Interactive Facial Animation: Enhancing Facial Rigs With Real-Time Shell And Contact Simulation

JOSÉ ANTONIO FERNÁNDEZ-FERNÁNDEZ, RWTH Aachen University, Germany and Meta, Switzerland
RYAN GOLDADE, Meta, Switzerland
LADISLAV KAVAN, Meta, Switzerland
JAN BENDER, RWTH Aachen University, Germany
PHILIPP HERHOLZ, Meta, Switzerland

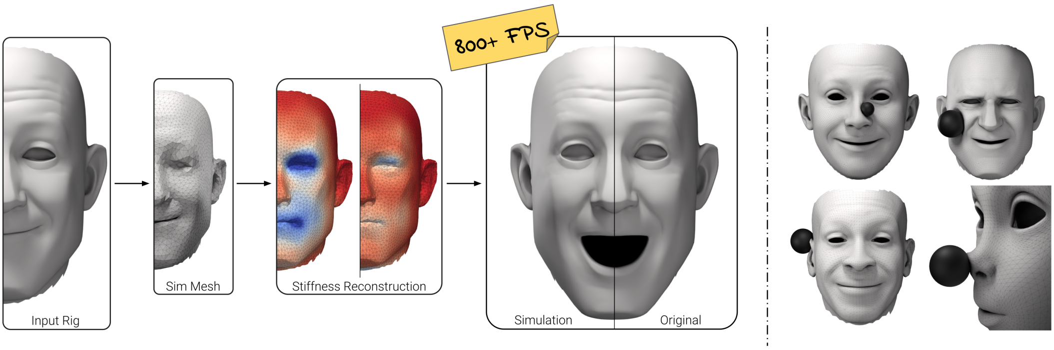

Fig. 1. Main stages of the Interactive Facial Animation pipeline and example contact interactions on four different characters. From left to right: The input is a face rig containing a surface mesh of the face and a set of facial expressions. A shell simulation mesh is obtained by decimating the input surface to a user-defined target number of triangles. Quasistatic inverse simulation is used to find suitable non-uniform bending and strain stiffness so that the forward shell simulation can reproduce the original expressions within a strict small tolerance. Using this stiffness, the upsampled forward shell simulation is able to closely reproduce the original rig output while enabling dynamic effects and user-driven contact interactions.

Demand for high-quality virtual 3D characters continues to grow in both entertainment and communication, and with it, the desire for interactive avatars. This presents unique challenges for high-fidelity interactive solutions that must be quick to set up, efficient to run in real-time on consumer hardware, and work well, not only on human characters, but also on fantastical and heavily stylized ones.

We present Interactive Facial Animation (IFA), a novel approach to augment facial animation rigs with shell simulation and contact without requiring anatomical priors or artist intervention. Our method uses an efficient shell model that tracks the output of a face rig in real-time and incorporates realistic contact. By designing our shell model to be differentiable, we can use inverse simulation to reconstruct realistic non-uniform stiffness that captures both stiff (ears, nose) and soft (cheeks, lips) areas, based on the rig's

---

Authors' Contact Information: José Antonio Fernández-Fernández, RWTH Aachen University, Aachen, Germany and Meta, Zurich, Switzerland, fernandez@cs.rwth-aachen.de; Ryan Goldade, Meta, Zurich, Switzerland, ryangoldade@meta.com; Ladislav Kavan, Meta, Zurich, Switzerland, lkavan@meta.com; Jan Bender, RWTH Aachen University, Aachen, Germany, bender@cs.rwth-aachen.de; Philipp Herholz, Meta, Zurich, Switzerland, herholz@meta.com.

ACM 2577-6193/2025/8-ART58
https://doi.org/10.1145/3747860

deformation space. The reconstruction process imposes hard limits on acceptable errors, which ensures that IFA preserves the resolution and expressiveness of the source material, a common shortcoming in many face simulation pipelines. With performance in mind, we demonstrate IFA using a variant of the Fast Mass-Springs method. However, the presented concept can be realized with any differentiable physics solver that provides reliable convergence.

We showcase our method in four different character rigs, achieving simulation update runtimes between 0.39 and 1.27 milliseconds (787-2564 FPS) on a single CPU thread. Unlike existing methods that require lengthy preparation times, our pipeline can be ready to use within minutes, also on the CPU.

CCS Concepts: • **Computing methodologies → Physical simulation**.

Additional Key Words and Phrases: Face Rigs, Facial Interaction, Physically-Based Simulation.

ACM Reference Format:
José Antonio Fernández-Fernández, Ryan Goldade, Ladislav Kavan, Jan Bender, and Philipp Herholz. 2025. Interactive Facial Animation: Enhancing Facial Rigs With Real-Time Shell And Contact Simulation. *Proc. ACM Comput. Graph. Interact. Tech.* 8, 4, Article 58 (August 2025), 21 pages. https://doi.org/10.1145/3747860

1 Introduction

The transition toward highly realistic and interactive virtual experiences is evident in communication, entertainment, training, and other applications. Telepresence, in particular, has seen significant advances in the visual fidelity of virtual faces [Li et al. 2024; Lombardi et al. 2018; Ma et al. 2021]. The next major milestone in immersion lies in enabling realistic user interactions through contact with these digital faces — whether for virtual character engagement in a video game or touching one's own face during a call. Without properly addressing these physical interactions, such experiences remain confined to *"look-but-don't-touch"*, thus limiting the sense of true engagement. However, existing methods that integrate contact physics with high-fidelity rigs often face one or more of the following limitations: (a) insufficient performance for strict real-time requirements, (b) limited reproducibility mechanisms that fail to maintain the rig's resolution and expressiveness, or (c) dependence on artist intervention or per-identity anatomical priors (e.g., skull or mandible systems).

Much of these experiences run on affordable consumer hardware. Moreover, face simulation is just one component of a broader real-time system that includes rendering, networking, audio, and more. With this resource-constrained environment in mind, in this work we aim to provide a contact simulation solution for facial animation that can operate at a rate of one millisecond per physics simulation update, ensuring compatibility with high refresh rates such as those often found in VR. Beyond runtime efficiency, setup time is also critical; while users may tolerate a few minutes the first time they set up a character, multi-hour waiting periods should be avoided.

In addition to performance considerations, the preservation of the resolution and expressiveness of high-quality rigs during simulation remains a priority. Many of the existing simulation solutions address contact by replacing the original rig geometry with simulation-ready models, e.g. using anatomically aware human head templates. Such replacements generally do not feature rigorous mechanisms for fidelity preservation and may also perform poorly with non-human characters [Yang et al. 2024].

Our method, Interactive Facial Animation (IFA), is a novel real-time facial animation pipeline that augments existing face rigs with shell mechanics and contact, preserves their expressiveness, and requires no anatomical priors. The pipeline is fully automatic and does not require artist intervention, enabling us to scale the method and to seamlessly integrate it into existing pipelines. While our approach is "hands-free", it is still possible to adjust the behavior of the physical simulation, when desired, by tailoring parameters such as stiffness, density, or damping, to specific identity or scenes. However, we demonstrate that a single parameter set works well across multiple characters and resolutions.

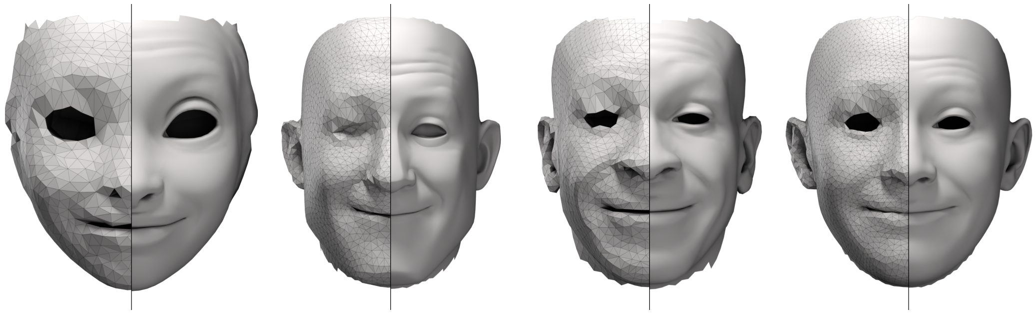

Fig. 2. Split screen of the simulation shell mesh (left) and original input mesh (right) for four different identities and resolutions. From left to right, the number of simulation triangles and character names are: 2k (Aura), 6k (Bowen), 4k (Jupiter) and 8k (Proteus).

Our design decisions are guided by strict real-time performance requirements. The only inputs to our method are a face mesh with the desired render resolution, a set of facial expressions representative of the rig's expressive range, and simulation parameters. In this work, we use classical blendshape rigs to conveniently provide both the mesh and expressions in an industry-standard format.

At the core of our method is a surface-only shell simulation that tracks the input rig through actuation forces while providing passive elasticity and contact handling. This two-dimensional discretization offers a compact representation of the face geometry, enabling efficient simulation of strain and bending directly on the skin surface. Furthermore, in order to maintain a bounded runtime, we operate on a decimated version of the input face mesh with a user-specified number of triangles (see Figure 2). For a given target expression and potential objects in contact, the simulation calculates an equilibrium for the actuation, passive elasticity, and contact on the coarse mesh. We then warp the full-resolution mesh from neutral pose with the resulting displacements and add back fine details lost during decimation. Consequently, the output of our pipeline retains the rig's original fidelity while enabling contact-aware effects.

A key challenge is ensuring that the simulation can reproduce the entire range of input facial expressions since only certain combinations of actuation and passive shell stiffness can achieve these target expressions while responding realistically to contact. To address this, we formulate an inverse simulation problem with an upper limit on actuation forces and a strict reproducibility tolerance for the simulation to obtain the input facial expressions. Solving for non-uniform distributions of strain and bending stiffness yields a material parametrization of the shell that can be safely used in the forward simulation pipeline in terms of reproducibility. Note that while the forward simulation incorporates dynamics, the inverse problem is solved in the quasistatic regime, which reduces the computational requirements to obtain a material parametrization. Finally, having a known actuation limit allows us to cap external contact forces accordingly, preventing unnatural large deformations during user interactions.

Our contributions are:

- A novel inverse quasistatic simulation procedure to recover non-uniform surface stiffness distributions from facial expression geometry, imposing strong reproducibility guarantees.
- A novel forward simulation pipeline that incorporates dynamics, shell mechanics, and contact into face rigs, with update times of around one millisecond on the CPU, all while preserving fine-detail fidelity.

To the best of our knowledge, our method is the first to simultaneously fulfill these requirements regarding reproducibility, automation and performance.

2 Related Work

Facial animation is a long-established field spanning several decades. It encompasses numerous topics, including capture, retargeting, and animation, among others. In what follows, we first summarize physically-based face simulation approaches in the context of facial animation. We then discuss existing works that specifically address contact with external objects.

2.1 Face Simulation

Early work on face animation dates back to the 1970s [Parke 1972]. Physically-based face simulation followed in subsequent decades [Lee et al. 1995; Platt and Badler 1981; Waters 1987], leading to the seminal work by Sifakis et al. [2005], who introduced an identity-specific anatomical human head model that uses a nonlinear constitutive model with FEM and muscle-based actuation, which is fit from MRI data. Simulating digital faces with expression ranges depicted by blendshape rigs [Lewis et al. 2014] has become an active field of research, with the common goal to improve the realism via simulation by incorporating physically-based effects. Several artist-friendly pipelines have been presented [Choi et al. 2022; Cong et al. 2016; Kozlov et al. 2017] to give artists fine control to the facial expressions while still being fully simulated. Surface-only multi-scale approaches are proposed by Bickel et al. [2007, 2008] to simulate detailed face geometries from a rest pose and marker displacements using a coarse shell model for global deformation and an additional fine-detail solver for wrinkles. A surface-only force-based model is proposed by Barrielle et al. [2016] to add dynamics to blendshape rigs. Ichim et al. [2016] present a volumetric FEM model with an interior bone structure that is strain-actuated and that can incorporate dynamics and contact, later improved by adding explicit muscles, muscle sliding and other effects [Ichim et al. 2017]. Several works also focused on material property estimation from capture [Kadleček and Kavan 2019; Kozlov et al. 2019], with focus on realistic deformations on different regions of the face. More recent work by Chandran and Zoss [2023] proposed an anatomical implicit face model that can reconstruct a dense anatomical substructure. Different methods have been introduced for real-time simulation of the face [Barrielle and Stoiber 2019; Kim et al. 2020]. A reduced actuation model [Romeo and Schvartzman 2020], a clustered muscle scheme [Romeo and Schvartzman 2020] and neural approaches [Srinivasan et al. 2021; Yang et al. 2022] have been proposed to address the shortcomings of representing actuation on dense meshes. Neural networks also have been used for expression and style disentanglement [Yang et al. 2023] and for generalized physical faces that can be quickly fit to unseen identities [Yang et al. 2024]. Neural physics approximations have been used in the context of low computation environments to achieve real-time simulation of anatomical face models [Wagner et al. 2023, 2024, 2022].

Contact aware hand-face tracking has been recently addressed by incorporating collision resolution to captured data before training [Shimada et al. 2023; Wu et al. 2024]. Super-resolution strategies have also been explored to drive a high resolution anatomical model of a face by simulating a coarser counterpart [Park et al. 2024] at near real-time rates. Most recently, Zhu and Joslin [2024], proposed a surface-only head template, featuring skull and deformable active and passive patches, that uses PBD to solve for actuation, inertia and contact and that can run real-time on consumer hardware. From this broad body of literature, we focus on methods that aim to handle contact with external objects, discussed below.

2.2 Facial Contact Interaction

We now frame our proposed solution in the context of existing methods that showcase contact interactions. It is important to emphasize that employing a physically-based face model (anatomical or otherwise) is not a guarantee by itself to produce realistic behavior under external contacts or collisions. For example, some of the aforementioned methods use physically-based constraints to deform a mesh in order to match a series of targets; however, this goal presents a very different proposition from that of contact. Specifically, a face can be unrealistically too soft or too stiff upon contact, but still be able to correctly deform into a target pose given some actuation mechanisms. In lieu of this, we consider methods that handle contact to be those which show actual contact with external objects in the reference material. We provide a high level comparison in Table 1.

When tasked with producing a sequence of contact interactions with an input rig, no method guarantees preservation of the input fidelity in full resolution since they do not feature strict reproducibility mechanisms. Methods that use head templates do not provide procedures to recover geometric detail lost in the transfer from input rig to simulation rig [Ichim et al. 2017; Sifakis et al. 2005; Wagner et al. 2024; Zhu and Joslin 2024]. Methods that rely on external tracking pipelines are constrained by these pipeline expression outputs, including resolution [Shimada et al. 2023; Wu et al. 2024]. All methods require anatomical priors either because they fit a head template (simulation methods) or because they train on human data (tracking methods) and therefore they can only support such anatomies. Finally, only two of the methods can in run real-time on affordable consumer hardware [Wagner et al. 2024; Zhu and Joslin 2024]. We then establish that, to the best of our knowledge, there is no solution available in the literature that satisfies our requirements of performance, reproducibility and not requiring anatomic priors.

Table 1. Qualitative comparison of existing contact-aware methods with respect to three key requirements: real-time capability on affordable consumer hardware, ability to reproduce the high-resolution input rig, and absence of anatomical priors or training on human-specific data.

	Real-Time	Reproducibility	Priors Free
Ours	Yes	Yes	Yes
Sifakis et al. [2005]	No	No	No
Ichim et al. [2017]	No	No	No
Shimada et al. [2023]	No	No	No
Wu et al. [2024]	No	No	No
Wagner et al. [2024]	Yes	No	No
Park et al. [2024]	No	No	No
Zhu and Joslin [2024]	Yes	No	No

3 Forward Simulation Pipeline

IFA employs a coarse shell simulation that uses actuation forces to track the input rig, incorporates passive strain and bending elasticity to model deformations, and handles contact. Given a target expression, an external object in contact with the face, and a material stiffness distribution, the simulation produces contact-aware skin deformation for that facial expression. A key feature of our solution is the use of an *actuation pressure field* with an upper limit. This enables high enough contact forces to deform the shell away from the target facial expression and thus yield realistic skin-level deformations. The simulation is differentiable, a requirement of the stiffness reconstruction procedure discussed in the next section.

For the remainder of this document, we assume that the face rig to be enhanced is represented by a blendshape rig [Lewis et al. 2014], although this is not a strict requirement of the proposed concept. Any rigging approach able to provide a set of representative facial expressions for the material estimation stage, and a sequence of targets during runtime for the forward pipeline is compatible with our method. A blendshape rig produces a face mesh with a target expression $\mathbf{x}$ via linear interpolation of a set of deformation modes (or *blendshapes*) as

$$\mathbf{x} = \mathbf{X} + \sum_{b=1}^{n_b} \mathbf{u}_b \, w_b, \tag{1}$$

where n_b is the number of blendshapes, $\{\mathbf{u}\}$ is the set of blendshapes and $\{w\}$ are corresponding interpolation weights that define the expression. Note that this is a purely geometric approach without the involvement of physics.

With this context, IFA's forward simulation pipeline proceeds in three stages:

(1) A coarse version of the rig uses the expression input (blendshape weights $\mathbf{w}$) to generate a coarse surface target expression.
(2) Actuation forces drive the simulation shell toward this target and the dynamic simulation is solved, including actuation, passive elasticity and contact.
(3) The original resolution face mesh in neutral pose is warped using the resulting coarse displacement field, followed by a fine-detail correction step to recover small-scale features. If fine-detail reproducibility is not necessary, the latter step can be skipped for extra performance gains.

IFA can be used with any shell solver, provided it (a) offers convergence guarantees to ensure reproducibility and (b) is differentiable so that we can perform stiffness reconstruction via inverse simulation. In this work, we present an implementation using an efficient differentiable linear elasticity solver with constant second-order derivatives, based on the Fast Mass-Springs [Liu et al. 2013] method. This elasticity solver is highly efficient thanks to a precomputed Cholesky factorization of the global stiffness matrix, a well-known technique in simulation in computer graphics [Bouaziz et al. 2014; Kugelstadt et al. 2018; Liu et al. 2013; Narain et al. 2016]. More sophisticated combinations of discretization, constitutive models, and solvers could be employed, e.g., Neo-Hookean elasticity on a triangulated surface solved with Projective Dynamics [Bouaziz et al. 2014]. Though such an approach might yield improved material response compared to a spring-based model, its higher per-element cost due to material evaluation and other discretization considerations, such as per-element rotation extraction, can noticeably reduce the framerate despite remaining real-time. Given our emphasis on performance, we present our pipeline using the spring-based approach.

Finally, our method can be categorized as a two-scale solution, where the main simulation runs at a coarse resolution and finer details are recovered afterward. While this design has parallels to existing multi-scale methods [Bickel et al. 2007, 2008], note that our fine-detail correction aims to geometrically match the input data, rather than introduce new features via simulation or other means.

3.1 Shell Simulation

Our dynamic system is governed by the equations of motion

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{f}(\mathbf{x}) = -\frac{\partial \Psi^p(\mathbf{x})}{\partial \mathbf{x}}, \quad \dot{\mathbf{x}} = \mathbf{v}, \tag{2}$$

where $\mathbf{x}$ is the vector of node positions, $\mathbf{v}$ the velocity, $\mathbf{M}$ the lumped mass matrix (with surface density ρ), and $\Psi^p(\mathbf{x})$ the total potential energy. We employ an optimization-based time-integration scheme to evolve the positions of the simulation mesh [Martin et al. 2011]. Each time step Δt is advanced via backward Euler by minimizing

$$\Psi(\mathbf{x}) = \frac{1}{2\,\Delta t^2}\left\|\mathbf{M}^{\frac{1}{2}}(\mathbf{x} - \tilde{\mathbf{x}})\right\|_2^2, + \Psi^p(\mathbf{x}) \tag{3}$$

with respect to $\mathbf{x}$, given $\mathbf{v} = (\mathbf{x} - \mathbf{x}^n)/\Delta t$, where $\tilde{\mathbf{x}} = \mathbf{x}^n + \Delta t\,\mathbf{v}^n$ is the inertial estimate from the current state $(\mathbf{x}^n, \mathbf{v}^n)$. We decompose Ψ^p as

$$\Psi^p = \Psi^a + \Psi^s + \Psi^b + \Psi^d, \tag{4}$$

where Ψ^a is the actuation potential, Ψ^s accounts for in-plane strain (mass-springs), Ψ^b handles thin-shell bending, and Ψ^d represents inertial damping.

Actuation. We incorporate actuation forces to guide the mesh toward user-specified facial expressions via

$$\Psi^a = \sum_{i=1}^{n_v} \frac{1}{2}\frac{p^{\max} A_i}{\epsilon^r}\left\|\bar{\mathbf{x}}_i^t - \mathbf{x}_i\right\|_2^2, \tag{5}$$

where n_v is the number of vertices, $\bar{\mathbf{x}}_i^t$ is the (clamped) target position of vertex i, $\mathbf{x}_i$ its simulated position, $p^{\max}$ the maximum actuation pressure, A_i the Voronoi area of the vertex in the rest pose, and ϵ^r the reproducibility tolerance. The term $\frac{p^{\max} A_i}{\epsilon^r}$ sets the per-vertex actuation stiffness, corresponding to a maximum force $p^{\max} A_i$ at a distance ϵ^r. To ensure that the actuation force never exceeds this bound while maintaining a constant global Hessian, we do *not* directly use the vertices $\mathbf{x}^t$ from the coarse rig. Instead, for each component $j \in \{1, \ldots, 3n_v\}$, we clamp:

$$\bar{\mathbf{x}}_j^t = \mathbf{x}_j + \max\left(-\epsilon^r, \min\left(\mathbf{x}_j^t - \mathbf{x}_j, \epsilon^r\right)\right), \tag{6}$$

thus pulling the shell toward $\mathbf{x}^t$ without exceeding a pressure field of $p^{\max}$. As a result, actuation forces remain significant within ϵ^r but crucially, do not grow unbounded when large deviations occur due to contact, which would result in unnaturally large forces, e.g. when bending an ear. Since this bound is only required for deformations beyond ϵ^r, it is not necessary during material reconstruction and therefore does not affect differentiability.

Strain. We model in-plane elasticity via springs using the edge-based potential

$$\Psi^s = \sum_{i=1}^{n_e} \frac{1}{2} l_i^r \kappa_i^s \left\|\mathbf{p}_{i1} - \mathbf{p}_{i2} - \mathbf{d}_i\right\|_2^2, \tag{7}$$

for edge i connecting points $\mathbf{p}_{i1}$ and $\mathbf{p}_{i2}$, n_e the number of edges, l_i^r the edge's rest length and κ_i^s its strain stiffness. $\mathbf{d}_i$ is the target distance vector

$$\mathbf{d}_i = \left\|\mathbf{p}_{i1}^r - \mathbf{p}_{i2}^r\right\|_2 \frac{\hat{\mathbf{p}}_{i1} - \hat{\mathbf{p}}_{i2}}{\left\|\hat{\mathbf{p}}_{i1} - \hat{\mathbf{p}}_{i2}\right\|_2}, \tag{8}$$

where $\mathbf{p}^r$ are rest positions, and $\hat{\mathbf{p}}$ are per-time-step-constant (or *lagged*) positions.

Bending. We add a bending potential Ψ^b for each internal edge (or dihedral) following Volino and Magnenat-Thalmann [2006]

$$\Psi^b = \sum_{i=1}^{n_d} \frac{1}{2}\frac{\|\mathbf{e}_i^0\|_2}{\bar{h}_i} \kappa_i^b \left\|\mathbf{q}_{i1} - \mathbf{q}_{i2} - \mathbf{d}_i\right\|_2^2, \tag{9}$$

where n_d is the number of dihedrals, $\mathbf{e}_i^0$ the common edge at rest pose, $\bar{h}_i$ is a third of the average of the heights of the two triangles incident to the common edge [Grinspun et al. 2003], and κ_i^b the bending stiffness. The points $\mathbf{q}_{i1}$ and $\mathbf{q}_{i2}$ are the closest points between the common edge and the line that connects the other two vertices of the dihedral, computed at rest pose.

Damping. We penalize nodal momentum to reduce spurious long-lasting oscillations using

$$\Psi^d = \frac{1}{2\,\Delta t} v \left\| \mathbf{M}^{\frac{1}{2}} (\mathbf{x} - \mathbf{x}^n) \right\|_2^2, \tag{10}$$

where v is a small damping coefficient.

Time Step Solve. Each time step requires solving

$$\frac{\partial^2 \Psi}{\partial \mathbf{x}^2} \Delta x = -\frac{\partial \Psi}{\partial \mathbf{x}}, \tag{11}$$

where Δx is the displacement update. Because our chosen potentials have constant second derivatives, the global Hessian $\frac{\partial^2 \Psi}{\partial \mathbf{x}^2}$ is constant and hence is factorized via Cholesky before the simulation begins. At runtime, only the gradient $\frac{\partial \Psi}{\partial \mathbf{x}}$ must be updated, making each solve a straightforward forward/backward substitution. This *direct* solve guarantees stable and accurate results, suitable for the reproducibility requirement.

Time Sub-stepping. We split large time steps into sub-steps [Macklin et al. 2019] and perform exactly one elastic solve in each. For the per-time-step-constant positions, we use a blend between the current configuration and the explicitly integrated positions

$$\hat{\mathbf{x}} = \alpha^{\mathrm{e}}\, \tilde{\mathbf{x}} + (1 - \alpha^{\mathrm{e}})\, \mathbf{x}^n, \tag{12}$$

where $\alpha^{\mathrm{e}} = \frac{1}{2}$. For our application, this approach results in significantly better inertial behavior, i.e. less numerical damping, than performing multiple elastic iterations on a large time step size, at the same computational cost.

Constraints. We incorporate contact and maximum depth constraints into our forward simulation solver. To avoid refactoring the global matrix when new constraints become active, we do not incorporate constraints into the global elastic solve; instead, we opt for direct positional corrections instead [Müller et al. 2007]. This approximation works well in practice for our application and suits our strict computational budget.

Contacts are detected using analytic Signed Distance Fields (SDF), but discretized fields can also be used [Koschier et al. 2017]. Maximum depth constraints limit how far in the negative normal direction each vertex can deviate from the target, which approximates a bone substructure. After each elastic solve, vertices of the simulation mesh are projected to the closest feasible point from their invalid position. An approximated correction force is calculated based on the acceleration required to perform such correction, which can be used for further actions, such as controlling colliding object dynamics. Finally, velocities are determined considering the initial and final (corrected) vertex positions and the time step size.

3.2 Warping and Fine-Detail Correction

After each simulation step, we obtain a *coarse displacement field* $\mathbf{u} = \mathbf{x} - \mathbf{X}$, where $\mathbf{x}$ are the simulated node positions and $\mathbf{X}$ their rest state. To recover high-resolution output, we *warp* the original rig face mesh $\mathbf{X}_o$ by interpolating $\mathbf{u}$ onto the fine mesh, thereby transferring simulated contact deformations while retaining the rig's overall resolution. However, the coarse simulation mesh cannot capture details finer than its resolution, therefore some high-frequency features contained in the blendshapes (e.g., fine wrinkles) may be lost. Figure 3 illustrates this issue for a frowning expression, where subtle creases in the nose bridge are missing in the warped result.

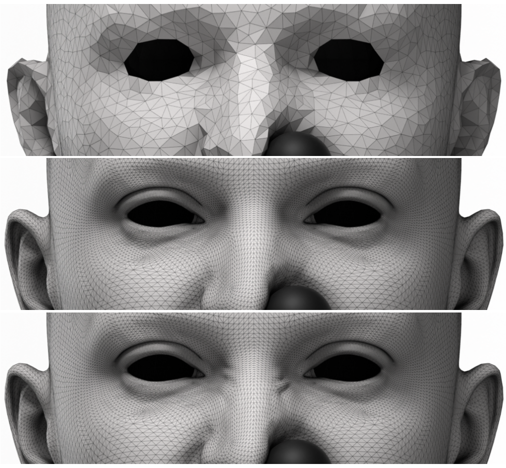

Fig. 3. Warping and fine-detail correction for a frowning expression with contact. *Top*: coarse simulation result. *Middle*: warped original mesh without fine-detail correction. *Bottom*: final mesh after applying the corrections, restoring wrinkles at the nose bridge.

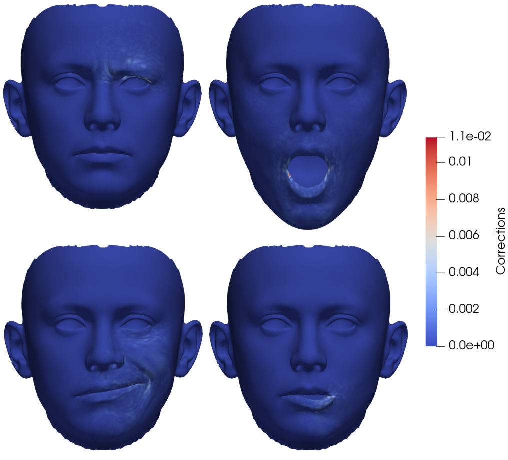

Fig. 4. Deviations between the warped mesh and the original rig output for four blendshapes of the Proteus 8k rig. The maximum difference for the entire rig, 0.032 m, occurs at the eyelids.

Since the target high-resolution geometry $\mathbf{x}_o^t$ is known from the original rig, we can precompute a *detail correction* $\mathbf{u}_c$ that, when added back after warping, reintroduces the transient small-scale features in the final surface. In blendshape rigs, we define a correction $\mathbf{u}_{c,b}$ for each blendshape b:

$$\mathbf{u}_{c,b} = \mathbf{x}_{o,b} - \mathbf{x}_{w,b}, \tag{13}$$

where $\mathbf{x}_{o,b}$ is the original blendshape geometry, and $\mathbf{x}_{w,b}$ the corresponding simulated and warped version. Clipping away very small corrections, e.g. below 10^{-4} m, results in sparse distributions localized around features like folds, wrinkles, or around the eyelids (see Figure 4). Note that $\{\mathbf{u}_c\}$ are *not* rotation invariant, which is acceptable because these details are already defined in the target configuration and do not require transformation from the neutral pose. While this assumption does not strictly hold during interactions, we found that incorporating such invariant did not offer a notable visual improvement, but it did add a significant computational cost.

The final mesh geometry that the IFA pipeline outputs is then

$$\mathbf{x}_o = \mathrm{warp}\Big(\mathbf{X}_o,\ \mathrm{sim}\big(\mathbf{X} + \sum_{b=1}^{n_b} \mathbf{u}_b w_b\big)\Big) + \sum_{b=1}^{n_b} \mathbf{u}_{c,b} w_b, \tag{14}$$

where $\mathbf{u}_b$ are the coarse blendshape displacements used to compute the coarse target expression. Equation (14) highlights our *two-scale* approach: a coarse resolution is used for simulation, while the geometric detail is preserved at a high resolution. Algorithm 1 summarizes the main forward pipeline.

Algorithm 1: Forward Simulation Pipeline

```
foreach frame $f$ do
    $\mathbf{x}^f \leftarrow \text{coarse\_rig}(\mathbf{w}^f)$    ▷ End of frame shell target
    foreach sub-step $k \in \{0, \dots, n_{\text{substeps}} - 1\}$ do
        Update colliders
        $\beta \leftarrow \frac{k}{n_{substeps}}$
        $\mathbf{x}^t \leftarrow \beta\,\mathbf{x}^f + (1-\beta)\,\mathbf{x}^{f-1}$    ▷ Interpolate from frame targets
        $\bar{\mathbf{x}}^t \leftarrow \text{cwise\_maxmin}(-\epsilon^r, \mathbf{x}^t, \epsilon^r)$    ▷ Cap actuation forces using Eq. (6)
        Assemble $\nabla\Psi(\bar{\mathbf{x}}^t)$
        $\mathbf{x} \leftarrow \mathbf{x}^n - \left(\frac{\partial^2\Psi}{\partial\mathbf{x}^2}\right)^{-1}\frac{\partial\Psi}{\partial\mathbf{x}}$    ▷ Forward and Backward Substitution
        $\mathbf{x} \leftarrow \text{constraints}(\mathbf{x})$    ▷ Apply positional constraints
        $\mathbf{v} \leftarrow \frac{\mathbf{x}-\mathbf{x}^n}{\Delta t^s}$
    $\mathbf{x}^o \leftarrow \text{warp\_and\_correct}(\mathbf{X}^o, \mathbf{x})$    ▷ Transfer deformations and restore details
```

4 Material Stiffness Reconstruction

The forward simulation pipeline described in the previous section relies on carefully chosen material stiffness distributions to reproduce target facial expressions within the reproducibility tolerance ϵ^r, while also responding realistically to contact. For reproducibility, the stiffness must be locally feasible across the entire shell and for all facial expressions. Further, a uniform stiffness fails to capture the stark difference upon contact between soft areas (e.g., cheeks, lips) and stiffer regions (e.g., nose, ears). Meanwhile, material parameters cannot simply be transferred from one character to another, since each has distinct facial geometry and deformation patterns. Using overly soft parameters (e.g. zero stiffness) trivially meets the reproducibility requirement but cannot generate convincing contact deformations. Conversely, too high stiffness will result in the passive shell not deforming enough to match the output of the rig during forward simulation.

To achieve an artist-free, scalable pipeline, suitable material parametrizations must be determined automatically and without the need for parameter fine-tuning. We do this via inverse simulation in a precomputation step, performed once per character identity and for the chosen mesh resolution. Using a differentiable simulation, we solve for non-uniform strain and bending stiffness distributions that allow, in the absence of external interactions, strict reproduction of the original rig expressions within ϵ^r in a quasistatic regime. In our blendshape-based setup, we use the set of non-corrective blendshape expressions for this stage.

As illustrated in Figure 5, the resulting stiffness fields produced in the inverse simulation step exhibit locally softer material in high-deformation zones (e.g., cheeks and lips) and stiffer material where bone or cartilage is near the skin surface (e.g., forehead, ears, nose). Our automatic reconstruction method leverages this correlation between deformation and stiffness, enabling IFA rigs to produce realistic contact responses across different regions of the face without the need for additional priors.

4.1 Inverse Simulation

We denote by $\boldsymbol{\kappa}^s$ and $\boldsymbol{\kappa}^b$ continuous scalar fields of strain and bending stiffness, respectively, discretized at the mesh vertices. Each element's stiffness is computed by averaging its vertex values. Formally, we seek:

$$(\boldsymbol{\kappa}^s, \boldsymbol{\kappa}^b) = \arg\min_{\boldsymbol{\kappa}^s, \boldsymbol{\kappa}^b} \mathcal{L}, \tag{15}$$

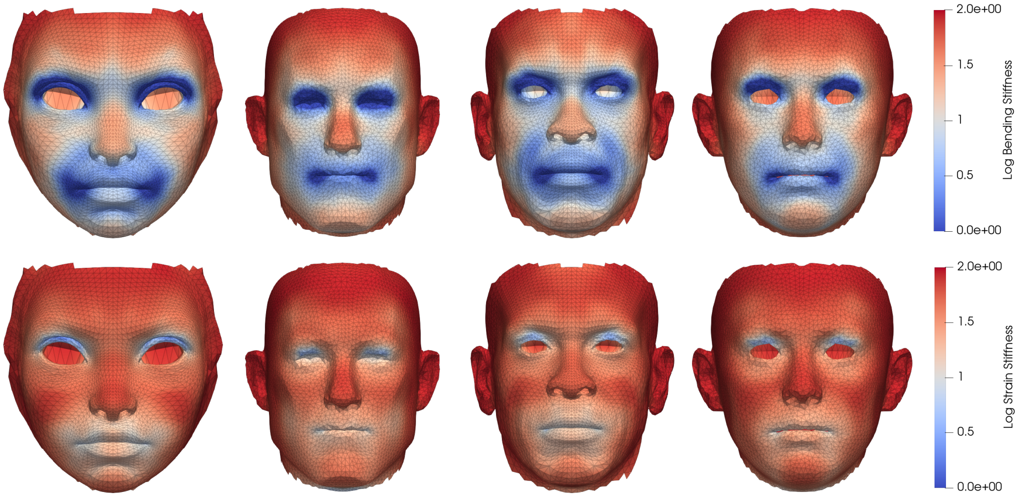

Fig. 5. Reconstructed bending (top) and strain (bottom) stiffness fields for the highest-resolution (8k triangles) simulation meshes of the four rigs.

where the global loss $\mathcal{L}$ consists of:

$$\mathcal{L} = \mathcal{L}^r + \mathcal{L}^s + \mathcal{L}^b + \mathcal{L}^{ls} + \mathcal{L}^{lb}. \tag{16}$$

Here, $\mathcal{L}^r$ is a barrier-type shape-matching loss for reproducibility, $\mathcal{L}^s$ and $\mathcal{L}^b$ are target strain and bending stiffness losses, and $\mathcal{L}^{ls}$ and $\mathcal{L}^{lb}$ are their corresponding smoothing regularizers.

Reproducibility Barrier. The term $\mathcal{L}^r$ enforces all expressions to remain within ϵ^r of their targets after simulation. We use a log barrier for such constraint defined as

$$\mathcal{L}^r = \sum_{b}^{n_f} \sum_{i}^{n_v} -\lambda^r \log\left(1 - \left(\frac{\|\mathbf{x}_{b,i} - \mathbf{x}^t_{b,i}\|_2}{\epsilon^r}\right)^2\right), \tag{17}$$

where λ^r is the barrier stiffness, $\mathbf{x}_{b,i}$ the simulated position of vertex i in expression b, and $\mathbf{x}^t_{b,i}$ its corresponding target. Violations of the reproducibility constraint incur an infinite penalty, so a valid minimizer (if it exists) must remain strictly within ϵ^r of each target.

Target Stiffness. Although zero stiffness trivially achieves reproducibility, we typically want to be as close as possible to user-defined target stiffness values that promote the desired contact responses. For strain, we write:

$$\mathcal{L}^s = \sum_{i}^{n_v} \frac{1}{2} A_i \lambda^{ts} \left\|\kappa^s_i - \kappa^{ts}\right\|_2^2, \tag{18}$$

where λ^{ts} is a scaling factor, A_i the Voronoi area of vertex i in the rest pose, κ^s_i the strain stiffness, and κ^{ts} a target stiffness. Although κ^{ts} can be tuned on a per-identity or per-scene basis to match artistic or stylistic requirements, this is not strictly necessary. In our experiments, we show that a fixed value works well across different character styles and resolutions. The corresponding edge-based smoothing term is

$$\mathcal{L}^{ls} = \sum_{i}^{n_e} \frac{1}{2} \lambda^l \left(\frac{\kappa_{i1}^s - \kappa_{i2}^s}{\|\mathbf{e}_i^0\|_2} \right)^2, \tag{19}$$

where λ^l is the smoothing parameter, κ_{i1}^s and κ_{i2}^s are the stiffness values at the edge's vertices, and $\|\mathbf{e}_i^0\|_2$ its rest length. Analogous definitions hold for bending stiffness κ^b.

4.2 Adjoint Method

Let $\boldsymbol{\theta} = (\boldsymbol{\kappa}^s, \boldsymbol{\kappa}^b)$. To solve the above optimization problem using gradient-based minimization methods we need $\frac{\mathrm{d}\mathcal{L}(\mathbf{x},\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}}$, that is, the *total* derivative of the loss with respect to the unknowns $\boldsymbol{\theta}$. To compute it, we use the adjoint method [Coros et al. 2021; McNamara et al. 2004]

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} + \boldsymbol{\lambda}^T \frac{\partial^2 \Psi}{\partial \mathbf{x}\, \partial \boldsymbol{\theta}}, \quad \text{for} \quad \left(\frac{\partial^2 \Psi}{\partial \mathbf{x}^2} \right)^T \boldsymbol{\lambda} = - \left(\frac{\partial \mathcal{L}}{\partial \mathbf{x}} \right)^T, \tag{20}$$

where $\boldsymbol{\lambda}$ is the adjoint state vector. For each candidate $\boldsymbol{\theta}$, we first solve the quasistatic simulation (11) for all target expressions, then compute $\boldsymbol{\lambda}$ and assemble the gradient. The dimensionality of the inverse problem is $2n_v$ (the nodal strain and bending stiffness values), whereas the quasistatic simulation has $3n_v n_b$ degrees of freedom (the 3D positions of all the simulation vertices for all the facial expressions).

Unlike for the forward simulation where we can use position estimations ($\hat{\mathbf{p}}$ in Eq. (8)) from the previous time step, this is not an option in quasistatics. Therefore, we solve the inverse simulation with $\hat{\mathbf{x}} = \mathbf{x}$ (no lagging). Also, since stiffness values typically span several orders of magnitude, we instead solve for $\log(\boldsymbol{\theta})$ to keep optimization variables within a manageable range and improve convergence. Besides these, the adjoint method generally introduces significant differentiation complexity, however, modern differentiation frameworks can be used to circumvent this complication [Fernández-Fernández et al. 2023; Herholz et al. 2024; Schmidt et al. 2022].

4.3 Optimization

We recommend an interior-point method due to the barrier term in Eq. (17), which is undefined outside $\|\mathbf{x} - \mathbf{x}^t\|_\infty < \epsilon^r$. Consequently, the initial guess $(\boldsymbol{\kappa}^s, \boldsymbol{\kappa}^b)$ must already be feasible (starting from $\mathbf{x} = \mathbf{x}^t$ suffices), and the optimizer must never step outside feasible space, which can be enforced with a line search. To improve efficiency, it is possible to significantly reduce the dimensionality of the quasistatic simulation by only considering regions of the facial expressions featuring non-zero deformations. Finally, adding a mild regularization to penalize excessively stiff values can prevent ill-conditioning and improve optimization stability.

4.4 Geometrical Concerns

As described in Section 3.1, we impose a maximum depth constraint to produce realistic contact responses. Without it, an object could easily be pushed into the face interior. To remain fully anatomic-free, we estimate a maximum depth for each vertex δ_i from the provided rig expressions, taking the largest normal displacement observed across all expressions. This heuristic allows realistic deeper normal displacements in fleshy areas such as the cheeks, while keeping thinner areas, such as the forehead, shallow. Ears and noses, which show minimal normal-direction deformation, are identified (e.g., via Shape Diameter Function [Shapira et al. 2008]) and assigned a fixed depth δ^{thin}. Jaw-intensive expressions are excluded from this depth computation to avoid overly softening the mandible area.

Table 2. *Vertices* and *Triangles* refer to the original input mesh. *Fit Elements* indicates the total number of elastic elements used in the inverse simulation. *IFA* represents the full forward simulation pipeline. For reference, *Orig. Rig* shows timings for the original rig run in a sparse Eigen implementation.

	Vertices	**Triangles**	**Fit Elements**	**Fit Time [s]**	**IFA [ms]**	**Orig. Rig [ms]**
Aura 2k	5230	10363	56175	44.488	0.390	0.296
Aura 4k	5230	10363	109411	152.170	0.578	0.296
Aura 6k	5230	10363	162863	400.183	0.691	0.296
Aura 8k	5230	10363	216579	1487.670	0.834	0.296
Bowen 2k	21706	43258	48166	46.334	1.048	1.188
Bowen 4k	21706	43258	91913	106.222	1.075	1.188
Bowen 6k	21706	43258	137219	231.864	1.154	1.188
Bowen 8k	21706	43258	180267	668.068	1.273	1.188
Jupiter 2k	5273	10454	48497	18.180	0.368	0.258
Jupiter 4k	5273	10454	94248	98.230	0.560	0.258
Jupiter 6k	5273	10454	137734	324.941	0.648	0.258
Jupiter 8k	5273	10454	186378	1095.340	0.764	0.258
Proteus 2k	21117	42051	50072	53.374	1.035	1.103
Proteus 4k	21117	42051	95571	150.082	1.051	1.103
Proteus 6k	21117	42051	139480	295.872	1.106	1.103
Proteus 8k	21117	42051	184760	896.287	1.245	1.103

We omit the eyelids from the shell simulation and directly prescribe their displacements since they are difficult to handle reliably and we are not interested in simulating elasticity and contact in there. Lastly, we use uniform isotropic decimation to provide a consistent contact response throughout the face, even for very coarse resolutions, and minimize visual artifacts that could arise from contact with elongated triangles.

5 Results

In this section, we present performance metrics for our method and demonstrate its applicability through representative examples. All experiments were conducted on a laptop with a 12th Gen Intel® Core™ i7-12700H CPU at 2.7 GHz (14 cores, 20 threads) and 32 GB of RAM. We utilize four blendshape rigs from the open-source release by Kavan et al. [2024], each containing around 250 blendshapes. Although these rigs include the full character bust, we extract the face region for the simulation, and normalize it to a height of 1 m. Table 2 indicates the resolution of each extracted face mesh.

5.1 Inverse Simulation: Stiffness Reconstruction

We begin by examining the inverse simulation material stiffness reconstruction in four face rigs and four simulation resolutions each. We use the 72 non-corrective blendshapes provided in each input rig as the target expressions that our pipeline should reproduce. All examples in this paper use the same reconstruction parameters, listed in Table 3, demonstrating that our method does not require fine-tuning per character, resolution, or artistic style.

The target strain and bending stiffness are chosen based on realistic ear interactions since they get assigned this target stiffness due to their lack of deformation in the expressions. The maximum actuation pressure $p^{\max}$ determines how closely the shell can approach this target stiffness while still maintaining reproducibility. Its value was established empirically by testing contact responses across the face for a sample of characters and resolutions, and selecting the one that provided the most satisfactory outcome. Note that these parameters are not tailored to any specific identity or resolution and that they work well across different rigs.

For the non-linear quasistatic shell simulation, we use the open-source STARK library [Fernández-Fernández et al. 2024] with a Project-On-Demand Newton solver [Longva et al. 2023] in conjunction with block diagonal Preconditioned Conjugate Gradient (PCG). The solver runs in parallel with double-precision floating-point arithmetic.

We employ L-BFGS [Nocedal and Wright 2006] with history size 10 for the material optimization, solving for both strain and bending stiffness simultaneously. The partial derivatives needed by the adjoint method are generated by SymX [Fernández-Fernández et al. 2023]. Figure 6 reports the convergence of our material parametrization across the four characters and four resolutions. We find L-BFGS to be well-suited as it exhibits near-quadratic convergence when approaching the solution while keeping the number of iterations relatively low, even at higher resolutions.

Here we note some measures that improve the convergence and stability of L-BFGS for our problem. We cap step sizes at 0.5 in log space to curb overly large stiffness changes before reaching the near-quadratic regime. The optimization starts with a low stiffness (10^{-4}) so that L-BFGS can accumulate curvature information cheaply. A backtracking line search employs three checks in sequence: (1) ensuring reproducibility constraints remain satisfied, (2) assessing whether the simulator requires excessive Newton or CG iterations, and (3) verifying sufficient descent [Nocedal and Wright 2006]. Should the line search fail, the L-BFGS history is reset, reverting to gradient-based steps but avoiding stagnation. Though the problem is essentially nested with three layers of optimization (L-BFGS, Newton, and PCG), these strategies yield robust and efficient performance.

Finally, Figure 7 illustrates that our stiffness reconstruction converges consistently across multiple simulation resolutions of the same rig.

Table 3. Stiffness Reconstruction Parameters.

Symbol	**Description**	**Value**
ϵ_{opt}	Optimization tolerance	10^{-3}
ϵ_{sim}	Newton tolerance	10^{-3} N
ϵ^r	Reproducibility tolerance	10^{-2} m
$p^{\max}$	Maximum actuation pressure	10^2 N m^{-2}
κ^{ts}	Target strain stiffness	10^2 N m^{-1}
κ^{tb}	Target bending stiffness	10^2 N m^{-1}
λ^r	Barrier parameter	0.002
λ^{ts}, λ^{tb}	Target stiffness scaling	1
λ^l	Smoothness regularization	5×10^{-7}

Table 4. Forward Simulation Parameters.

Symbol	**Description**	**Value**
Δt	Time step	0.01 s
Δt^s	Sub-stepping time step	0.002 s
ρ	Surface density	0.1 Kg m^{-2}
ν	Damping coefficient	0.01
δ^{thin}	Thin features depth	0.05 m
$\varepsilon^{\max}$	Max strain threshold	1.0

5.2 Reproducibility

We now compare the output of our method with the original input rig by running an example animation sequence for all characters at different resolutions. As with the stiffness reconstruction, the same simulation parameters apply to all characters and resolutions (see Table 4).

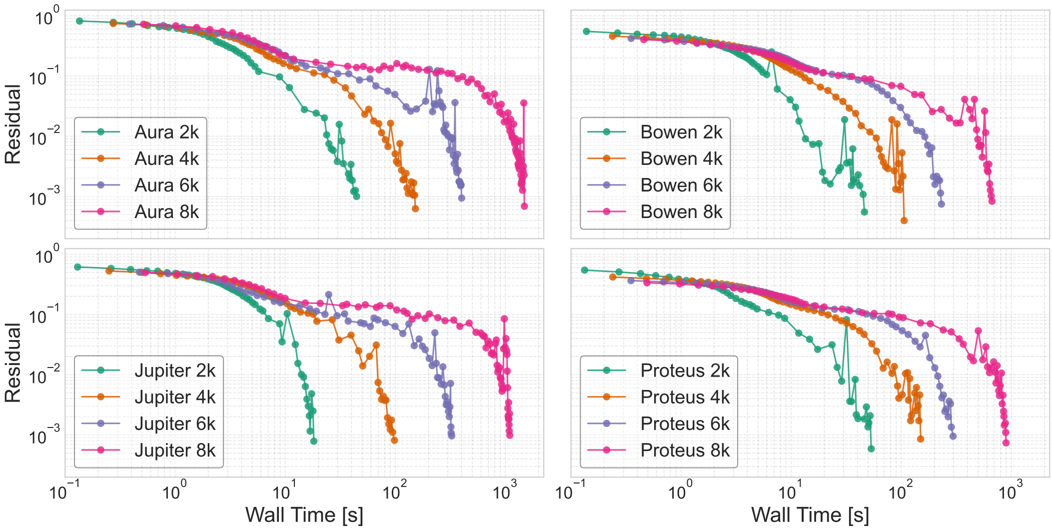

Fig. 6. Convergence of the material stiffness optimization for all characters and resolutions. On average, L-BFGS requires 48, 58, 60, and 70 iterations for 2k, 4k, 6k, and 8k resolutions, respectively.

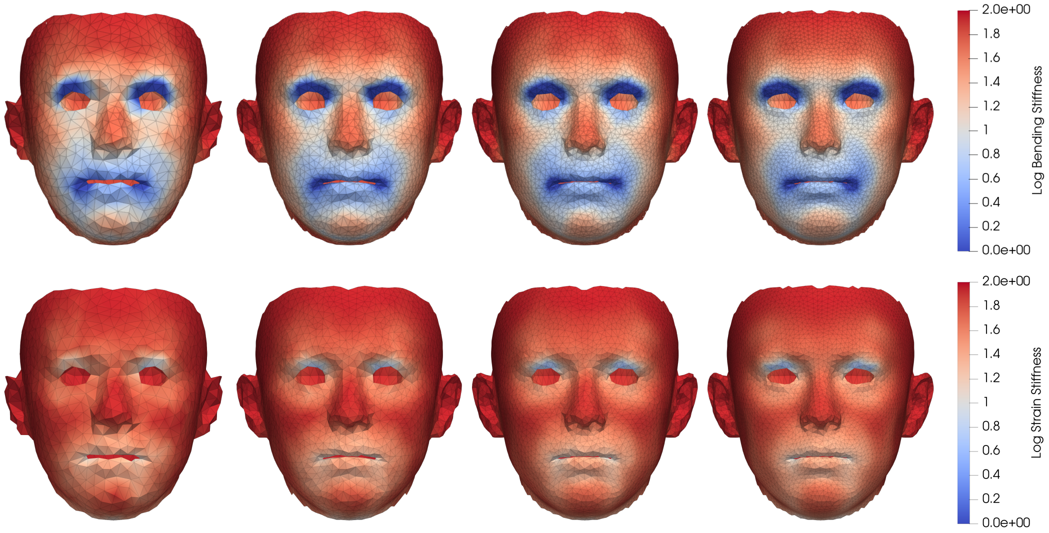

Fig. 7. Reconstructed bending (top) and strain (bottom) stiffness at four simulation resolutions for the Proteus rig (2k, 4k, 6k, and 8k triangles).

A split-screen comparison in Figure 9 shows that the IFA pipeline perceptually reproduces the original rig's resolution and expressiveness for all characters and resolutions. The key benefit is that the IFA rig performs a contact-aware *simulation*, whereas the original rig is purely *geometrical*. Since IFA effectively decouples the render and the simulation resolutions, it adapts to different computational budgets (for both inverse and forward components) without sacrificing visual fidelity.

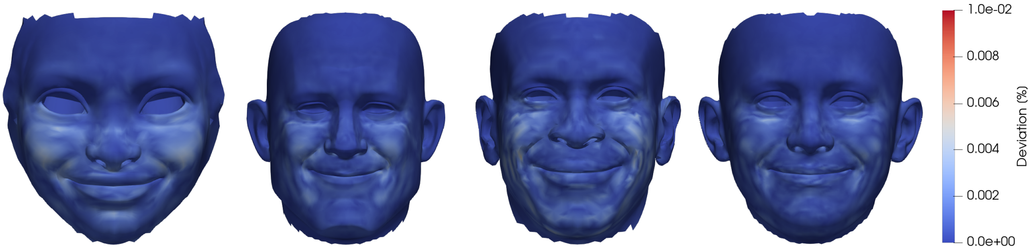

Fig. 8. Deviation between the output of the IFA pipeline and the original rig for a given frame and the exemplary characters and resolutions used previously: Aura 2k, Bowen 6k, Jupiter 4k and Proteus 8k. The corresponding maximum deviations for the entire sequence for each model are: 3.4%, 1.8%, 2.5%, 3.2%, respectively.

Figure 8 provides color-coded deviations, and the accompanying video includes full-sequence visualizations.

Notably, none of the expressions shown in Figure 9 corresponds directly to a single blendshape from the stiffness-fitting set, because all expressions in this animation are linear combinations derived by the rig. Nonetheless, our optimization strategy remains effective as long as the runtime expressions lie within the rig's expected expression space (i.e., $\|\mathbf{w}\|_\infty \leq 1$). If extrapolated expressions are expected to appear, the material optimization process should be made aware of this, for example fitting stiffness on "exaggerated" blendshapes, e.g. $1.2\mathbf{w}$, representative of the use at run time. Still, minor discrepancies can arise due to dynamic effects during fast motions, such as the rapid mouth movement illustrated in the third row of Figure 9. Reducing shell density or adjusting damping can mitigate these dynamic artifacts if desired.

5.3 Interactions

We evaluate our simulation pipeline's contact capabilities using a variety of interactions, characters, and resolutions. Figures 1 and 10 show different contact interactions performed in real-time sessions. Recorded interactive sessions with each of the characters are presented in the supplemental video.

In all examples, the corresponding blendshape rig is actively driving facial expressions. We use a sphere as the SDF-based collider, though any rigid shape is possible. The sphere moves due to a tracking force that follows the mouse and that is bounded in magnitude, ensuring safe interactions. Skin pulling is achieved by pinning simulation mesh vertices that contact the sphere, triggered by a user action. All interactions are frictionless.

These examples confirm that our method successfully incorporates realistic skin-level deformations from contact into existing facial animation rigs across various characters and simulation resolutions. Contact at the cheeks, forehead, nose, lips, and ears appear realistic, the coarse simulation resolution is hardly perceptible, and the input fidelity is not compromised.

5.4 Forward Pipeline Performance

Our implementation of the forward IFA pipeline runs single-threaded in floating-point precision. This approach accommodates broader interactive applications (e.g., telepresence or games) where the pipeline operates alongside other processes, and also leaves room for running multiple characters in parallel. We use Eigen [Guennebaud et al. 2010] for most linear algebra routines, except for the linear system solve, for which we employ an open-source optimized and vectorized version of forward/backward substitution released along with the method by Kugelstadt et al. [2021].

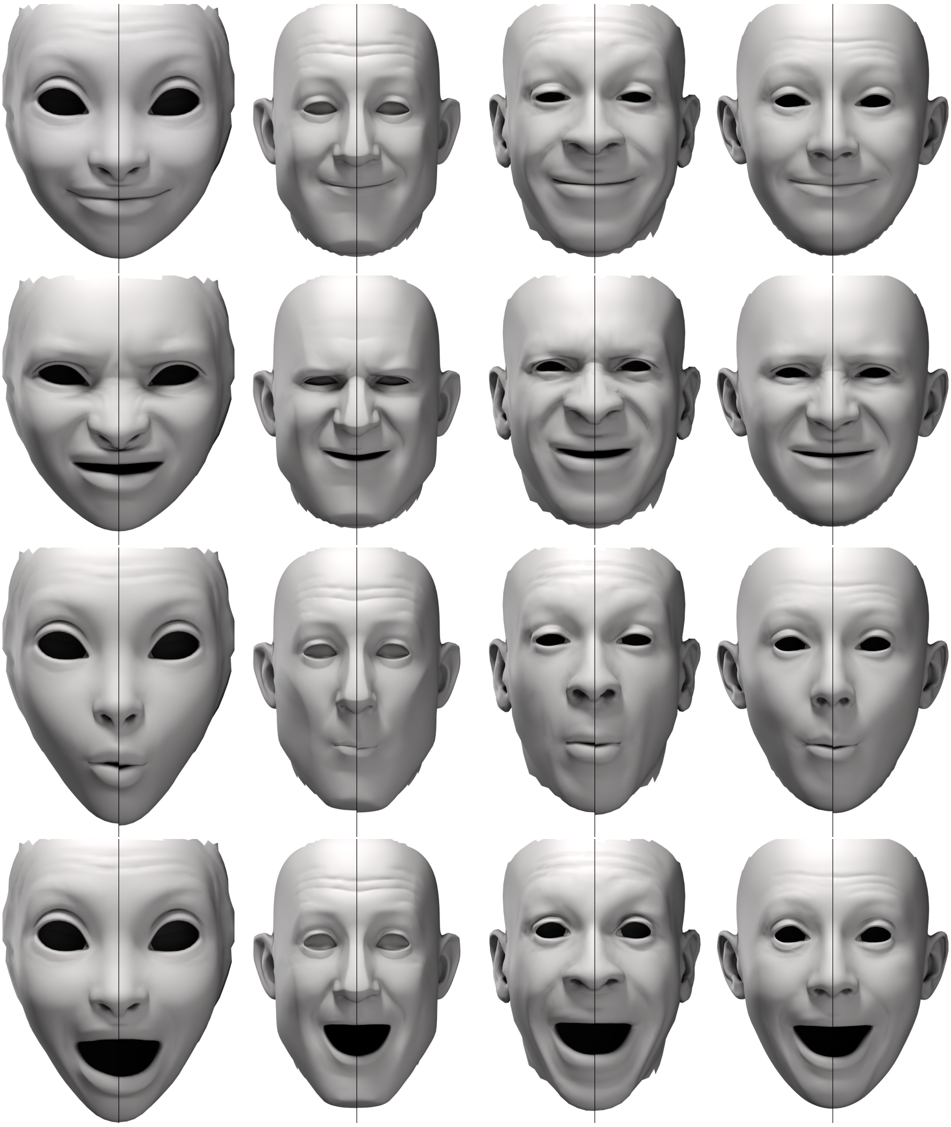

Fig. 9. Split-screen comparison between the output of the IFA simulation pipeline (left) and the original rig (right) for four different characters on the same animation sequence. One character per column: Aura 2k, Bowen 6k, Jupiter 4k and Proteus 8k. Each row corresponds to the same frame of the animation sequence.

Figure 11 presents the runtimes for four characters across four resolutions.On average, 48.2% of the simulation time is spent building the gradient, 41.7% on the linear system solve, and 10.1% on the remaining components. As expected, the runtime of the coarse rig and simulation grows roughly linearly with the shell mesh resolution. Meanwhile, the fine-detail correction stage depends on the resolution of the original input rig (see Table 2), as do the original rig timings.

All times lie between 0.39 and 1.27 milliseconds per frame update, comfortably within the runtime budget required by high refresh rate applications.

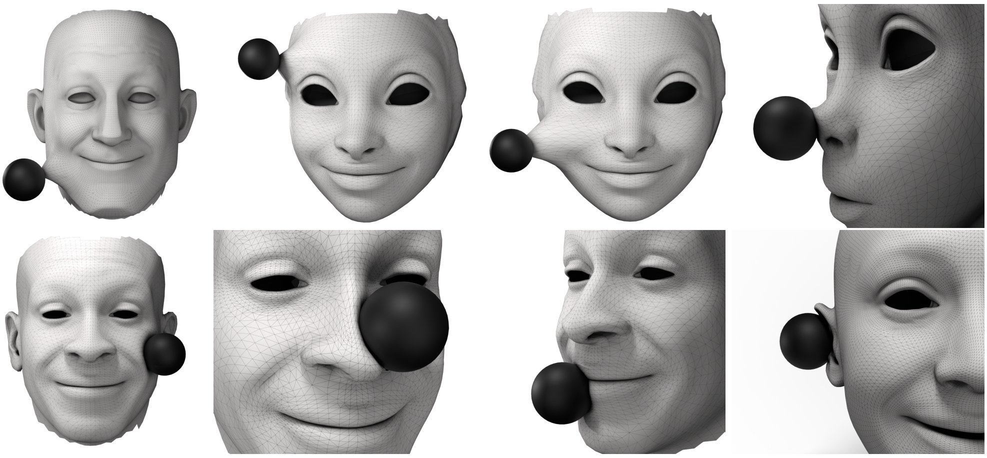

Fig. 10. A variety of contact and pull interactions performed in real time on different characters. Frames are rendered offline for improved lighting and camera settings.

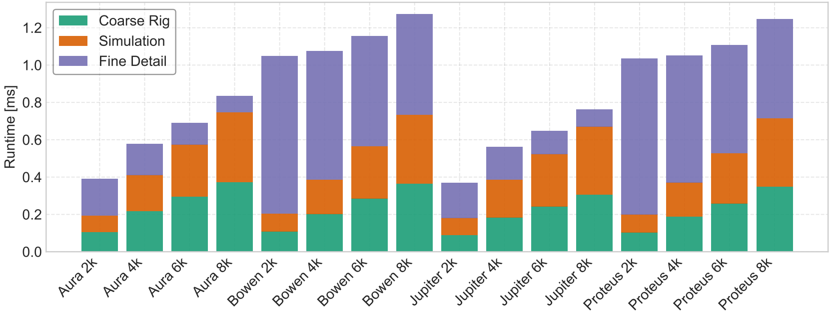

Fig. 11. Average per-frame runtimes for the forward IFA pipeline on four characters and four resolutions, shown as a stacked bar plot for coarse rig, simulation, and fine-detail correction stages.

6 Limitations and Future Work

While the IFA pipeline addresses the problem of contact in an animation setting, as presented in this work it does not *improve* the input rig's expressions (apart from adding dynamics). In any case, our method remains compatible with any approach that modifies the rig before feeding it into our contact simulation.

Because IFA uses a surface-only simulation, capturing volumetric effects (e.g., volume conservation) is challenging. In the future, we believe an approximate volume-based measure could be derived by analyzing the volume swept by the surface under contact, or by incorporating volumetric elements specifically for protruding regions such as the nose and lips.

Moreover, our method is designed to simulate local deformations caused by contact. Movements affecting the entire head (e.g., a strong push to the forehead) cannot emerge within the proposed method alone. However, such effects can be integrated if the skeletal rig is used in tandem with IFA's computed contact forces. Furthermore, our maximum depth calculation is a heuristic that performs well in practice without anatomical priors. However, more sophisticated methods (or even a skull if available) can be readily incorporated. Handling self-contact (e.g., between the lips) remains a task for future exploration.

Finally, another interesting avenue for future work is to explore mechanisms to switch between the original rig and the IFA rig at runtime. This could allow resource savings on battery-powered devices when no interaction is taking place, provided seamless transitions can hide any visual discontinuities.

7 Conclusions

We presented a novel surface-only real-time simulation pipeline that augments existing face rigs with shell mechanics and contact handling, backed by strong fidelity preservation guarantees. Our method requires only the rig itself and standard simulation parameters; no artist intervention or anatomical priors are needed. Additionally, we introduced a new material reconstruction procedure for shells, using inverse simulation, to obtain non-uniform stiffness fields that support realistic contact while ensuring reproducibility. This setup step runs in a few minutes on commodity hardware.

We have demonstrated that our simulation solution perceptually reproduces the original rig without compromising the resolution and expressiveness while delivering realistic interactions on different areas of the face, including cheeks, forehead, nose, lips and ears. Our method generalizes across different characters (including heavily stylized ones) and simulation mesh resolutions without the need for per-case parameter fine-tuning.

References

V. Barrielle and N. Stoiber. 2019. Realtime Performance-Driven Physical Simulation for Facial Animation. *Computer Graphics Forum* 38, 1 (2019), 151–166. doi:10.1111/cgf.13450 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13450.

Vincent Barrielle, Nicolas Stoiber, and Cédric Cagniart. 2016. BlendForces: A Dynamic Framework for Facial Animation. *Computer Graphics Forum* 35, 2 (2016), 341–352. doi:10.1111/cgf.12836 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12836.

Bernd Bickel, Mario Botsch, Roland Angst, Wojciech Matusik, Miguel Otaduy, Hanspeter Pfister, and Markus Gross. 2007. Multi-scale capture of facial geometry and motion. *ACM Trans. Graph.* 26, 3 (July 2007), 33–es. doi:10.1145/1276377.1276419

Bernd Bickel, Manuel Lang, Mario Botsch, Miguel A. Otaduy, and Markus Gross. 2008. Pose-space animation and transfer of facial details. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '08)*. Eurographics Association, Goslar, DEU, 57–66.

Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective dynamics: fusing constraint projections for fast simulation. *ACM Trans. Graph.* 33, 4 (July 2014), 154:1–154:11. doi:10.1145/2601097.2601116

Prashanth Chandran and Gaspard Zoss. 2023. Anatomically Constrained Implicit Face Models. doi:10.48550/arXiv.2312.07538 arXiv:2312.07538.

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. 2022. Animatomy: an Animator-centric, Anatomically Inspired System for 3D Facial Modeling, Animation and Transfer. In *SIGGRAPH Asia 2022 Conference Papers (SA '22)*. Association for Computing Machinery, New York, NY, USA, 1–9. doi:10.1145/3550469.3555398

Matthew Cong, Kiran S. Bhat, and Ronald Fedkiw. 2016. Art-directed muscle simulation for high-end facial animation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '16)*. Eurographics Association, Goslar, DEU, 119–127.

Stelian Coros, Miles Macklin, Bernhard Thomaszewski, and Nils Thürey. 2021. Differentiable simulation. In *SIGGRAPH Asia 2021 Courses* (Tokyo, Japan) *(SA '21)*. Association for Computing Machinery, New York, NY, USA, Article 3, 142 pages. doi:10.1145/3476117.3483433

José Antonio Fernández-Fernández, Fabian Löschner, Lukas Westhofen, Andreas Longva, and Jan Bender. 2023. Symx: Energy-based simulation from symbolic expressions. *arXiv preprint arXiv:2303.02156* (2023).

José Antonio Fernández-Fernández, Ralph Lange, Stefan Laible, Kai O. Arras, and Jan Bender. 2024. STARK: A Unified Framework for Strongly Coupled Simulation of Rigid and Deformable Bodies with Frictional Contact. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*.

Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '03)*. Eurographics Association, Goslar, DEU, 62–67.

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

Philipp Herholz, Tuur Stuyck, and Ladislav Kavan. 2024. A Mesh-based Simulation Framework using Automatic Code Generation. *ACM Trans. Graph.* 43, 6, Article 215 (Nov. 2024), 17 pages. doi:10.1145/3687986

Alexandru-Eugen Ichim, Petr Kadleček, Ladislav Kavan, and Mark Pauly. 2017. Phace: physics-based face modeling and animation. *ACM Trans. Graph.* 36, 4 (July 2017), 153:1–153:14. doi:10.1145/3072959.3073664

Alexandru-Eugen Ichim, Ladislav Kavan, Merlin Nimier-David, and Mark Pauly. 2016. Building and animating user-specific volumetric face rigs. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '16)*. Eurographics Association, Goslar, DEU, 107–117.

Petr Kadleček and Ladislav Kavan. 2019. Building Accurate Physics-based Face Models from Data. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2 (July 2019), 15:1–15:16. doi:10.1145/3340256

Ladislav Kavan, John Doublestein, Martin Prazak, Matthew Cioffi, and Doug Roble. 2024. Compressed Skinning for Facial Blendshapes. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) *(SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, Article 47, 9 pages. doi:10.1145/3641519.3657477

Jungmin Kim, Min Gyu Choi, and Young J. Kim. 2020. Real-time Muscle-based Facial Animation using Shell Elements and Force Decomposition. In *Symposium on Interactive 3D Graphics and Games (I3D '20)*. Association for Computing Machinery, New York, NY, USA, 1–9. doi:10.1145/3384382.3384531

Dan Koschier, Crispin Deul, Magnus Brand, and Jan Bender. 2017. An <inline-formula><tex-math notation="LaTeX">*hp*</tex-math><alternatives><inline-graphic xlink:href="koschier-ieq1-2730202.gif"/></alternatives></inline-formula>-Adaptive Discretization Algorithm for Signed Distance Field Generation. *IEEE Transactions on Visualization and Computer Graphics* 23, 10 (Oct. 2017), 2208–2221. doi:10.1109/TVCG.2017.2730202

Yeara Kozlov, Derek Bradley, Moritz Bächer, Bernhard Thomaszewski, Thabo Beeler, and Markus Gross. 2017. Enriching Facial Blendshape Rigs with Physical Simulation. *Computer Graphics Forum* 36, 2 (May 2017), 75–84. doi:10.1111/cgf.13108

Yeara Kozlov, Hongyi Xu, Moritz Bächer, Derek Bradley, Markus Gross, and Thabo Beeler. 2019. Data-Driven Physical Face Inversion. doi:10.48550/arXiv.1907.10402 arXiv:1907.10402 [cs].

Tassilo Kugelstadt, Jan Bender, José Antonio Fernández-Fernández, Stefan Rhys Jeske, Fabian Löschner, and Andreas Longva. 2021. Fast Corotated Elastic SPH Solids with Implicit Zero-Energy Mode Control. *Proc. ACM Comput. Graph. Interact. Tech.* 4, 3, Article 33 (Sept. 2021), 21 pages. doi:10.1145/3480142

T. Kugelstadt, D. Koschier, and J. Bender. 2018. Fast Corotated FEM using Operator Splitting. *Computer Graphics Forum* 37, 8 (2018), 149–160. doi:10.1111/cgf.13520 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13520.

Yuencheng Lee, Demetri Terzopoulos, and Keith Waters. 1995. Realistic modeling for facial animation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (SIGGRAPH '95)*. Association for Computing Machinery, New York, NY, USA, 55–62. doi:10.1145/218380.218407

J. P. Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Fred Pighin, and Zhigang Deng. 2014. Practice and Theory of Blendshape Facial Models. (2014). https://doi.org/10.2312/egst.20141042 Publisher: The Eurographics Association.

Junxuan Li, Chen Cao, Gabriel Schwartz, Rawal Khirodkar, Christian Richardt, Tomas Simon, Yaser Sheikh, and Shunsuke Saito. 2024. URAvatar: Universal Relightable Gaussian Codec Avatars. In *SIGGRAPH Asia 2024 Conference Papers* (Tokyo, Japan) *(SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 128, 11 pages. doi:10.1145/3680528.3687653

Tiantian Liu, Adam W. Bargteil, James F. O'Brien, and Ladislav Kavan. 2013. Fast simulation of mass-spring systems. *ACM Trans. Graph.* 32, 6 (Nov. 2013), 214:1–214:7. doi:10.1145/2508363.2508406

Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. 2018. Deep appearance models for face rendering. *ACM Trans. Graph.* 37, 4, Article 68 (July 2018), 13 pages. doi:10.1145/3197517.3201401

Andreas Longva, Fabian Löschner, José Antonio Fernández-Fernández, Egor Larionov, Uri M Ascher, and Jan Bender. 2023. Pitfalls of Projection: A study of Newton-type solvers for incremental potentials. *arXiv preprint arXiv:2311.14526* (2023).

Shugao Ma, Tomas Simon, Jason Saragih, Dawei Wang, Yuecheng Li, Fernando De La Torre, and Yaser Sheikh. 2021. Pixel codec avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 64–73.

Miles Macklin, Kier Storey, Michelle Lu, Pierre Terdiman, Nuttapong Chentanez, Stefan Jeschke, and Matthias Müller. 2019. Small steps in physics simulation. In *Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '19)*. Association for Computing Machinery, New York, NY, USA, 1–7. doi:10.1145/3309486.3340247

Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. 2011. Example-based elastic materials. *ACM Trans. Graph.* 30, 4, Article 72 (July 2011), 8 pages. doi:10.1145/2010324.1964967

Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. 2004. Fluid control using the adjoint method. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 449–456. doi:10.1145/1015706.1015744

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118.

Rahul Narain, Matthew Overby, and George E. Brown. 2016. ADMM $\supseteq$ projective dynamics: fast simulation of general constitutive models. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '16)*. Eurographics Association, Goslar, DEU, 21–28.

Jorge Nocedal and Stephen J Wright. 2006. *Numerical optimization*. Springer.

Hyojoon Park, Sangeetha Grama Srinivasan, Matthew Cong, Doyub Kim, Byungsoo Kim, Jonathan Swartz, Ken Museth, and Eftychios Sifakis. 2024. Near-realtime Facial Animation by Deep 3D Simulation Super-Resolution. *ACM Trans. Graph.* 43, 5 (Aug. 2024), 158:1–158:20. doi:10.1145/3670687

Frederick I. Parke. 1972. Computer generated animation of faces. In *Proceedings of the ACM annual conference - Volume 1 (ACM '72, Vol. 1)*. Association for Computing Machinery, New York, NY, USA, 451–457. doi:10.1145/800193.569955

Stephen M. Platt and Norman I. Badler. 1981. Animating facial expressions. *SIGGRAPH Comput. Graph.* 15, 3 (Aug. 1981), 245–252. doi:10.1145/965161.806812

M. Romeo and S. C. Schvartzman. 2020. Data-Driven Facial Simulation. *Computer Graphics Forum* 39, 6 (2020), 513–526. doi:10.1111/cgf.14089 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14089.

Patrick Schmidt, Janis Born, David Bommes, Marcel Campen, and Leif Kobbelt. 2022. TinyAD: Automatic differentiation in geometry processing made simple. In *Computer graphics forum*, Vol. 41. Wiley Online Library, 113–124.

Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer* 24 (2008), 249–259.

Soshi Shimada, Vladislav Golyanik, Patrick Pérez, and Christian Theobalt. 2023. Decaf: Monocular Deformation Capture for Face and Hand Interactions. https://arxiv.org/abs/2309.16670v2

Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph.* 24, 3 (July 2005), 417–425. doi:10.1145/1073204.1073208

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* 40, 4 (Aug. 2021), 1–14. doi:10.1145/3450626.3459883

Pascal Volino and Nadia Magnenat-Thalmann. 2006. Simple linear bending stiffness in particle systems. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '06)*. Eurographics Association, Goslar, DEU, 101–105.

Nicolas Wagner, Mario Botsch, and Ulrich Schwanecke. 2023. SoftDECA: Computationally Efficient Physics-Based Facial Animations. In *Proceedings of the 16th ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG '23)*. Association for Computing Machinery, New York, NY, USA, 1–11. doi:10.1145/3623264.3624439

Nicolas Wagner, Mario Botsch, and Ulrich Schwanecke. 2024. NePHIM: A Neural Physics-Based Head-Hand Interaction Model. doi:10.48550/arXiv.2410.13503 arXiv:2410.13503.

Nicolas Wagner, Ulrich Schwanecke, and Mario Botsch. 2022. Neural Volumetric Blendshapes: Computationally Efficient Physics-Based Facial Blendshapes. https://arxiv.org/abs/2212.14784v2

Keith Waters. 1987. A muscle model for animation three-dimensional facial expression. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques (SIGGRAPH '87)*. Association for Computing Machinery, New York, NY, USA, 17–24. doi:10.1145/37401.37405

Qingxuan Wu, Zhiyang Dou, Sirui Xu, Soshi Shimada, Chen Wang, Zhengming Yu, Yuan Liu, Cheng Lin, Zeyu Cao, Taku Komura, Vladislav Golyanik, Christian Theobalt, Wenping Wang, and Lingjie Liu. 2024. DICE: End-to-end Deformation Capture of Hand-Face Interactions from a Single Image. doi:10.48550/arXiv.2406.17988 arXiv:2406.17988 version: 1.

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* 41, 4 (July 2022), 1–10. doi:10.1145/3528223.3530156

Lingchen Yang, Gaspard Zoss, Prashanth Chandran, Paulo Gotardo, Markus Gross, Barbara Solenthaler, Eftychios Sifakis, and Derek Bradley. 2023. An Implicit Physical Face Model Driven by Expression and Style. In *SIGGRAPH Asia 2023 Conference Papers (SA '23)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3610548.3618156

Lingchen Yang, Gaspard Zoss, Prashanth Chandran, Markus Gross, Barbara Solenthaler, Eftychios Sifakis, and Derek Bradley. 2024. Learning a Generalized Physical Face Model From Data. *ACM Trans. Graph.* 43, 4 (July 2024), 94:1–94:14. doi:10.1145/3658189

ChangAn Zhu and Chris Joslin. 2024. Fabrig: A Cloth-Simulated Transferable 3D Face Parameterization. In *SIGGRAPH Asia 2024 Conference Papers (SA '24)*. Association for Computing Machinery, New York, NY, USA, 1–10. doi:10.1145/3680528.3687695