b) Illustration of a finger cylinder with radius  $r$ , length  $l$ , and an exemplary ridge shape that is used for pulling as described in Algorithm 3.

tissue vertices that previously did not collide to move inside the hands. To correct most remaining colliding vertices, summarized with their indices in  $I_{corr}$ , we perform the previous PD simulation again but add an additional constraint. This constraint

$$E_{corr}(S^{int}, I_{corr}) = \sum_{i \in I_{corr}} \|nn(v_i, L_t^{tra}, R_t^{tra}) - v_i\| \quad (10)$$

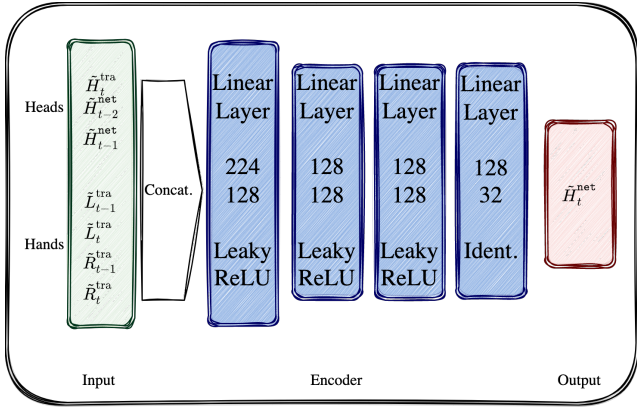
draws each colliding vertex  $v_i \in S^{int}$  to the nearest neighbor  $nn(v_i, L_t^{tra}, R_t^{tra})$  of  $v_i$  on the left or right hand  $L_t^{tra}, R_t^{tra}$ . The outputs of the optimization are the corrected tetrahedral meshes

$$\begin{aligned}
 (S^{cor}, J^{cor}, C^{cor}) = \operatorname{argmin}_{S^{int}, J^{int}, C^{int}} w_{corr} E_{corr}(S^{int}, I_{corr}) + \\
 E_{inter}(C_{push}, C_{pull}, H_t^{tra}, S^{int}, J^{int}, C^{int}).
 \end{aligned} \quad (11)$$

The deformed surface  $H_t^{phy} = phy(\mathcal{R}_t, \mathcal{L}_t, \mathcal{H}_t, H) \subset S^{cor}$  can now be extracted as the outer boundary of the soft tissue mesh. After the four steps of  $phy$  described in Sections 3.3.1–3.3.4 have been carried out consecutively for all time steps through to  $T$ ,  $H_T^{phy}$  is obtained.

### 3.4 Recursive Formulation

The previous description of  $phy$  serves the intuitive derivation, but suggests that the computational effort increases linearly with each additional frame. However, this is not the case, since by the design



**Figure 5:** An overview of the efficient network architecture of  $net$ . Basically, a simple MLP with only 65536 parameters.

of  $phy$ , we can rewrite Equation (1) recursively as

$$H_T^{phy} = phy(L_T^{tra}, R_T^{tra}, H_T^{tra}, L_{T-1}^{tra}, R_{T-1}^{tra}, H_{T-1}^{phy}, H_{T-2}^{phy}). \quad (12)$$

Hence, we can reuse simulated frames instead of always simulating all time steps.

### 3.5 Neural Simulation Approximation

As the derivations in the previous sections already indicate,  $phy$  is not real-time capable. Therefore, we construct  $net$ , a neural network that can be evaluated even on CPUs with 50 FPS (Table 4) and that closely approximates  $phy$ . From the wide corpus of techniques that already exist for approximating physic-based simulations (Section 2.3), we adapt subspace neural physics (SNP) [HDDN19] to our needs. Here, we only explain our adapted architecture of  $net$ , as the original publication extensively describes the training algorithm and we do not modify it.

The principle idea of SNP is to project all inputs and outputs into smaller linear subspaces (e.g. using principal component analysis (PCA)) and to train  $net$  on the projection. In the following, the pedant of a variable in its respective subspace is referenced with an overlying tilde. The inputs of  $net$  with regard to  $phy$  as defined in Equation (12) are

$$L_T^{tra}, R_T^{tra}, H_T^{tra}, L_{T-1}^{tra}, R_{T-1}^{tra}, H_{T-1}^{net}, H_{T-2}^{net}. \quad (13)$$

Consequently, we have to create PCA subspaces for the tracked left hand, the tracked right hand, the tracked head, and the simulated head, respectively. The overall training goal is then to minimize

$$\min_{net} \left\| \tilde{H}_T^{net} - \tilde{H}_T^{phy} \right\|^2, \quad (14)$$

where

$$\tilde{H}_T^{net} = net(\tilde{L}_T^{tra}, \tilde{R}_T^{tra}, \tilde{H}_T^{tra}, \tilde{L}_{T-1}^{tra}, \tilde{R}_{T-1}^{tra}, \tilde{H}_{T-1}^{net}, \tilde{H}_{T-2}^{net}). \quad (15)$$

A visual illustration of the inputs and outputs of  $net$  is depicted in Figure 3c and our architecture can be found in Figure 5. To recover  $H_T^{net}$  from  $\tilde{H}_T^{net}$ , the PCA of the simulated heads is applied. By selecting an appropriate number of components of the subspace, we prevent the loss of geometric details.

## 4 Results

The result section is organized as follows. First, we outline how we capture and process real head-hand interactions to form training and test data (Section 4.1). The same subsection also contains a description of the resulting dataset and details on training and evaluation protocols. In Section 4.2, we discuss qualitative characteristics of the simulation  $phy$  and the approximation  $net$  using visual examples. In Section 4.3, we examine quantitative characteristics and also take a closer look at running times as well as training times. Finally, we present the results of a user study (Section 4.4) that supports the more natural perception of our approach.

### 4.1 Dataset & Training

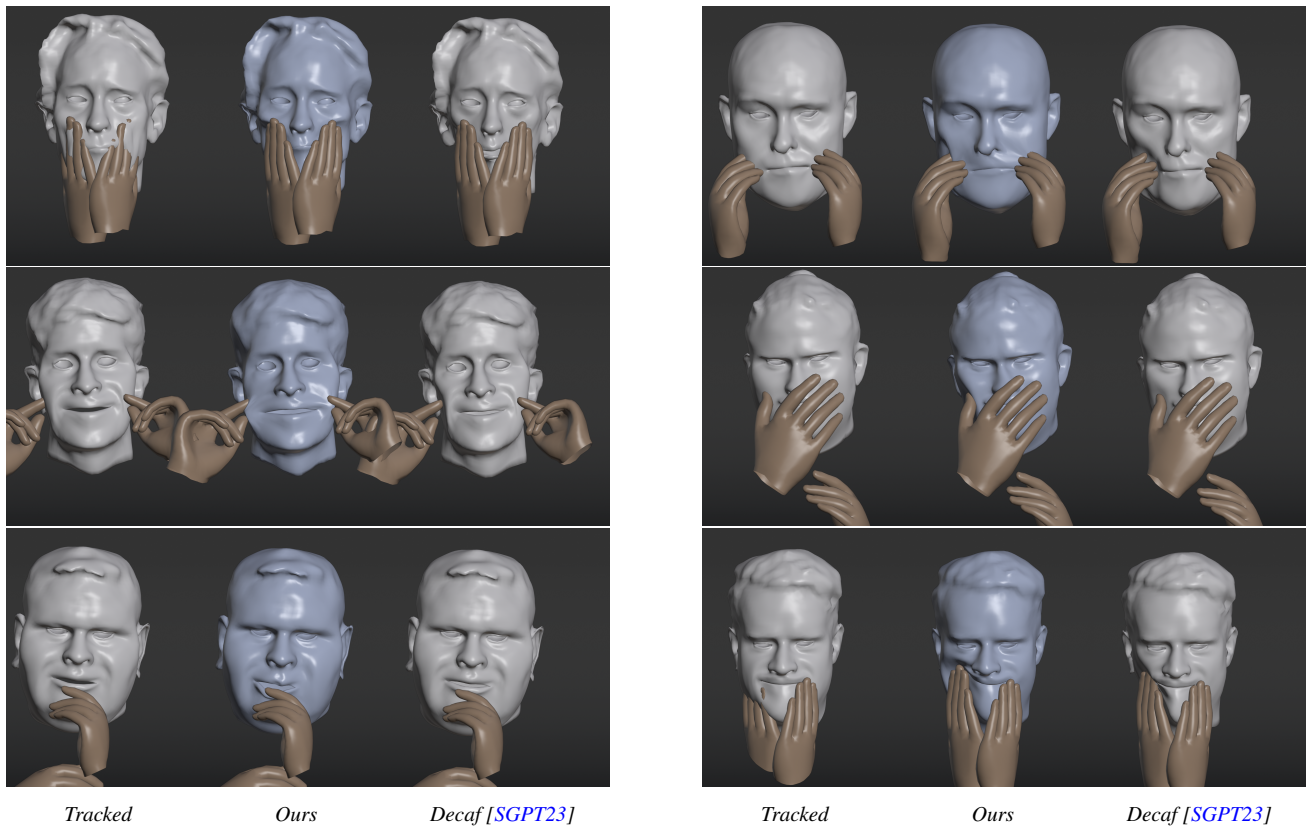
To capture real head-hand interactions, we use a multi-view rig consisting of 16 synchronized and calibrated XIMEA [Xim24] RGB cameras generating 12-megapixel images at 20 FPS. In each captured image, we predict 2D landmarks for both hands and head using existing tracking methods [BT17, ZBV\*20]. For the hands, a landmark is predicted for each joint, each fingertip, and the wrists. For heads, we only capture the contours of the eyes and the mouth, as can be seen in Figure 1. From the 2D landmarks, we generate 3D landmarks per frame using a basic bundle adjustment algorithm.

Since our simulation  $phy$  is conceptualized to work on tracked surfaces, the last step in the capturing pipeline is to fit appropriate template surfaces to the 3D landmarks. Regarding the head, we initially create a high-resolution personalized head avatar for the recorded person with an automated 3D reconstruction and (non-linear) template fitting pipeline [WAB\*20]. Afterward, we add linear blendshapes to the avatar by an *automated* volumetric deformation transfer [WSB24, SP04] of a set of template blendshapes. The template blendshapes represent the 52 ARKit expressions [App24] and were *manually* sculpted *once* by a professional digital artist.<sup>†</sup> Finally, we optimize per frame a set of corresponding blendshape weights, a translation vector, and a rotation matrix to fit the head surface to the respective 3D landmarks. Regarding the hands, we adopt a similar approach. Here, however, we do not use a personalized hand model but optimize the pose and shape parameters of the MANO [RTB17] hand model to match the respective 3D landmarks. Contrary to the pose parameters, the shape parameters are the same for each frame. We use gradient descent as the optimizer for the surface fittings. Figure 3a illustrates all steps of the capturing pipeline.

The dataset we compiled contains up to 10 recordings of each of 8 individuals. The individuals are Caucasian males aged 26 to 54 with a body mass index ranging from slightly underweight to obese. Each recording lasts approximately 30 seconds and captures arbitrary head-hand interactions. In particular, we did not instruct

<sup>†</sup> The template blendshapes are part of the code release.





**Figure 6:** The figure shows examples of our simulation `phy` and compares them to the tracked surfaces as well as the simulation of Decaf [SGPT23]. In the top left, for example, the advantage of simulating the skull becomes apparent near the cheekbone. In the top right image, a pulling interaction is shown and the lower images demonstrate the importance of (time-dependent) collision paths.

the individuals on which hand movements or facial expressions they should perform. Appendix D summarizes the frequency and the types of interactions. Overall, we captured, reconstructed, and simulated around 50000 frames for this work. All of the following experiments concerning the neural network `net` are always stated as an average of five runs, and we uniformly (i.e., non-consecutively) draw random train/test splits (90%/10%) for each run. All PCA subspaces have 32 components, which is sufficient in our case as we do not intend to generalize over large head or hand models. We rebuilt the subspaces for each run on the respective training data, and if several individuals are part of an experiment, we form joint subspaces. Neural networks and the inference of PCAs are implemented with PyTorch [PGM\*19] while PCA subspaces are constructed with the default implementation of Scikit [PVG\*11].

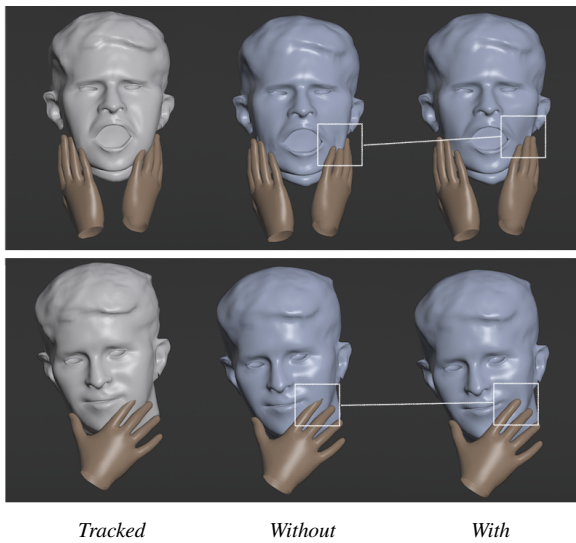
## 4.2 Qualitative Evaluation

Figure 6 (and additional examples in Appendix E) display instances of the simulation `phy` in comparison to the tracked surfaces as well as the simulation of the current state-of-the-art Decaf [SGPT23]. Please note that we implemented the latter simulation ourselves as the announced implementations are not (yet) available. Decaf results sometimes appear slightly different to those from [SGPT23], which mainly stems from the fact that our head avatars are more

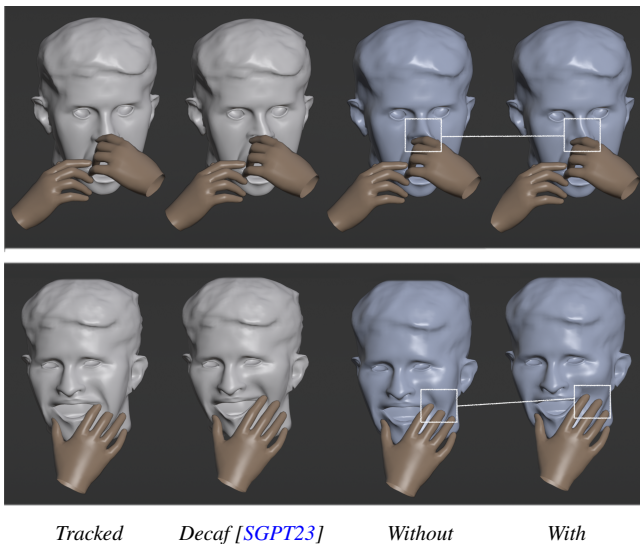
detailed than FLAME [LBB\*17] and that we did not instruct the recorded persons which head-hand interactions they should perform. In the shown examples, it is especially striking that our temporal processing of hand pushes leads to effects such as a bent nose, a pushed-up mouth corner, or even a pushed-down lip. Moreover, the pulling of skin is readily recognizable and appears natural. None of these effects can be observed with the other methods. The accompanying video demonstrates the advantages of our method for dynamic scenes.

Besides the more general examples, we show further visual comparisons to inspect individual stages of our approach. Figure 7 emphasizes the necessity of the correction step of `phy` (Section 3.3.4) while Figure 8 stresses the relevance of modeling temporal effects in our simulation. However, Figure 8 not only exhibits the impact of temporal effects on our simulation but also contrasts our simulation without temporal effects to the Decaf [SGPT23] simulation. Finally, Figure 9 underpins the advantage of a volumetric anatomy simulation by contrasting bendable and rigid bones.

Examples of our simulation `phy` along with the learned approximation `net` are depicted in Figure 10. For this purpose, we trained `net` on all identities in our dataset simultaneously. Although minor discrepancies can be recognized, these do not appear to be decisive for visual perception. Moreover, the quality of the approximation is



**Figure 7:** Examples of our simulation  $\text{phy}$  without and with the correction step (Section 3.3.4). Errors due to the missing correction step can accumulate over time.



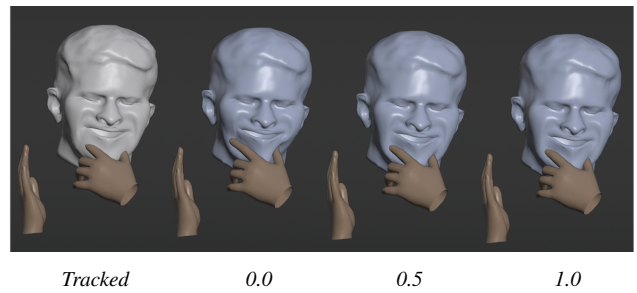
**Figure 8:** Examples of our simulation  $\text{phy}$  without and with temporal effects as well as the Decaf [SGPT23] simulation.

not affected by whether the head-hand interactions are pushing or pulling. Again, the accompanying video contains further examples.

### 4.3 Quantitative Evaluation

#### 4.3.1 Accuracy

This section mainly investigates the quantitative properties of the network  $\text{net}$  and the simulation  $\text{phy}$ . To begin with, we have a look at the approximation accuracy of  $\text{net}$ . For this purpose, Table 2 summarizes average train and test subspace errors (mean squared error) of  $\text{net}$  as well as the average and maximum reconstruction



**Figure 9:** Example of our simulation  $\text{phy}$  applying either 0%, 50%, or 100% of the bone weights  $w_J, w_C$ .

Dataset	# Identities	Subspace	Reconstruction	
		Mean MSE	Mean $\ell^2$	Max $\ell^2$
Train	One	0.011	0.02 cm	0.11 cm
	Eight	0.041	0.04 cm	0.18 cm
Test	One	0.052	0.09 cm	0.23 cm
	Eight	0.056	0.10 cm	0.35 cm

**Table 2:** Train and test errors of the neural approximation  $\text{net}$  of the simulation  $\text{phy}$ . The table is separated by the number of identities  $\text{net}$  was trained on. The errors stated for one identity are the average over separate networks for all identities in our dataset.

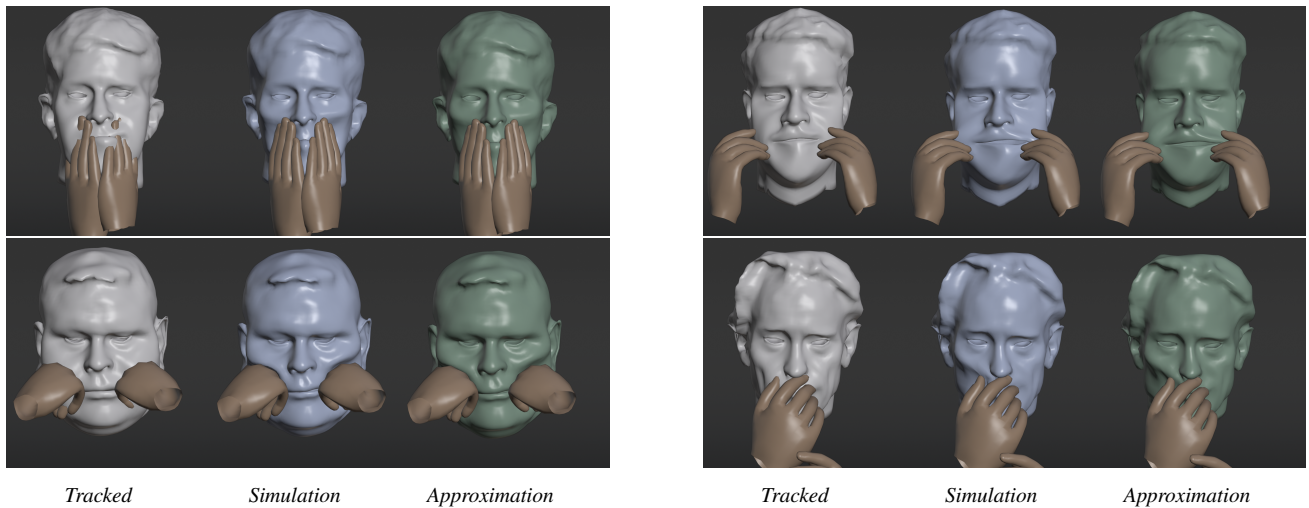
errors on the actual surfaces ( $\ell^2$  error). There is also a breakdown by the number of identities with which we trained and tested  $\text{net}$ . The table indicates that the reconstruction test errors are never greater than a millimeter on average, and our implementation of  $\text{net}$  has sufficient capacity to generalize over several identities. Moreover, the likewise small maximum reconstruction errors indicate that all kinds of simulated deformations can be adequately approximated by  $\text{net}$  without hallucinating non-existent interactions.

#### 4.3.2 Plausibility

In Table 3, we compare the plausibility of our network  $\text{net}$ , our simulation  $\text{phy}$ , and the simulation of Decaf [SGPT23] by means of quantitative metrics introduced in Shimada et al. [SGPT23]. Among them is the *Non Collisions* metric, which captures the number of collision-free frames after applying a method. We also state the *Collision Distance*, which measures the average per-vertex depth of the remaining collisions. We complement the existing metrics with the *Deformation Distance*, which calculates the average per-vertex deformation of the tracked head caused by a method. Table 3 indicates that all methods significantly reduce the number of colliding frames, and the remaining collisions are less deep. Although Decaf appears to better resolve collisions at first glance, this is to be expected, as it is able to bend bones unnaturally, for instance. This expectation is also supported by the *Deformation Distance*, which demonstrates that Decaf tends to apply larger deformations in general.

#### 4.3.3 Timings

Table 4 summarizes the average running times of  $\text{phy}$  and  $\text{net}$ . On the one hand, with a runtime of 876 ms per frame on an AMD



**Figure 10:** Examples of our simulation  $phy$  along with the tracked surfaces as well as the learned neural approximation  $net$  (trained on all identities in our dataset). The quality of the approximation is independent of whether it is a pushing or a pulling interaction.

Method	Non Collisions	Collision Dist.	Deformation Dist.
Tracked	53 %	1.20 cm	0.00 cm
$phy$	69 %	0.19 cm	0.11 cm
$net$	68 %	0.20 cm	0.09 cm
Decaf	8 %	0.09 cm	0.16 cm

**Table 3:** Plausibility metrics to compare our simulation  $phy$ , the Decaf simulation [SGPT23], and our network  $net$  to the tracked input. Non Collisions is the percentage of frames without collisions, Collision Distance measures the average per-vertex penetration depth of the remaining collisions, and Deformation Distance indicates the average per-vertex deformation by the respective method.

Input Size	$phy$ CPU	$net$ CPU	$net$ GPU
1 ×	876 ms	19.2 ms	5.1 ms
2 ×	2248 ms	34.6 ms	7.3 ms
4 ×	6553 ms	79.2 ms	9.4 ms

**Table 4:** The average inference times of the simulation  $phy$  and the neural approximation  $net$  depending on the input size, i.e., the number of surface vertices, the number of volumetric vertices, and the size of the PCA subspaces.

Ryzen Threadripper PRO 3995WX,  $phy$  is evidently not real-time-capable. On the other hand,  $net$  can be executed not only on a consumer-grade GPU (NVIDIA RTX 3090) but also on a weaker CPU (Intel i5 12600K) with more than 50 FPS. In comparison, our implementation of the simulation of Decaf [SGPT23] runs in 178 ms per frame on the Threadripper CPU. The entire pipeline, as shown in Figure 3, from data acquisition to training  $net$  only takes about 20 hours for eight identities and 3.5 hours for one identity. We trained on a NVIDIA A6000 GPU for four hours (eight identities) or one

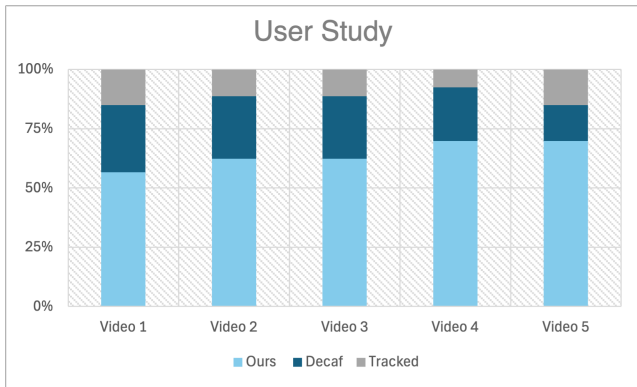
Tracking + $net$	Decaf	Dice
15.3 ms / 66 FPS	88 ms / 11.5 FPS	19590 ms / 0.05 FPS

**Table 5:** GPU inference times of our neural approximation  $net$  compared to Decaf [SGPT23] and Dice [WDX\*24]. For a fair comparison, since Decaf and Dice include tracking components, we added Mediapipe’s [BT17, ZBV\*20] head and hand tracking ahead of our network. The times were measured on a 128-core AMD Ryzen CPU and a NVIDIA A6000 GPU.

and a half hours (one identity). The short training time is mainly due to the efficient network architecture.

The aforementioned running times depend on the resolution of the underlying template. Although our template is already able to capture detailed deformations, Table 4 also shows that we can still efficiently execute  $net$  if the template is further refined. To that end, we doubled and quadrupled the number of surface and volumetric template vertices (remeshing) as well as the size of the PCA subspaces. On the CPU,  $net$  still runs at interactive rates if the resolution is doubled, whereas on the GPU, even a quadrupling is feasible. Nevertheless, as can also be seen in Table 4, the running time of the simulation increases substantially, and so does the time needed for generating training data.

We have intentionally designed  $net$  to be independent of any particular tracking method, and the running times stated in Table 4 imply that it can readily be integrated with other applications. However, in order to compare the inference times with those of Decaf [SGPT23] and Dice [WDX\*24], we trained a slightly modified  $net$ . For this modification, we input 2D head and hand landmarks tracked by Mediapipe [BT17, ZBV\*20] on the most frontal camera of our multi-view rig instead of PCA subspace representations of the undeformed surfaces. The test errors of this network are close



**Figure 11:** A user study among 53 participants supports that our approach is recognized as more natural. For each video shown in the user study, our approach received the most votes.

to those stated in Table 2 (see Appendix F). Since the other two methods are not intended to run on the CPU, we only compare the GPU (A6000) running times listed in Table 5. It becomes apparent that, including visual tracking, our approach is still around 6 times faster than Dice and 1300 times faster than Decaf.

#### 4.4 User Study

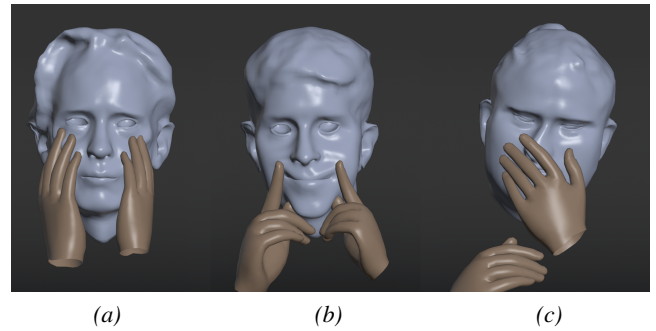
To support the qualitative results, we conducted an online user study with 53 participants from two universities. Each participant watched five random example videos that compared the tracked surfaces, the simulated surfaces of Shimada et al. [SGPT23], and our (neural) approximated surfaces as in Figure 6. The videos are randomly drawn from the sequences in our dataset. Participants were asked to choose the most natural-looking of the three variants for each video. To avoid any bias, we rendered all surfaces in the same color and arranged the variants in random orders. To ensure independent documentation, we used [survio.com](https://survio.com) for the technical implementation.

Figure 11 summarizes the outcome of the user study as the proportion of votes each variant received. Our approach achieved the most votes for all videos by a considerable margin.

#### 5 Limitations

The most significant limitations of our approach result from missing details in the foundational physics-based simulation `phy` as demonstrated in Figure 12. For instance, tracking errors can cause hands to move too deep into the head such that the skull is penetrated. In this case, we consider it more natural to not fully resolve collisions rather than bend bones (Figure 12a). We also do not resolve self-collisions between lips or lips and teeth (Figure 12b). Finally, in our anatomical head model, cartilage components are not sufficiently taken into account, causing the nose or ears to bend a bit too much when the hands push firmly (Figure 12c).

Regarding the efficient approximation of `phy` by the neural network `net`, one can consider a lack of generalization over an extensive set of head shapes as a limitation. However, in contrast to pre-



**Figure 12:** (a) displays remaining collisions due to rigid bones, (b) self-intersections of lips, and (c) a too bendy nose due to missing cartilage.

vious work [SGPT23, WDX\*24] our focus is on personalized head avatars that exhibit a much higher level of detail and authenticity than commonly used head models [LBB\*17, FFBB21]. Moreover, our experiments with multiple head shapes (Section 4.3) indicate generalization capacities of `net`, and the short training time of our approach should be sufficient in most scenarios to train `net` to a given personalized head avatar.

Finally, a greater diversity in our dataset would be desirable. Although we cover a wide range of head shapes with different anatomical compositions, a more diverse coverage of genders and ethnicities would strengthen our results.

#### 6 Conclusion

In this work, we presented NePHIM, a neural physics-based head-hand interaction model. NePHIM extends previous interaction simulations [SGPT23, WDX\*24] with various features such as time-dependent collision paths, pulling of skin, and a higher anatomical precision. Comprehensive experiments and a user study show that our approach is perceived as being considerably closer to reality than the previous state-of-the-art [SGPT23]. Furthermore, we successfully learned a neural approximator of the simulation that allows for rapid inference even on consumer-grade devices.

Nevertheless, we also discussed limitations that provide various starting points for future work. For instance, more detailed anatomical structures and physical properties may enhance the simulation. Moreover, learning the deformation of interactions directly from multi-view videos can contribute to further improvements.

#### Acknowledgments

This research was supported by the German Federal Ministry of Education and Research (BMBF) through the project HiAvA (ID 16SV8785). Open Access funding enabled and organized by Projekt DEAL.

#### References

[ABG\*18] ACHENBACH J., BRYLKA R., GIETZEN T., ZUM HEBEL K., SCHÖMER E., SCHULZE R., BOTSCH M., SCHWANECKE U.: A multi-linear model for bidirectional craniofacial reconstruction. In *Proceedings*

- of the Eurographics Workshop on Visual Computing for Biology and Medicine (2018), pp. 67–76. 3
- [App24] APPLE INC., September 2024. <https://developer.apple.com/augmented-reality/arkit/>. 7
- [AXS\*22] ATHAR S., XU Z., SUNKAVALLI K., SHECHTMAN E., SHU Z.: RigNeRF: Fully controllable neural 3D portraits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 20364–20373. 2
- [BCGF19] BAO M., CONG M., GRABLI S., FEDKIW R.: High-quality face capture using anatomical muscles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 10802–10811. 2
- [BK05] BOTSCH M., KOBBELT L.: Real-time shape editing using radial basis functions. *Computer Graphics Forum* 24, 3 (2005). 3
- [BML\*14] BOUAZIZ S., MARTIN S., LIU T., KAVAN L., PAULY M.: Projective dynamics: fusing constraint projections for fast simulation. *ACM Transactions on Graphics (ToG)* 33, 4 (2014), 1–11. 4
- [BT17] BULAT A., TZIMIROPOULOS G.: How far are we from solving the 2D & 3D face alignment problem?(and a dataset of 230,000 3D facial landmarks). In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 1021–1030. 7, 10, 14
- [CEM\*22] CHOI B., EOM H., MOUSCADET B., CULLINGFORD S., MA K., GASSEL S., KIM S., MOFFAT A., MAIER M., REVELANT M., ET AL.: Animatomy: an Animator-centric, Anatomically Inspired System for 3D Facial Modeling, Animation and Transfer. In *SIGGRAPH Asia Conference Papers* (2022), pp. 1–9. 2
- [CF19] CONG M., FEDKIW R.: Muscle-based facial retargeting with anatomical constraints. In *ACM SIGGRAPH 2019 Talks* (New York, NY, USA, 2019), SIGGRAPH '19, Association for Computing Machinery. 2
- [CO18] CASAS D., OTADUY M. A.: Learning nonlinear soft-tissue dynamics for interactive avatars. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 1 (2018), 1–15. 3
- [Con16] CONG M. D.: *Art-directed muscle simulation for high-end facial animation*. PhD thesis, Stanford University, 2016. 2
- [CZ24] CHANDRAN P., ZOSS G.: Anatomically constrained implicit face models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 2220–2229. 2
- [DWM\*21] DU T., WU K., MA P., WAH S., SPIELBERG A., RUS D., MATUSIK W.: DiffPD: differentiable projective dynamics. *ACM Transactions on Graphics (ToG)* 41, 2 (2021), 1–21. 2
- [FFBB21] FENG Y., FENG H., BLACK M. J., BOLKART T.: Learning an animatable detailed 3D face model from in-the-wild images. *ACM Transactions on Graphics (ToG)* 40, 4 (2021), 1–13. 11
- [GKE\*22] GARBIN S. J., KOWALSKI M., ESTELLERS V., SZYMANOWICZ S., REZAEIFAR S., SHEN J., JOHNSON M., VALENTIN J.: VolTeMorph: realtime, controllable and generalisable animation of volumetric representations. *arXiv preprint arXiv:2208.00949* (2022). 2
- [Han15] HANG S.: Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw* 41, 2 (2015), 11. 3
- [HDDN19] HOLDEN D., DUONG B. C., DATTA S., NOWROUZEZAHRAI D.: Subspace neural physics: fast data-driven interactive simulation. In *Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2019), pp. 1–12. 3, 7
- [IKKP17] ICHIM A.-E., KADLEČEK P., KAVAN L., PAULY M.: Phace: Physics-based face modeling and animation. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 1–14. 2, 4
- [IKNDP16] ICHIM A. E., KAVAN L., NIMIER-DAVID M., PAULY M.: Building and animating user-specific volumetric face rigs. In *Symposium on Computer Animation* (2016), pp. 107–117. 2
- [KGM15] KWOK Y. L. A., GRALTON J., MCLAWS M.-L.: Face touching: a frequent habit that has implications for hand hygiene. *American journal of infection control* 43, 2 (2015), 112–114. 1
- [KK19] KADLEČEK P., KAVAN L.: Building accurate physics-based face models from data. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2, 2 (2019), 1–16. 2
- [LAR\*14] LEWIS J. P., ANJYO K., RHEE T., ZHANG M., PIGHIN F. H., DENG Z.: Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)* 1, 8 (2014), 2. 2
- [LBB\*17] LI T., BOLKART T., BLACK M. J., LI H., ROMERO J.: Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics (ToG)* 36, 6 (2017). 2, 8, 11
- [LFS\*20] LI M., FERGUSON Z., SCHNEIDER T., LANGLOIS T. R., ZORIN D., PANOZZO D., JIANG C., KAUFMAN D. M.: Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Trans. Graph.* 39, 4 (2020), 49. 4
- [MHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118. 3
- [MMG19] MUELLER S. M., MARTIN S., GRUNWALD M.: Self-touch: contact durations and point of touch of spontaneous facial self-touches differ depending on cognitive and emotional load. *PloS one* 14, 3 (2019), e0213677. 1
- [MWSZ24] MA S., WENG Y., SHAO T., ZHOU K.: 3d gaussian blend-shapes for head avatar animation. In *ACM SIGGRAPH 2024 Conference Papers* (2024), SIGGRAPH '24, Association for Computing Machinery. 2
- [PGM\*19] PASZKE A., GROSS S., MASSA F., LERER A., BRADBURY J., CHANAN G., KILLEEN T., LIN Z., GIMELSHEIN N., ANTIGA L., DESMAISON A., KOPF A., YANG E., DEVITO Z., RAISON M., TEJANI A., CHILAMKURTHY S., STEINER B., FANG L., BAI J., CHINTALA S.: Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* 32. Curran Associates, Inc., 2019, pp. 8024–8035. 8
- [PVG\*11] PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COURNAPEAU D., BRUCHER M., PERROT M., DUCHESNAY E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830. 8
- [QKS\*24] QIAN S., KIRSCHSTEIN T., SCHONEVELD L., DAVOLI D., GIEBENHAIN S., NIESSNER M.: Gaussian avatars: photorealistic head avatars with rigged 3D gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 20299–20309. 2
- [RCCO22] ROMERO C., CASAS D., CHIARAMONTE M. M., OTADUY M. A.: Contact-centric deformation learning. *ACM Transactions on Graphics (ToG)* 41, 4 (2022), 1–11. 3
- [RCPO21] ROMERO C., CASAS D., PÉREZ J., OTADUY M.: Learning contact corrections for handle-based subspace dynamics. *ACM Transactions on Graphics (ToG)* 40, 4 (2021), 1–12. 3
- [RMF20] RAHMAN J., MUMIN J., FAKHRUDDIN B.: How frequently do we touch facial t-zone: a systematic review. *Annals of Global Health* 86, 1 (2020). 1
- [RTB17] ROMERO J., TZIONAS D., BLACK M. J.: Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)* 36, 6 (2017). 2, 7
- [SGOC20] SANTESTEBAN I., GARCES E., OTADUY M. A., CASAS D.: SoftSMPL: data-driven modeling of nonlinear soft-tissue dynamics for parametric humans. *Computer Graphics Forum* 39, 2 (2020). 3
- [SGPT23] SHIMADA S., GOLYANIK V., PÉREZ P., THEOBALT C.: Decaf: monocular deformation capture for face and hand interactions. *ACM Transactions on Graphics (ToG)* 42, 6 (2023), 1–16. 1, 2, 3, 4, 8, 9, 10, 11, 14
- [SNF05] SIFAKIS E., NEVEROV I., FEDKIW R.: Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph.* 24, 3 (July 2005), 417425. 2

- [SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Transactions on Graphics (ToG)* 23, 3 (2004). 7
- [SWR\*21] SRINIVASAN S. G., WANG Q., ROJAS J., KLÁR G., KAVAN L., SIFAKIS E.: Learning active quasistatic physics-based models from data. *ACM Transactions on Graphics (ToG)* 40, 4 (2021), 1–14. 3
- [WAB\*20] WENNINGER S., ACHENBACH J., BARTL A., LATOSCHIK M. E., BOTSCH M.: Realistic virtual humans from smartphone videos. In *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology* (2020), pp. 1–11. 7
- [WBS23] WAGNER N., BOTSCH M., SCHWANECKE U.: Softdeca: Computationally efficient physics-based facial animations. In *Proceedings of the 16th ACM SIGGRAPH Conference on Motion, Interaction and Games* (2023), pp. 1–11. 2, 3
- [WDX\*24] WU Q., DOU Z., XU S., SHIMADA S., WANG C., YU Z., LIU Y., LIN C., CAO Z., KOMURA T., ET AL.: Dice: end-to-end deformation capture of hand-face interactions from a single image. *arXiv preprint arXiv:2406.17988* (2024). 2, 3, 4, 10, 11
- [WSB24] WAGNER N., SCHWANECKE U., BOTSCH M.: Anacondar: Anatomically-constrained data-adaptive facial retargeting. *Computers & Graphics* 122 (2024), 103988. 7
- [Xim24] XIMEA., September 2024. <https://www.ximea.com>. 7
- [YKZ\*22] YANG L., KIM B., ZOSS G., GÖZCÜ B., GROSS M., SOLENTHALER B.: Implicit neural representation for physics-driven actuated soft bodies. *ACM Transactions on Graphics (ToG)* 41, 4 (2022), 1–10. 2, 4
- [YZC\*23] YANG L., ZOSS G., CHANDRAN P., GOTARDO P., GROSS M., SOLENTHALER B., SIFAKIS E., BRADLEY D.: An implicit physical face model driven by expression and style. In *SIGGRAPH Asia 2023 Conference Papers* (2023), Association for Computing Machinery. 2
- [ZBT23] ZIELONKA W., BOLKART T., THIES J.: Instant volumetric head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 4574–4584. 2
- [ZBV\*20] ZHANG F., BAZAREVSKY V., VAKUNOV A., TKACHENKA A., SUNG G., CHANG C.-L., GRUNDMANN M.: Mediapipe hands: on-device real-time hand tracking. *arXiv preprint arXiv:2006.10214* (2020). 7, 10, 14

## Appendix

### A Template Dimensions

Mesh	H	J	C	S	J	C
# Vertices	6688	886	4220	11001	899	3354
# Faces / Tets	13372	1768	8444	31456	4190	15634

**Table 6:** The dimensions of all template components in our experiments.

### B Weights & Parameters

$w_{\text{tar}}$	$w_{\text{S}}$	$w_{\text{J}}$	$w_{\text{C}}$	$w_{\text{push}}$	$w_{\text{pull}}$	$w_{\text{corr}}$
$10^2$	$10^1$	$10^4$	$10^4$	$10^2$	$10^2$	$10^2$

**Table 7:** The weights of the physics-based simulations of *phy*.

Proj. Dyn. Iterations	$\alpha$	$l_{\text{min}}$	$r$	$s$	$\Delta\epsilon$
10	0.01	2.5 cm	0.5 cm	50 ms	0.05

**Table 8:** The parameters of the physics-based simulations of *phy*.

### C Ridge Calculation

#### Algorithm 4 Cylinder Ridge

$c$  Cylinder  
 $H$  Head Surface  
 $I$  Indices of vertices that are in  $c$   
 $v_i^H$  Vertex  $i$  of  $H$   
 $\text{len, start, end}$  Length, start, end of a cylinder  
 $\text{plane}(r, n)$  Plane in normal form  
 $\text{mean}$  Mean of vertices  
 $\text{proj}(v, p)$  Project vertex  $v$  on plane  $p$

#### Function `ridge(I, H, c)`

```

// Initialize ridge targets
 $C_{\text{ridge}} = \{\}$ 

// Calculate mean of cylinder
 $r = (\text{start}(c) + \text{end}(c)) / 2$ 

// Calculate normal of cylinder plane
 $n = r - \text{mean}(H)$ 
 $n /= \|n\|$ 

// Calculate cylinder plane
 $p = \text{plane}(r, n)$ 

// Calculate targets
for  $i \in I$  do
    // Calculate plane position
     $v_i^p = \text{proj}(v_i^H, p)$ 

    // Calculate a height offset factor
     $\kappa = \min(\| \text{start}(c) - v_i^p \|, \| \text{end}(c) - v_i^p \|) / (\text{len}(c) / 2)$ 

    // Add ridge target
    Add  $(v_i^p + \kappa \cdot \text{len}(c) \cdot n, i)$  to  $C_{\text{ridge}}$ 

// Return the ridge targets
return  $C_{\text{ridge}}$ 

```

**D Dataset**

Movement	Single Finger		Multiple Fingers		Open Palm		Closed Fist	
	<i>I</i>	<i>II</i>	<i>I</i>	<i>II</i>	<i>I</i>	<i>II</i>	<i>I</i>	<i>II</i>
	<b>Poke / Touch</b>							
Cheeks	-	4	2	3	-	4	-	-
Nose	2	-	-	-	-	-	-	-
Forehead	-	-	1	-	-	-	-	-
Chin	-	-	-	-	-	-	4	-
<b>Pinch / Squeeze</b>								
Lips	-	-	5	-	-	-	-	-
Cheeks	-	-	13	-	-	2	-	3
<b>Rub / Stroke</b>								
B → F	-	2	6	1	1	12	-	1
F → B	-	-	-	2	-	5	-	2
D → U	-	-	-	-	4	7	-	2
U → D	-	2	3	3	2	5	-	1
L → R	-	-	-	-	-	1	-	-
R → L	-	-	-	-	-	1	-	-
Circle	-	2	-	7	-	7	-	-
<b>Punch</b>								
Cheeks	-	-	-	-	6	-	4	3
<b>Pull / Tug</b>								
Lips	1	3	-	-	-	-	-	-
Cheeks	-	7	-	8	-	-	-	-
Nose	2	-	-	-	-	-	-	-
<b>Sum</b>	<b>5</b>	<b>20</b>	<b>33</b>	<b>24</b>	<b>13</b>	<b>44</b>	<b>8</b>	<b>12</b>

**Table 9:** Quantitative description of the captured hand-head interactions. The number of involved hands is indicated by *I* and *II*. A direction is indicated by  $\rightarrow$  where B, F, D, U, L, and R abbreviate back, front, down, up, left, and right, respectively.

**E Additional Simulation Examples**

**Figure 13:** The figure shows examples of our simulation  $\phi_{hy}$  and compares them to the tracked surfaces as well as the simulation of Decaf [SGPT23]. Here, the hands are solely rendered for the tracked meshes to accentuate the simulated deformations.

**F Tracking Network**

Dataset	# Identities	Reconstruction	
		Mean $\ell^2$	Max $\ell^2$
Test	One	0.11 cm	0.29 cm
	Eight	0.14 cm	0.46 cm

**Table 10:** Test errors of the neural approximation net of the simulation  $\phi_{hy}$  paired with Mediapipe [BT17, ZBV\*20] tracking as described in Section 4.3.3.





## CONCLUSION

---

To conclude this thesis, we first concisely summarize our contributions (Section 7.1). Subsequently, we assess their potential impact and offer an outlook on future developments of facial animations in general (Section 7.2).

### 7.1 SUMMARY

In this thesis, we explored the multifaceted challenge of enhancing linear facial animations by integrating physics-based simulation (PBS). More precisely, we tackled four foundational research questions.

1. The first question (Section 1.2.1) we investigated was whether it was feasible to accelerate a state-of-the-art corrective PBS of head anatomy to improve *linear blendshapes* (*LBS*) in real-time. With *SoftDECA* [107] (Chapter 3), we provided a successful answer, which approximates such corrections with the help of a hypernetwork [15] that is generally applicable to manifold head- and blendshapes, and can be executed in less than 10 ms per frame even on consumer-grade CPUs. In addition, we can also manipulate simulation parameters and material properties in the learned approximation and, thus, enable a wide range of applications beyond a pure correction of *LBS* animations. Nonetheless, minor limitations remain, such as the oversimplified modeling of second-order effects and the absence of dynamic external influences.
2. As the second question (Section 1.2.2), we examined whether we can adapt *SoftDECA* to map sparsely tracked facial landmarks to dense facial expressions instead of correcting linear animations. With *SparseSoftDECA* [111] (Chapter 4), we introduced our solution to that question, which again relies on an efficient hypernetwork to approximate a PBS. While this simulation differs only slightly from the original *SoftDECA* simulation, achieving high generalization and ap-

proximation quality was a challenge. Mostly, as personalized tracked landmarks can carry more information than unpersonalized blendshape weights, we needed to implement augmentation strategies for the training data to achieve a comprehensive generalization. *SparseSoftDECA*, in general, enables more detailed facial animations. Yet, the underlying approximated PBS persists as a contingent limitation of realism, as we are still dependent on widely applicable heuristics that do not allow for person-specific details.

3. We addressed the third research question (Section 1.2.3) with *AnaConDaR* [110] (Chapter 5), aiming to enhance solutions for facial retargeting with PBSs. Specifically, we contributed to two subareas of facial retargeting: first, an anatomically more plausible *volumetric deformation transfer (DT)* [105] that facilitates the algorithmic generation of blendshapes *without* exemplary facial expressions of the targeted person. Second, we improved facial retargeting when exemplary expressions of the target *are known* via the simulation of *patchwise LBS* [18]. Especially, the *volumetric DT* proves beneficial within our overall thesis framework by enabling the creation of more authentic blendshapes for *SoftDECA*. Although both contributions of *AnaConDaR* enhance retargeting authenticity according to a user study we conducted, they can only partially compensate for limited access to genuine expressions of the target.
4. In the final research question (Section 1.2.4), we explored whether we can accelerate the simulation of dynamic external effects on heads to be executable in real-time. Specifically, due to their significance for non-verbal communication, with *NePHIM* [112] (Chapter 6), we focused on head-hand interactions. To begin with, we constructed an elaborate 3D video scanner to record exemplary interactions. Subsequently, we developed a simulation capable of realistically depicting the pulling and pushing animations observed in these recordings, even accounting for long-term effects. Finally, we demonstrated that an efficient neural network [39] could learn to approximate this simulation. Admittedly, the demonstration has been conducted only for individual cases so far; generalizing our approach to apply to everyone remains an ongoing task.

In summary, we went various ways to achieve the goal we set in Chapter 1 of an efficient, accessible, but at the same time authentic framework for facial animation. In the subsequent section, we will name a variety of arguments as to why it will be worthwhile to develop the core ideas of our contributions further in the future and assess their potential impact.

## 7.2 OUTLOOK & IMPACT

In the Chapters 3 – 6, we provided direct insights into how our methods can be expanded upon. At this stage, we offer a broader perspective on facial animation, aiming to evaluate the potential impact of our contributions. While inherently speculative, we anticipate advancements across three distinct time horizons.

### CURRENTLY

In the current time, traditional explicit *LBS*-based methods remain predominant in production environments [23], many animation engines [10, 106, 31] support them, and they come with precise tracking solutions [4, 126]. Additionally, a vast pool of digital artists is well-versed in their application, and there are established best practices for beginners [56]. Furthermore, these methods ensure stereoscopic consistency by design, making them highly suitable, alongside their speed, for virtual reality applications.

### MID-TERM

Overtly, research already investigates more advanced techniques, whereby the current focus predominantly centers on implicit radiance field methods [51, 74] for facial animations [90, 70, 33, 87]. However, these methods face several limitations: right now, they can only be executed in real-time on high-performance GPUs, do not adhere to stereoscopic consistency as they are dependent on the viewing direction, and research into artistic manipulations is still nascent [20, 38]. Additionally, such models are typically trained on images or videos of actual people, complicating their application to artificial humanoid characters.

Nonetheless, since the photorealistic results are convincing and the research activities in this field are enormously high, we reckon that they will be able to establish themselves soon and that existing disadvantages will

be quickly resolved. The most recent trend towards hybrid solutions [90, 70] underscores this expectation. For instance, developments that rely on foundational *LBS* systems [70] or 3D morphable models [90] benefit from exploiting the explicit properties of the long-established techniques. In the same spirit, the widespread use of Gaussian splats [51] to accelerate radiance fields is frankly just a revival of point-based rendering techniques [12].

#### LONG-TERM

But of course, fully implicit or hybrid methods in facial animation do not have a clearly defined timeline before becoming fully supported by production systems. Another potential future scenario is that image- or video-based approaches completely bypass traditional computer graphics, utilizing black-box neural networks for generating facial animations in videos [67] or even games [3]. Concepts like *DeepFakes* [95, 34], which animate faces directly in images through deep learning, exemplify this direction. In this context, the achievable level of realism has recently been greatly elevated [115]. The extensive advancements in diffusion-based generative models [96] and, more generally, artificial intelligence [85] suggest a future where everything can be “tokenized” and processed via black-boxes, if endeavored.

Our perspective on developing and integrating such methods is highly speculative, as we can only rely on publicly available research outcomes. Nonetheless, we expect that black-box approaches will take longer to become fully applicable for all facial animation tasks due to challenges similar to those before. Besides artist-friendly control mechanisms and computational efficiency, these are, above all, time consistency and the reproduction quality of personal details.

#### IMPACT

When examining possible developments in facial animation technology as described above, it is plausible to conclude that the wide usage of efficient explicit or hybrid methods will continue in the foreseeable future. These approaches leverage head geometry for animation purposes and are notably prevalent within current production systems. In other words, methods that, in one way or another, rely on the explicit modeling of head geometries. As in this thesis, we improve the most popular

and universal of such methods, *LBS*, with efficient PBSs, we can envision an influential impact of our work. To further amplify this impact, we have, for example, largely integrated *SoftDECA* into a soon-to-be-available open-source virtual reality framework and made parts of the unique *NePHIM* data publicly available at [https://gitlab.cs.hs-rm.de/cvmmr\\_releases/HeadHand](https://gitlab.cs.hs-rm.de/cvmmr_releases/HeadHand). At the same location, the code of the most important components of our simulations is also publicly available. As the topics addressed in this thesis are, in general, largely dominated by non-European companies, independent and non-commercial research such as ours is crucial for promoting digital independence.

Notable recognitions further substantiate the impact of our work. For instance, *SoftDECA* received an honorable mention as the best paper, highlighting its significance and innovation. Further, among others, *Disney Research* [125] recently paid attention to our work. The latter is noteworthy, given that direct insights into private research institutes and production companies are typically limited.

In general, the research findings presented within this thesis have been disseminated through highly regarded academic journals and conferences, ensuring that they reach a broad audience of scholars and practitioners in the related fields. In this spirit, this thesis aims not only to enhance the current state of research in facial animation but also to encourage researchers and users to leverage the many benefits that PBSs offer. Specifically, we hope to have contributed to making PBSs for facial animations more popular in practice and not just known for their theoretically valuable properties.



## OVERVIEW OF PUBLICATIONS

---

### THESIS PUBLICATIONS

#### **SoftDECA: Computationally Efficient Physics-Based Facial Animations**

Nicolas Wagner, Mario Botsch, and Ulrich Schwanecke

Proceedings of the 16th ACM SIGGRAPH Conference on Motion, Interaction, and Games, 2023

DOI: [10.1145/3623264.3624439](https://doi.org/10.1145/3623264.3624439)

© 2023 Copyright held by the authors. Published by the Association for Computing Machinery. This work is licensed under a Creative Commons Attribution 4.0 International license.

#### **SparseSoftDECA: Efficient High-Resolution Physics-Based Facial Animation from Sparse Landmarks**

Nicolas Wagner, Ulrich Schwanecke, and Mario Botsch

Computers and Graphics 119, 2024

DOI: [10.1016/j.cag.2024.103903](https://doi.org/10.1016/j.cag.2024.103903)

© 2024 Copyright held by the authors. Published by Elsevier Ltd. This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International license.

#### **AnaConDaR: Anatomically-Constrained Data-Adaptive Facial Retargeting**

Nicolas Wagner, Ulrich Schwanecke, and Mario Botsch

Computers and Graphics 122, 2024

DOI: [10.1016/j.cag.2024.103988](https://doi.org/10.1016/j.cag.2024.103988)

© 2024 Copyright held by the authors. Published by Elsevier Ltd. This work is licensed under a Creative Commons Attribution 4.0 International license.

**NePHIM: A Neural Physics-Based Head-Hand Interaction Model**

Nicolas Wagner, Ulrich Schwanecke, and Mario Botsch

Computer Graphics Forum 44, 2025

DOI: [10.1111/cgf.70045](https://doi.org/10.1111/cgf.70045)

© 2025 Copyright held by the authors. Published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd. This work is licensed under a Creative Commons Attribution 4.0 International license.

All publications that constitute this cumulative thesis are the work of the same three authors who carried out the same tasks for each publication. With regard to the *CRedit* [81] taxonomy, these are:

- Nicolas Wagner: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Project administration, Software, Validation, Visualization, Writing – original draft, Writing – review & editing
- Ulrich Schwanecke: Conceptualization, Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing
- Mario Botsch: Conceptualization, Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing

OTHER PUBLICATIONS

**Rule extraction from binary neural networks with convolutional rules for model validation**

Sophie Burkhardt, Jannis Brugger, Nicolas Wagner, Zahra Ahmadi, Kristian Kersting, and Stefan Kramer

Frontiers in Artificial Intelligence 4, 2021

DOI: [10.48550/arXiv.2012.08459](https://doi.org/10.48550/arXiv.2012.08459)



**NeuralQAAD: An Efficient Differentiable Framework for Compressing High Resolution Consistent Point Clouds Datasets.**

Nicolas Wagner and Ulrich Schwanecke

Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2022

DOI: [10.5220/0010772500003124](https://doi.org/10.5220/0010772500003124)

**Federated Stain Normalization for Computational Pathology**

Nicolas Wagner, Moritz Fuchs, Yuri Tolkach, and Anirban Mukhopadhyay

International Conference on Medical Image Computing and Computer-Assisted Intervention, 2022

DOI: [10.1007/978-3-031-16434-7\\_2](https://doi.org/10.1007/978-3-031-16434-7_2)



## BIBLIOGRAPHY

---

- [1] Jascha Achenbach, Robert Brylka, Thomas Gietzen, Katja zum Hebel, Elmar Schömer, Ralf Schulze, Mario Botsch, and Ulrich Schwanecke. “A multilinear model for bidirectional craniofacial reconstruction”. In: *Eurographics Symposium on Visual Computing for Biology and Medicine*. 2018, pp. 67–76. DOI: 10.2312/vcbm.20181230.
- [2] Dicko Ali-Hamadi, Tiantian Liu, Benjamin Gilles, Ladislav Kavan, François Faure, Olivier Palombi, and Marie-Paule Cani. “Anatomy transfer”. In: *ACM Transactions on Graphics (ToG)* 32.6 (2013), pp. 1–8. DOI: 10.1145/2508363.2508415.
- [3] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. “Diffusion for world modeling: visual details matter in atari”. In: *Advances in Neural Information Processing Systems*. 2024. DOI: 10.48550/arXiv.
- [4] Apple Inc. <https://developer.apple.com/augmented-reality/arkit/>. Accessed 04.04.2025.
- [5] Michael Bao, Matthew Cong, Stéphane Grabli, and Ronald Fedkiw. “High-quality face capture using anatomical muscles”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10802–10811. DOI: 10.1109/CVPR.2019.011106.
- [6] Vincent Barrielle and Nicolas Stoiber. “Realtime performance-driven physical simulation for facial animation”. In: *Computer Graphics Forum*. Vol. 38. 1. 2019, pp. 151–166. DOI: <https://doi.org/10.1111/cgf.13450>.
- [7] Vincent Barrielle, Nicolas Stoiber, and Cédric Cagniart. “Blendforces: a dynamic framework for facial animation”. In: *Computer Graphics Forum*. Vol. 35. 2. 2016, pp. 341–352. DOI: <https://doi.org/10.1111/cgf.12836>.

- [8] Jan Bender, Kenny Erleben, and Jeff Trinkle. “Interactive simulation of rigid body dynamics in computer graphics”. In: *Computer Graphics Forum*. Vol. 33. 1. 2014, pp. 246–270. DOI: 10.1111/cgf.12272.
- [9] Kiran S. Bhat, Rony Goldenthal, Yuting Ye, Ronald Mallet, and Michael Koperwas. “High fidelity facial animation capture and retargeting with contours”. In: *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2013, pp. 7–14. DOI: 10.1145/2485895.2485915.
- [10] Blender Foundation. <https://www.blender.org>. Accessed 04.04.2025.
- [11] Timo Bolkart, Tianye Li, and Michael J Black. “Instant multi-view head capture through learnable registration”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 768–779. DOI: 0.1109/CVPR52729.2023.00081.
- [12] Mario Botsch and Leif Kobbelt. “High-quality point-based rendering on modern gpu”. In: *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*. 2003, pp. 335–343.
- [13] Mario Botsch, Robert Sumner, Mark Pauly, and Markus Gross. “Deformation transfer for detail-preserving surface editing”. In: *Vision, Modeling & Visualization*. 2006, pp. 357–364.
- [14] Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. “Projective dynamics: fusing constraint projections for fast simulation”. In: *ACM Transactions on Graphics (ToG)* 33.4 (2014), pp. 1–11. DOI: 10.1145/2601097.2601116.
- [15] Christopher Brandt, Elmar Eisemann, and Klaus Hildebrandt. “Hyper-reduced projective dynamics”. In: *ACM Transactions on Graphics (ToG)* 37.4 (2018), pp. 1–13. DOI: 10.1145/3197517.3201387.
- [16] Sophie Burkhardt, Jannis Brugger, Nicolas Wagner, Zahra Ahmadi, Kristian Kersting, and Stefan Kramer. “Rule extraction from binary neural networks with convolutional rules for model validation”. In: 4 (2021), p. 642263. DOI: 10.48550/arXiv.2012.08459.

- [17] George Celniker and Dave Gossard. “Deformable curve and surface finite-elements for free-form shape design”. In: *Proceedings of the 18th Conference on Computer Graphics and Interactive Techniques*. 1991, pp. 257–266.
- [18] Prashanth Chandran, Loic Ciccone, Markus Gross, and Derek Bradley. “Local anatomically-constrained facial performance retargeting”. In: *ACM Transactions on Graphics (ToG)* 41.4 (2022), pp. 1–14. DOI: 10.1145/3528223.3530114.
- [19] Yan Chen, Qing-Hong Zhu, Arie Kaufman, and Shigeru Muraki. “Physically-based animation of volumetric objects”. In: *Proceedings of the Computer Animation*. 1998, pp. 154–160.
- [20] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. “Gaussianeditor: swift and controllable 3d editing with gaussian splatting”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 21476–21485. DOI: 10.1109/CVPR52733.2024.02029.
- [21] Yizhou Chen, Yushan Han, Jingyu Chen, Shiqian Ma, Ronald Fedkiw, and Joseph Teran. “Primal extended position based dynamics for hyperelasticity”. In: *Proceedings of the 16th ACM SIGGRAPH Conference on Motion, Interaction and Games*. 2023, pp. 1–10. DOI: 10.1145/3623264.3624437.
- [22] Yizhou Chen, Yushan Han, Jingyu Chen, Zhan Zhang, Alex Mcadams, and Joseph Teran. “Position-based nonlinear gauss-seidel for quasistatic hyperelasticity”. In: *ACM Transactions on Graphics (ToG)* 43.4 (2024). DOI: 10.1145/3658154.
- [23] Byungkuk Choi, Haekwang Eom, Benjamin Mouscadet, Stephen Cullingford, Kurt Ma, Stefanie Gassel, Suzi Kim, Andrew Moffat, Millicent Maier, Marco Revelant, Joe Letteri, and Karan Singh. “Anatomy: an animator-centric, anatomically inspired system for 3d facial modeling, animation and transfer”. In: *SIGGRAPH Asia Conference Papers*. 2022. DOI: 10.1145/3550469.355539.

- [24] Abdelmouttaleb Dakri, Vaibhav Arora, Léo Challier, Marilyn Keller, Michael J Black, and Sergi Pujades. “On predicting 3d bone locations inside the human body”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 2024, pp. 336–346. DOI: 10.1007/978-3-031-72384-1\_32.
- [25] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. “Acquiring the reflectance field of a human face”. In: *Proceedings of the 27th Conference on Computer Graphics and Interactive Techniques*. 2000, pp. 145–156. DOI: 10.1145/344779.344855.
- [26] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan H Barr. “Dynamic real-time deformations using space & time adaptive sampling”. In: *Proceedings of the 28th Conference on Computer Graphics and Interactive Techniques*. 2001, pp. 31–36.
- [27] ONNX Runtime developers. <https://onnxruntime.ai/>. Accessed 04.04.2025.
- [28] Digital Science & Research Solutions Inc. <https://app.dimensions.ai>. Accessed 04.04.2025.
- [29] Tao Du, Kui Wu, Pingchuan Ma, Sebastien Wah, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. “Diffpd: differentiable projective dynamics”. In: *ACM Transactions on Graphics (ToG)* 41.2 (2021), pp. 1–21. DOI: 10.1145/3490168.
- [30] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. “3d morphable face models—past, present, and future”. In: *ACM Transactions on Graphics (ToG)* 39.5 (2020), pp. 1–38. DOI: 10.1145/3395208.
- [31] Epic Games, Inc. <https://www.unrealengine.com/de>. Accessed 04.04.2025.
- [32] Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. “Learning an animatable detailed 3d face model from in-the-wild images”. In: *ACM Transactions on Graphics (ToG)* 40.4 (2021), pp. 1–13. DOI: 10.1145/3450626.3459936.

- [33] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. “Dynamic neural radiance fields for monocular 4d facial avatar reconstruction”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8649–8658. DOI: 10.1109/CVPR46437.2021.00854.
- [34] Pablo Garrido, Levi Valgaerts, Ole Rehmsen, Thorsten Thormahlen, Patrick Perez, and Christian Theobalt. “Automatic face reenactment”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2014, pp. 4217–4224. DOI: 10.1109/CVPR.2014.537.
- [35] Moritz Geilinger, David Hahn, Jonas Zehnder, Moritz Bächer, Bernhard Thomaszewski, and Stelian Coros. “Add: analytically differentiable dynamics for multi-body systems with frictional contact”. In: *ACM Transactions on Graphics (ToG)* 39.6 (2020), pp. 1–15. DOI: 10.1145/3414685.3417766.
- [36] David Ha, Andrew M Dai, and Quoc V Le. “Hypernetworks”. In: *International Conference on Learning Representations*. 2017.
- [37] David Hahn, Pol Banzet, James M Bern, and Stelian Coros. “Real2sim: visco-elastic parameter estimation from dynamic motion”. In: *ACM Transactions on Graphics (ToG)* 38.6 (2019), pp. 1–13. DOI: 10.1145/3355089.3356548.
- [38] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. “Instruct-nerf2nerf: editing 3d scenes with instructions”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 19740–19750. DOI: 10.48550/arXiv.2303.12789.
- [39] Daniel Holden, Bang Chi Duong, Sayantan Datta, and Derek Nowrouzezahrai. “Subspace neural physics: fast data-driven interactive simulation”. In: *Proceedings of the 18th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2019, pp. 1–12. DOI: 10.1145/3309486.3340245.
- [40] Daniel Holz, Stefan Rhys Jeske, Fabian Lössner, Jan Bender, Yin Yang, and Sheldon Andrews. “Multiphysics simulation methods in computer graphics”. In: *Computer Graphics Forum*. 2025, e70082. DOI: 10.1111/cgf.70082.

- [41] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Fredo Durand. “DiffTaichi: differentiable programming for physical simulation”. In: *International Conference on Learning Representations*. 2020. DOI: 10.48550/arXiv.1910.00935.
- [42] Yuanming Hu, Jiancheng Liu, Andrew Spielberg, Joshua B. Tenenbaum, William T. Freeman, Jiajun Wu, Daniela Rus, and Wojciech Matusik. “Chainqueen: a real-time differentiable physical simulator for soft robotics”. In: *International Conference on Robotics and Automation*. 2019, pp. 6265–6271. DOI: 10.1109/ICRA.2019.8794333.
- [43] Peter Hunter and Andrew Pullan. *FEM/BEM Notes*. 2005.
- [44] Alexandru Eugen Ichim, Ladislav Kavan, Merlin Eléazar Nimier-David, and Mark Pauly. “Building and animating user-specific volumetric face rigs”. In: *Proceedings of the 15th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2016, pp. 107–117. DOI: 10.2312/sca.20161228.
- [45] Alexandru-Eugen Ichim, Petr Kadleček, Ladislav Kavan, and Mark Pauly. “Phace: physics-based face modeling and animation”. In: *ACM Transactions on Graphics (ToG)* 36.4 (2017), pp. 1–14. DOI: 10.1145/3072959.3073664.
- [46] VES Jeffrey Okun and VES Susan Zwerman. *The VES handbook of visual effects: industry standard VFX practices and procedures*. Routledge, 2020. DOI: 10.4324/9781351009409.
- [47] Petr Kadleček and Ladislav Kavan. “Building accurate physics-based face models from data”. In: *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2.2 (2019), pp. 1–16. DOI: 10.1145/3340256.
- [48] James T Kajiya and Brian P Von Herzen. “Ray tracing volume densities”. In: *ACM SIGGRAPH Computer Graphics* 18.3 (1984), pp. 165–174. DOI: 10.1145/964965.808594.
- [49] Marilyn Keller, Vaibhav Arora, Abdelmoultaleb Dakri, Shivam Chandhok, Jürgen Machann, Andreas Fritsche, Michael J Black, and Sergi Pujades. “Hit: estimating internal human implicit tissues from the body surface”. In: *Proceedings of the IEEE/CVF Con-*



- ference on Computer Vision and Pattern Recognition*. 2024, pp. 3480–3490. DOI: 10.1109/CVPR52733.2024.00334.
- [50] Marilyn Keller, Silvia Zuffi, Michael J Black, and Sergi Pujades. “Osso: obtaining skeletal shape from outside”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 20492–20501. DOI: 10.48550/arXiv.2204.10129.
- [51] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. “3d gaussian splatting for real-time radiance field rendering.” In: *ACM Transactions on Graphics (ToG)* 42.4 (2023), pp. 139–1. DOI: 10.1145/3592433.
- [52] Jungmin Kim, Min Gyu Choi, and Young J Kim. “Real-time muscle-based facial animation using shell elements and force decomposition”. In: *Symposium on Interactive 3D Graphics and Games*. 2020. DOI: 10.1145/3384382.3384531.
- [53] Yera Kozlov, Derek Bradley, Moritz Bächer, Bernhard Thomaszewski, Thabo Beeler, and Markus Gross. “Enriching facial blendshape rigs with physical simulation”. In: *Computer Graphics Forum*. Vol. 36. 2. 2017, pp. 75–84. DOI: <https://doi.org/10.1111/cgf.13108>.
- [54] Yen Lee Angela Kwok, Jan Gralton, and Mary-Louise McLaws. “Face touching: a frequent habit that has implications for hand hygiene”. In: *American Journal of Infection Control* 43.2 (2015), pp. 112–114. DOI: 10.1016/j.ajic.2014.10.015.
- [55] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. “Set transformer: a framework for attention-based permutation-invariant neural networks”. In: *International Conference on Machine Learning*. 2019, pp. 3744–3753.
- [56] John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H Pighin, and Zhigang Deng. “Practice and theory of blendshape facial models.” In: *Eurographics (State of the Art Reports)* 1.8 (2014), p. 2. DOI: 10.2312/egst.20141042.
- [57] Bo Li, Lingchen Yang, and Barbara Solenthaler. “Efficient incremental potential contact for actuated face simulation”. In: *SIGGRAPH Asia 2023 Technical Communications*. 2023, pp. 1–4. DOI: 10.1145/3610543.3626161.

- [58] Hao Li, Thibaut Weise, and Mark Pauly. "Example-based facial rigging". In: *ACM Transactions on Graphics (ToG)* 29.4 (2010), pp. 1–6. DOI: 10.1145/1778765.1778769.
- [59] Jiaman Li, Zhengfei Kuang, Yajie Zhao, Mingming He, Karl Bladin, and Hao Li. "Dynamic facial asset and rig generation from a single scan." In: *ACM Transactions on Graphics (ToG)* 39.6 (2020), pp. 215–1. DOI: 10.1145/3414685.3417817.
- [60] Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy R Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. "Incremental potential contact: intersection- and inversion-free, large-deformation dynamics." In: *ACM Transactions on Graphics (ToG)* 39.4 (2020), p. 49. DOI: 10.1145/3386569.3392425.
- [61] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. "Learning a model of facial shape and expression from 4d scans." In: *ACM Transactions on Graphics (ToG)* 36.6 (2017), pp. 194–1. DOI: 10.1145/3130800.3130813.
- [62] Tianye Li, Shichen Liu, Timo Bolkart, Jiayi Liu, Hao Li, and Yajie Zhao. "Topologically consistent multi-view face inference using volumetric sampling". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3824–3834. DOI: 10.1109/ICCV48922.2021.00380.
- [63] Yifei Li, Tao Du, Kui Wu, Jie Xu, and Wojciech Matusik. "Diff-cloth: differentiable cloth simulation with dry frictional contact". In: *ACM Transactions on Graphics (ToG)* 42.1 (2022), pp. 1–20. DOI: 10.1145/3527660.
- [64] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B Tenenbaum, and Antonio Torralba. "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids". In: (2018). DOI: 10.48550/arXiv.1810.01566.
- [65] Tiantian Liu, Adam W Bargteil, James F O'Brien, and Ladislav Kavan. "Fast simulation of mass-spring systems". In: *ACM Transactions on Graphics (ToG)* 32.6 (2013), pp. 1–7. DOI: 10.1145/2508363.2508406.

- [66] Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. “Quasi-newton methods for real-time simulation of hyperelastic materials”. In: *ACM Transactions on Graphics (ToG)* 36.3 (2017), pp. 1–16. DOI: 10.1145/3072959.2990496.
- [67] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. “Sora: a review on background, technology, limitations, and opportunities of large vision models”. In: *arXiv preprint arXiv:2402.17177* (2024).
- [68] Mickaël Ly, Jean Jouve, Laurence Boissieux, and Florence Bertails-Descoubes. “Projective dynamics with dry frictional contact”. In: *ACM Transactions on Graphics (ToG)* 39.4 (2020), pp. 57–1. DOI: 10.1145/3386569.3392396.
- [69] Pingchuan Ma, Peter Yichen Chen, Bolei Deng, Joshua B Tenenbaum, Tao Du, Chuang Gan, and Wojciech Matusik. “Learning neural constitutive laws from motion observations for generalizable pde dynamics”. In: *International Conference on Machine Learning*. 2023, pp. 23279–23300. DOI: 10.48550/arXiv.2304.14369.
- [70] Shengjie Ma, Yanlin Weng, Tianjia Shao, and Kun Zhou. “3d gaussian blendshapes for head avatar animation”. In: *ACM SIGGRAPH Conference Papers*. 2024, pp. 1–10. DOI: 10.1145/3641519.3657462.
- [71] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. “Xpbd: position-based simulation of compliant constrained dynamics”. In: *Proceedings of the 9th International Conference on Motion in Games*. 2016, pp. 49–54. DOI: 10.1145/2994258.2994272.
- [72] C Matthew, KS Bhat, and R Fedkiw. “Art-directed muscle simulation for high-end facial animation”. In: *Proceedings of the 15th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2016, pp. 457–465. DOI: 10.2312/sca.20161229.
- [73] Timo Menzel, Mario Botsch, and Marc Erich Latoschik. “Automated blendshape personalization for faithful face animations using commodity smartphones”. In: *Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology*. 2022, pp. 1–9. DOI: 10.1145/3562939.35656.

- [74] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. “Nerf: representing scenes as neural radiance fields for view synthesis”. In: *European Conference on Computer Vision*. 2020, pp. 405–421. DOI: 10.1145/3503250.
- [75] Aleksandar Milojevic, Daniel Peter, Niko B Huber, Luis Azevedo, Andrei Latyshev, Irena Sailer, Markus Gross, Bernhard Thomaszewski, Barbara Solenthaler, and Baran Gözcü. “Autoskull: learning-based skull estimation for automated pipelines”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 2024, pp. 109–118. DOI: 10.1007/978-3-031-72104-5\_11.
- [76] Masahiro Mori, Karl F MacDorman, and Norri Kageki. “The uncanny valley [from the field]”. In: *IEEE Robotics & Automation Magazine* 19.2 (2012), pp. 98–100. DOI: 10.1109/MRA.2012.2192811.
- [77] Stephanie Margarete Mueller, Sven Martin, and Martin Grunwald. “Self-touch: contact durations and point of touch of spontaneous facial self-touches differ depending on cognitive and emotional load”. In: *PloS one* 14.3 (2019), e0213677. DOI: 10.1371/journal.pone.0213677.
- [78] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. “Position based dynamics”. In: *Journal of Visual Communication and Image Representation* 18.2 (2007), pp. 109–118. DOI: 10.1016/j.jvcir.2007.01.005.
- [79] Matthias Müller, Miles Macklin, Nuttapong Chentanez, Stefan Jeschke, and Tae-Yong Kim. “Detailed rigid body simulation with extended position based dynamics”. In: *Computer Graphics Forum*. Vol. 39. 8. 2020, pp. 101–112. DOI: 10.1111/cgf.14105.
- [80] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. “Instant neural graphics primitives with a multiresolution hash encoding”. In: *ACM Transactions on Graphics (ToG)* 41.4 (2022), pp. 1–15. DOI: 10.1145/3528223.3530127.
- [81] National Information Standards Organization. <https://credit.niso.org>. Accessed 04.04.2025.

- [82] Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, and Mark Carlson. “Physically based deformable models in computer graphics”. In: *Computer Graphics Forum*. Vol. 25. 4. 2006, pp. 809–836. DOI: <https://doi.org/10.1111/j.1467-8659.2006.01000.x>.
- [83] Rhys Newbury, Jack Collins, Kerry He, Jiahe Pan, Ingmar Posner, David Howard, and Akansel Cosgun. “A review of differentiable simulators”. In: *IEEE Access* (2024). DOI: 10.1109/ACCESS.2024.3425448.
- [84] Hayato Onizuka, Diego Thomas, Hideaki Uchiyama, and Rin-ichiro Taniguchi. “Landmark-guided deformation transfer of template facial expressions for automatic generation of avatar blendshapes”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019. DOI: 10.1109/ICCVW.2019.00265.
- [85] OpenAI Inc. <https://openai.com/index/chatgpt/>. Accessed 04.04.2025.
- [86] Hyojoon Park, Sangeetha Grama Srinivasan, Matthew Cong, Doyub Kim, Byungsoo Kim, Jonathan Swartz, Ken Museth, and Eftychios Sifakis. “Near-realtime facial animation by deep 3d simulation super-resolution”. In: *ACM Transactions on Graphics* 43.5 (2024), pp. 1–20. DOI: 10.1145/3670687.
- [87] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. “Nerfies: deformable neural radiance fields”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 5865–5874. DOI: 10.1109/ICCV48922.2021.00581.
- [88] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. “Hypernerf: a higher-dimensional representation for topologically varying neural radiance fields”. In: *ACM Transactions on Graphics (ToG)* 40.6 (2021), pp. 1–12. DOI: 10.1145/3478513.3480487.
- [89] Frederic I Parke and Keith Waters. *Computer facial animation*. CRC press, 2008. DOI: 10.1007/978-1-84628-907-1\_1.

- [90] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. “Gaussianavatars: photorealistic head avatars with rigged 3d gaussians”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 20299–20309. DOI: 10.1109/CVPR52733.2024.01919.
- [91] Yiling Qiao, Junbang Liang, Vladlen Koltun, and Ming Lin. “Differentiable simulation of soft multi-body systems”. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021. DOI: 10.48550/arXiv.2205.01758.
- [92] Dafei Qin, Jun Saito, Noam Aigerman, Thibault Groueix, and Taku Komura. “Neural face rigging for animating and retargeting facial meshes in the wild”. In: *ACM SIGGRAPH Conference Proceedings*. 2023, pp. 1–11. DOI: 10.48550/arXiv.2305.08296.
- [93] Juma Rahman, Jubayer Mumin, and Bapon Fakhruddin. “How frequently do we touch facial t-zone: a systematic review”. In: *Annals of Global Health* 86.1 (2020). DOI: 10.5334/aogh.2956.
- [94] B Raitt. “The making of gollum. presentation at u. southern california institute for creative technologies’s”. In: *Frontiers of Facial Animation* (2004).
- [95] Yurui Ren, Ge Li, Yuanqi Chen, Thomas H Li, and Shan Liu. “Pirenderer: controllable portrait image generation via semantic neural rendering”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13759–13768. DOI: 10.1109/ICCV48922.2021.01350.
- [96] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. “High-resolution image synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10684–10695. DOI: 10.1109/CVPR52688.2022.01042.
- [97] Cristian Romero, Dan Casas, Maurizio Chiaramonte, and Miguel A Otaduy. “Learning contact deformations with general collider descriptors”. In: *SIGGRAPH Asia Conference Papers*. 2023, pp. 1–10. DOI: 10.1145/3610548.3618229.

- [98] Cristian Romero, Dan Casas, Maurizio M Chiaramonte, and Miguel A Otaduy. “Contact-centric deformation learning”. In: *ACM Transactions on Graphics (ToG)* 41.4 (2022), pp. 1–11. DOI: 10.1145/3528223.3530182.
- [99] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. “Learning to simulate complex physics with graph networks”. In: *International Conference on Machine Learning*. 2020, pp. 8459–8468.
- [100] Patrick Schmidt, Dörte Pieper, and Leif Kobbelt. “Surface maps via adaptive triangulations”. In: *Computer Graphics Forum*. Vol. 42. 2. 2023, pp. 103–117. DOI: 10.1111/cgf.14747.
- [101] Soshi Shimada, Vladislav Golyanik, Patrick Pérez, and Christian Theobalt. “Decaf: monocular deformation capture for face and hand interactions”. In: *ACM Transactions on Graphics (ToG)* 42.6 (2023), pp. 1–16. DOI: 10.1145/3618329.
- [102] Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. “Automatic determination of facial muscle activations from sparse motion capture marker data”. In: *ACM SIGGRAPH Papers*. 2005, pp. 417–425. DOI: 10.1145/1073204.1073208.
- [103] Sangeetha Grama Srinivasan, Qisi Wang, Junior Rojas, Gergely Klár, Ladislav Kavan, and Eftychios Sifakis. “Learning active quasistatic physics-based models from data”. In: *ACM Transactions on Graphics (ToG)* 40.4 (2021), pp. 1–14. DOI: 10.1145/3450626.3459883.
- [104] Tuur Stuyck and Hsiao-yu Chen. “Diffxpbd: differentiable position-based simulation of compliant constraint dynamics”. In: *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 6.3 (2023), pp. 1–14. DOI: 10.1145/3606923.
- [105] Robert W Sumner and Jovan Popović. “Deformation transfer for triangle meshes”. In: *ACM Transactions on Graphics (ToG)* 23.3 (2004), pp. 399–405. DOI: 10.1145/1015706.1015736.
- [106] Unity Technologies Inc.  
<https://unity.com/>. Accessed 04.04.2025.

- [107] Nicolas Wagner, Mario Botsch, and Ulrich Schwanecke. “Soft-deca: computationally efficient physics-based facial animations”. In: *Proceedings of the 16th ACM SIGGRAPH Conference on Motion, Interaction, and Games*. 2023, pp. 1–11. DOI: 10.1145/3623264.3624439.
- [108] Nicolas Wagner, Moritz Fuchs, Yuri Tolkach, and Anirban Mukhopadhyay. “Federated stain normalization for computational pathology”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Singapore, Singapore: Springer-Verlag, 2022, pp. 14–23. DOI: 10.1007/978-3-031-16434-7\_2.
- [109] Nicolas Wagner and Ulrich Schwanecke. “Neuralqaad: an efficient differentiable framework for compressing high resolution consistent point clouds datasets.” In: *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. 2022, pp. 811–822. DOI: 10.5220/0010772500003124.
- [110] Nicolas Wagner, Ulrich Schwanecke, and Mario Botsch. “Anacondar: anatomically-constrained data-adaptive facial retargeting”. In: 122.C (Aug. 2024). DOI: 10.1016/j.cag.2024.103988.
- [111] Nicolas Wagner, Ulrich Schwanecke, and Mario Botsch. “Sparsesoftdeca: efficient high-resolution physics-based facial animation from sparse landmarks”. In: 119.C (Apr. 2024). DOI: 10.1016/j.cag.2024.103903.
- [112] Nicolas Wagner, Ulrich Schwanecke, and Mario Botsch. “Nephim: a neural physics-based head-hand interaction model”. In: *Computer Graphics Forum*. Vol. 44. 2025, e70045. DOI: 10.1111/cgf.70045.
- [113] Qisi Wang, Yutian Tao, Eric Brandt, Court Cutting, and Eftychios Sifakis. “Optimized processing of localized collisions in projective dynamics”. In: *Computer Graphics Forum*. Vol. 40. 6. 2021, pp. 382–393. DOI: <https://doi.org/10.1111/cgf.14385>.



- [114] Xiaokun Wang, Yanrui Xu, Sinuo Liu, Bo Ren, Jirí Kosinka, Alexandru C Telea, Jiamin Wang, Chongming Song, Jian Chang, Chenfeng Li, et al. “Physics-based fluid simulation in computer graphics: survey, research trends, and challenges”. In: *Computational Visual Media* 10.5 (2024), pp. 803–858. DOI: 10.1007/s41095-023-0368-y.
- [115] Cong Wei, Bo Sun, Haoyu Ma, Ji Hou, Felix Juefei-Xu, Zecheng He, Xiaoliang Dai, Luxin Zhang, Kunpeng Li, Tingbo Hou, et al. “Mocha: towards movie-grade talking character synthesis”. In: *arXiv preprint arXiv:2503.23307* (2025). DOI: 10.48550/arXiv.2503.23307.
- [116] Stephan Wenninger, Jascha Achenbach, Andrea Bartl, Marc Erich Latoschik, and Mario Botsch. “Realistic virtual humans from smartphone videos”. In: *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology*. 2020, pp. 1–11. DOI: 10.1145/3385956.3418940.
- [117] Stephan Wenninger, Fabian Kemper, Ulrich Schwanecke, and Mario Botsch. “Tailorme: self-supervised learning of an anatomically constrained volumetric human shape model”. In: *Computer Graphics Forum*. Vol. 43. 2. 2024, e15046. DOI: <https://doi.org/10.1111/cgf.15046>.
- [118] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Matthew Johnson, Jingjing Shen, Nikola Milosavljević, Daniel Wilde, Stephan Garbin, Toby Sharp, Ivan Stojiljković, et al. “3d face reconstruction with dense landmarks”. In: *European Conference on Computer Vision*. 2022, pp. 160–177. DOI: 10.1007/978-3-031-19778-9\_.
- [119] Chenglei Wu, Derek Bradley, Markus Gross, and Thabo Beeler. “An anatomically-constrained local deformation model for monocular face capture”. In: *ACM Transactions on Graphics (ToG)* 35.4 (2016), pp. 1–12. DOI: 10.1145/2897824.2925882.
- [120] Qingxuan Wu, Zhiyang Dou, Sirui Xu, Soshi Shimada, Chen Wang, Zhengming Yu, Yuan Liu, Cheng Lin, Zeyu Cao, Taku Komura, et al. “Dice: end-to-end deformation capture of hand-face interactions from a single image”. In: *Poster at the International Conference on Learning Representations*. 2025. DOI: 10.48550/arXiv.2406.17988.

- [121] Zangyueyang Xian, Xin Tong, and Tiantian Liu. “A scalable galerkin multigrid method for real-time simulation of deformable objects”. In: *ACM Transactions on Graphics (ToG)* 38.6 (2019), pp. 1–13. DOI: 10.1145/3355089.3356486.
- [122] Feng Xu, Jinxiang Chai, Yilong Liu, and Xin Tong. “Controllable high-fidelity facial performance transfer”. In: *ACM Transactions on Graphics (ToG)* 33.4 (2014), pp. 1–11. DOI: 10.1145/2601097.2601210.
- [123] Lingchen Yang, Byungsoo Kim, Gaspard Zoss, Baran Gözcü, Markus Gross, and Barbara Solenthaler. “Implicit neural representation for physics-driven actuated soft bodies”. In: *ACM Transactions on Graphics (ToG)* 41.4 (2022), pp. 1–10. DOI: 10.1145/3528223.3530156.
- [124] Lingchen Yang, Gaspard Zoss, Prashanth Chandran, Paulo Gotardo, Markus Gross, Barbara Solenthaler, Eftychios Sifakis, and Derek Bradley. “An implicit physical face model driven by expression and style”. In: *SIGGRAPH Asia Conference Papers*. 2023, pp. 1–12. DOI: 10.1145/3610548.3618156.
- [125] Lingchen Yang, Gaspard Zoss, Prashanth Chandran, Markus Gross, Barbara Solenthaler, Eftychios Sifakis, and Derek Bradley. “Learning a generalized physical face model from data”. In: *ACM Transactions on Graphics (ToG)* 43.4 (2024), pp. 1–14. DOI: 10.1145/3658189.
- [126] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. “Mediapipe hands: on-device real-time hand tracking”. In: *arXiv preprint arXiv:2006.10214* (2020).
- [127] ChangAn Zhu and Chris Joslin. “A review of motion retargeting techniques for 3d character facial animation”. In: *Computers & Graphics* (2024), p. 104037. DOI: 10.1016/j.cag.2024.104037.
- [128] Marc Zuckerberg. *Founders Letter*. <https://about.fb.com/news/2021/10/founders-letter/>. Accessed 04.04.2025.