Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2016)
Ladislav Kavan and Chris Wojtan (Editors)

Building and Animating User-Specific Volumetric Face Rigs

Alexandru-Eugen Ichim[†1], Ladislav Kavan[‡2], Merlin Nimier-David[§1] and Mark Pauly[¶1]

[1]EPFL
[2]University of Utah

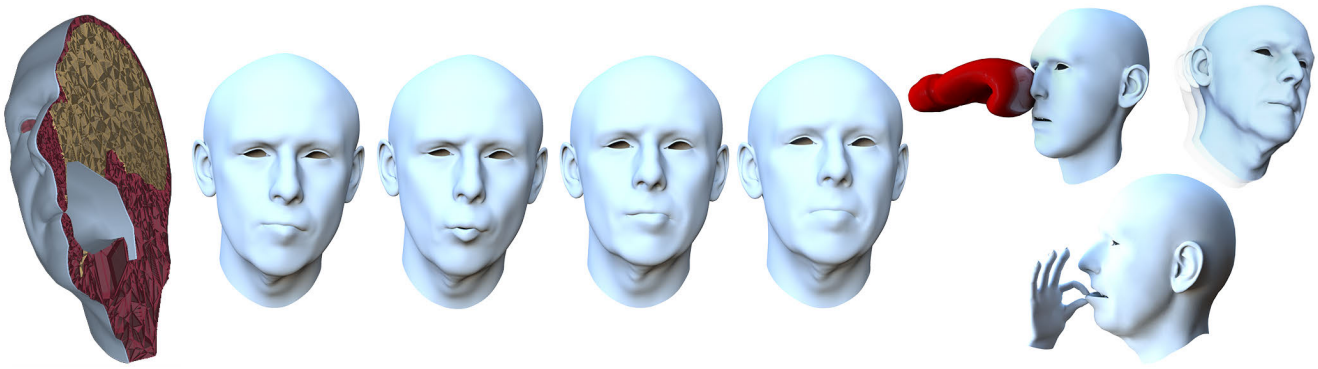


Figure 1: We present a facial animation system that can simulate physics-based volumetric effects such as self-collisions and collision with external objects. Our method is data driven and avoids the burden of detailed anatomical modeling.

Abstract

Currently, the two main approaches to realistic facial animation are 1) blendshape models and 2) physics-based simulation. Blendshapes are fast and directly controllable, but it is not easy to incorporate features such as dynamics, collision resolution, or incompressibility of the flesh. Physics-based methods can deliver these effects automatically, but modeling of muscles, bones, and other anatomical features of the face is difficult, and direct control over the resulting shape is lost. We propose a method that combines the benefits of blendshapes with the advantages of physics-based simulation. We acquire 3D scans of a given actor with various facial expressions and compute a set of volumetric blendshapes *that are compatible with physics-based simulation, while accurately matching the input scans. Furthermore, our volumetric blendshapes are driven by the same weights as traditional blendshapes, which many users are familiar with. Our final facial rig is capable of delivering physics-based effects such as dynamics and secondary motion, collision response, and volume preservation without the burden of detailed anatomical modeling.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

1. Introduction

Realistic animation of human faces is a long standing problem in computer graphics. Blendshape models are currently the most widely used solution in animation production [LAR*14] and impressive facial animations have been created with blendshape models in recent high-end productions. However, this process can be very labor-intensive and time-consuming even for experienced digital artists. Physics-based simulation of anatomically-based face models can potentially eliminate much of this manual work, because non-linear effects such as incompressibility of biological soft tissues or prevention of self-collisions (e.g. lips-lips or lips-teeth) can be handled automatically. However, the anatomy of the human face is

† alexandru.ichim@epfl.ch
‡ ladislav.kavan@gmail.com
§ merlin.nimier-david@epfl.ch
¶ mark.pauly@epfl.ch

DOI: 10.2312/sca.20161228

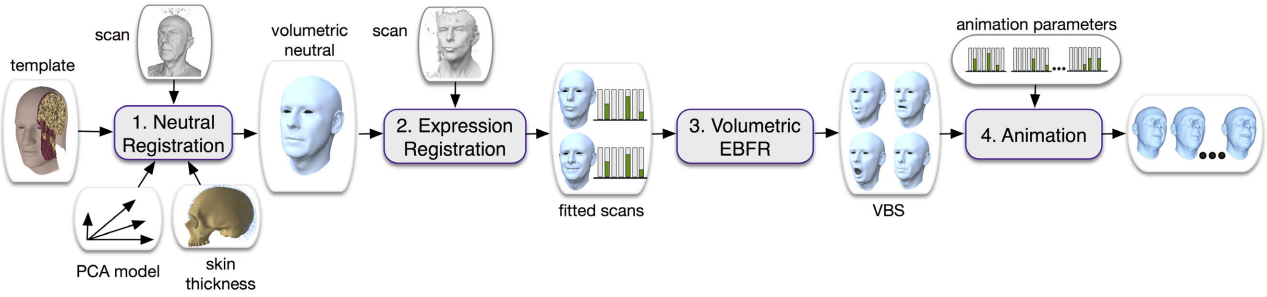


Figure 2: Workflow of our method: from a template model and input 3D scans, our system produces a subject-specific facial animation model. We propose a volumetric formulation of example-based facial rigging (EBFR) to generate the volumetric blendshapes (VBS).

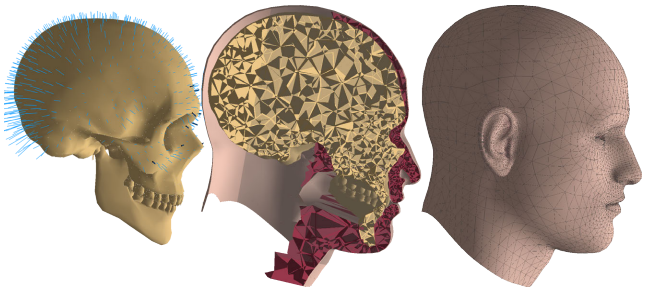


Figure 3: Template model: skull of an average subject with expected flesh thicknesses (left), tet-mesh of the interior (middle), and skin (right).

highly complex, posing significant difficulties in creating accurate anatomical face models of specific people.

Instead, we explore a new route, proposing a facial animation model that leverages the benefits of physics-based simulation without the burden and complexity of full anatomical modeling. Specifically, our technique helps prevent geometric inconsistencies such as volume loss, inter-penetrations, or unnatural facial expressions commonly observed in traditional blendshape models. Even though these deficiencies can be manually fixed by a skilled artist using corrective blendshapes, our method achieves physically-realistic behavior automatically, without the need of user intervention.

Our goal is to build an animatable facial rig of a specific actor. We start by acquiring 3D scans of several facial expressions of the actor including a neutral face shape. These scans are used to adapt a volumetric head template, corresponding to an average human (see Figure 3), to the specific actor. To achieve physics-based behavior, we propose a novel *volumetric blendshape* model, which controls the deformation gradients in the entire face volume.

The proposed volumetric blendshapes model retains the key desirable properties of traditional blendshapes: posing with intuitive blendshape weights and direct control over the resulting deformations. This means that any animator familiar with traditional blendshape models will be able to readily use our method. In contrast to traditional blendshapes, our model performs a full physics-based simulation, allowing even effects such as inertia or collisions with external objects. This is enabled by the fact that our volumetric blendshapes control deformation gradients of the flesh instead of absolute positions. However, we do not model individual muscles, which would require significant modeling effort and simulation time. Instead, the volumetric blendshapes discretize the entire deformable volume of the face using a tetrahedral mesh.

Our method (see Figure 2) assumes an average-human volumetric head model as input. To create an actor-specific face model, we scan the actor in a neutral pose and several (in the order of 10) facial expressions. In the first step, *Neutral Registration* in Figure 2, we volumetrically warp the template to align with the input scan of the actor's neutral facial expression. In step 2, we perform *Expression Registration* to deform this neutral shape into the acquired facial expressions, such as smile, frown, etc. The key difference from the first step is that now we assume the bones are rigid and the soft tissues are incompressible, because at this stage we do not model a new human being, but rather explain different facial expressions of the same actor. Due to the fact that our models are volumetric, we obtain full volumetric deformation for each of the facial expressions.

In order to create a facial rig compatible with traditional blendshape models, step 3: *Volumetric EBFR* executes a volumetric version of Example-Based Facial Rigging [LWP10], i.e., explaining each of the expression scans using a blend of volumetric blendshapes. The key idea of volumetric blendshapes is to perform nonlinear blending of deformation gradients of all tetrahedra in our face model. On one hand, volumetric blendshapes are driven by the same weights as traditional blendshapes, constituting a convenient interface for the *Animation* stage of our pipeline. On the other hand, volumetric blendshapes approximate muscle contraction forces, i.e., the generators of facial expressions. This allows us to combine them with other competing forces in a physics-based simulation, enabling us to deliver effects such as secondary motion and inertia, volume preservation, and contact forces.

Contributions. We present a pipeline to turn 3D scans of an actor's face into physics-based simulation-ready models that are able to respond to inertia or external forces, e.g., due to self-collisions of the face or collisions with external objects. We formulate our pipeline in a coherent optimization framework – all components are built using the concepts of Projective Dynamics [BML*14], which 1) results in efficient run times and 2) can be easily reproduced using open source implementations of Projective Dynamics such as ShapeOp [DDB*15]. Several novel technical contributions make

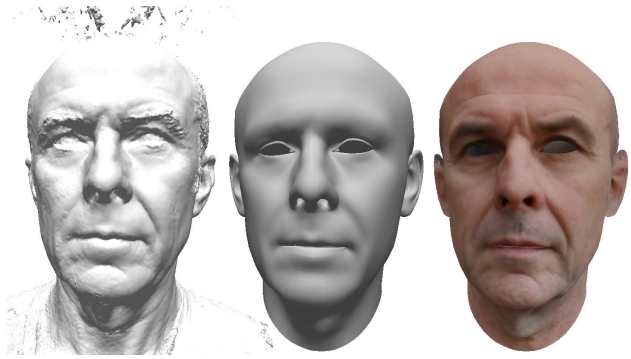


Figure 4: Input hi-res 3D scan (left). Our volumetric physics-based model (middle) uses only a medium-resolution mesh, but details can be re-introduced using high-resolution textures (right), as is commonly done in high-end productions.

this approach practically viable: 1) novel registration methods using physics-based priors such as volume preservation and self-collision handling, 2) advanced collision handling for Projective Dynamics, and 3) a "baking" system for generating higher-order corrective blendshapes which explain physical effects such as volume preservation and collisions with performance comparable to traditional blendshapes.

In this paper we focus on creating simulation-ready volumetric models. We do not aim for complete production-quality facial rigs that are commonly equipped with high resolution textures, normal, or displacement maps, see Figure 4. Compared to traditional blendshape models, our approach provides more accurate volume and area preservation, as well as rigid motion of the skull and the jaw. Our model also handles interactions between the lips and the teeth, often prone to self-intersections with traditional blendshape models, in particular for speech or chewing sequences. We can also simulate interactions with external objects, e.g., responding to contacts with rigid bodies.

2. Related Work

Facial reconstruction. Research in the field of facial animation has mostly focused on data-driven techniques, due to the high complexity of facial morphology. The seminal work of [BV99] builds a statistical (PCA) model of facial geometry and later on [CWZ*14] builds a bilinear facial model, which can be employed to create blendshape models from a single image [BV99], [CWZ*14], from multiview stereo [ABF*07], [ARL*10], or for the creation of personalized real-time tracking profiles from RGB-D data [WBLP11], [BWP13] or monocular video [IBP15], [GVWT13], [SWTC14].

Anatomical models. Dicko et al. [DLG*13] propose a method for transferring and editing the internal structure of human bodies. They use a template human body model containing the skeleton and internal organs and register it to new surface-mesh humanoid models. The exterior surfaces are registered and the internal volume is adapted using harmonic deformation. Additional constraints are used for manually tuning the amount of fat tissue and keeping the bones straight. In a similar vein, [ZHK15] adapts the bone structure of upper and lower limbs given an RGB-D sequence of moving limbs. [CBB*15] propose a technique to transfer facial anatomy to challenging non-human creatures using sophisticated correspondences between the template and target shapes. However, their method relies only on a single neutral facial expression. In contrast, our approach uses multiple scans of facial expressions and is able to reproduce them with high accuracy.

[VCL*06] present a review of computerized techniques for craniofacial reconstruction, i.e., generating the skin surface of faces from 3D skull information. An algorithm to reconstruct the skin surface, as well as an animatable muscle system from 3D scans of skulls is proposed by [KHS03]. Their method registers a template face model to the 3D mesh of the skull by RBF deformation on a sparse set of landmarks with user-specified skin thicknesses. A mass-spring system is then adapted to the fitted template and the face can be animated. For more application-specific use cases of anatomical models, [BB14] present an approach for rigid stabilization of the head in high quality 3D scans by fitting a simple skull model with physically-inspired constraints. [BBK*15] use high quality facial scanning and a simplified physical model in order to recover spatio-temporal details of the eyelids.

Physics-based facial animation. [SNF05] build a system for physics-based animation of one human subject. The subject's face is captured using a laser scanner (high-resolution, surface only) and an MRI scanner (low-resolution, volumetric). A simulation-ready 3D model is created using custom software tools, medical atlases, and multiple months of manual work. The resulting face model is biomechanically accurate in the sense that realistic facial expressions are created by physics-based simulation of muscle activations. In addition, the model can be used to track a facial performance of the subject, captured using a sparse set of markers attached to the face. The physics simulator is based on a quasi-static FEM approach, numerically solved using Newton's method.

More recent techniques such as Position-based [MHHR07] and Projective Dynamics [LBOK13, BML*14] propose to substitute Newton's method with faster numerical solution procedures. In particular, Projective Dynamics [BML*14] yields faster per-iteration times while simultaneously enjoying high robustness and support of many different types of deformation constraints.

Combining simulation and data. Our volumetric blendshapes blend deformation gradients, similarly to MeshIK [SZGP05]. However, MeshIK relies only on deformation gradients of surface triangles and does not support dynamics or collisions. Similar approaches such as deformation transfer [SP04] and FaceShift [WBLP11] also do not take collisions into account, see Figure 6. We use a complete volumetric model combined with full physics-based simulation, enabling us to deliver inertial and secondary motion effects (such as flesh jiggling) as well as realistic response to collisions while preserving the volume of biological soft tissues. [MWF*12] build a mass-spring system model for the face that is able to deliver some of these effects. However, volume preservation with mass-spring systems is problematic. A concurrent work [BSC16] uses Projective Dynamics to deform the surface of a face combined with a new concept of "blendforces", which are similar to our volumetric blendshapes. However, [BSC16] model only the surface of the face. In

contrast, our method explicitly models volume preservation of the flesh, as well as rigidity of the skull and the jaw bones.

3. Method

As input, we assume a *template* model of an average human face. This model consists of a volumetric tetrahedral mesh for the neutral expression which discretizes the interior of the head, including a realistic model of the oral cavity, see Figure 3. We obtain this model by converting a commercial anatomical CAD model of the head [Zyg16] into a tet-mesh using the method of [JKSH13]. The *skin* is the boundary of this tet-mesh. To get an initial model of facial deformations, we use an artist-created surface blendshape model [WBLP11], which also comes with parameterization (UV coordinates). We register this model against the boundary of our volumetric model, which allows us to animate the skin, but not the interior. Extending the surface deformations to the interior is one aspect of our pipeline, discussed below.

Our final volumetric template model is a single connected tet-mesh where we can identify the following components corresponding to high-level anatomical features of the head (see Figure 3): 1) skin – a UV-mapped surface mesh, 2) bones – tet-meshes for the cranium and the mandible, including teeth, 3) flesh – in-between tet-mesh conforming to the skin and the boundaries of the bones.

Our volumetric model corresponds to a hypothetical average human subject and must be adapted to a given actor. The scanning of our actor's face is performed using a custom multiview stereo rig with 12 DSLR cameras with uniform lighting, similar to [BBB*10]. Note that our method is not dependent on the specific scanning method. Any approach for creating high-resolution scans of a face, e.g. laser scanning, RGB-D, are equally suitable. The captured photos are processed in AgiSoft PhotoScan which creates detailed triangle meshes for each expression.

3.1. Volumetric modeling of actor's neutral face

Registration. The 3D scan of the actor's neutral face is a triangle mesh containing noise, topological errors, and other imperfections, see Figure 7. We overcome these issues by regularized registration, i.e., by deforming our volumetric template model to align well with the 3D scan of the actor. We follow the paradigm of Iterative Closest Point (ICP) algorithms and iterate between finding correspondences and volumetric deformations of our template. We find surface correspondences using the standard approach of closest points with distance and normal-based rejection [RL01]. The non-rigid deformation steps are alternated with shape-preserving rigid fitting steps, which only allow for translation, rotation, and uniform scale (necessary because multi-view stereo does not determine scale).

Deformation model. We model volumetric deformations in the Projective Dynamics framework due to its speed, robustness, and flexibility [BML*14]. The key concept of Projective Dynamics is to use elastic energy potentials expressed in the following "projective" form:

$$E_i(\mathbf{x}) = \|\mathbf{G}_i\mathbf{x} - P_i(\mathbf{G}_i\mathbf{x})\|_F^2, \quad (1)$$

where E_i is the energy contribution due to element number i (e.g., tetrahedron), $\mathbf{x}$ is a column vector concatenating all of the nodal coordinates (deformed state), $\mathbf{G}_i$ is a sparse matrix, typically representing a discrete differential operator, and P_i is a projection operator. For example, the finite element As-Rigid-As-Possible model (E_i^{ARAP}) [CPSS10] can be expressed with $\mathbf{G}_i$ representing the deformation gradient of a tetrahedron [SB12] and P_i representing the projection onto $SO(3)$, i.e., the group of 3D rotations.

Correspondence terms. Our registration process utilizes a set of 26 landmark correspondences initialized automatically using [SLC11] and fine-tuned by the user (see the figure on the right). In the Projective Dynamics framework, these correspondences are implemented using an "attachment" term E_i^{attach} where $\mathbf{G}_i$ is simply a selector matrix and P_i is the constant target position (i.e., projection onto a fixed point). The correspondences found through closest point search by the ICP algorithm are handled similarly; the only difference is that we do not "trust" the absolute positions of these correspondences and therefore use a point-to-plane energy term $E_i^{\text{planeDist}}$, where $\mathbf{G}_i$ is still a selector, but P_i projects on the plane tangent to the scan at the closest point. This allows for tangential sliding, which improves the convergence of the ICP process [LSP08]. The point-to-plane energy is also used as a collision response mechanism, projecting inter-penetrated vertices outside of the volume; we elaborate on collision processing in Section 3.5.

Face priors. We also add energy terms specific to faces, i.e., utilizing the prior knowledge that the resulting surface must correspond to a plausible human face. As we are solving for deformations of the interior too, ideally we would also use a statistical shape model of skulls. However, so far we were not successful in obtaining a sufficiently large database of 3D skull shapes. Instead, we utilize flesh thickness measurements from a forensic study [DGCV*06], inspired by the work of Beeler and Bradley [BB14] on rigid stabilization.

Statistical shape models of neutral faces of various people are available; we use the established PCA model of Blanz and Vetter [BV99]. This model consists of a mean face shape $\mathbf{m}$ and 50 PCA basis vectors, represented as orthonormal columns of a matrix $\mathbf{B}$. Each of the basis vectors is associated with a standard deviation, represented as a 50×50 diagonal matrix Σ. Let us also denote by $\mathbf{S}$ a surface selector matrix, i.e., $\mathbf{Sx}$ represents the boundary (skin) vertices, discarding the interior ones. The skin shape $\mathbf{Sx}$ can be additively decomposed into two parts: one in the column space of $\mathbf{B}$ and the other one orthogonal to it. We introduce a different energy term for each part. For the component of $\mathbf{Sx}$ in the column space of $\mathbf{B}$ we can measure its likelihood of corresponding to a natural face shape, as predicted by our PCA model. This leads to $E^{\text{faceLike}}(\mathbf{x}) = \|\Sigma^{-1/2}\mathbf{B}^{\mathsf{T}}(\mathbf{Sx} - \mathbf{m})\|^2$. The orthogonal complement $(\mathbf{I} - \mathbf{BB}^{\mathsf{T}})(\mathbf{Sx} - \mathbf{m})$ corresponds to modes outside of our PCA model. We do not have standard deviations for these modes and therefore we penalize them uniformly using the term $E^{\text{faceDist}}(\mathbf{x}) = \|(\mathbf{I} - \mathbf{BB}^{\mathsf{T}})(\mathbf{Sx} - \mathbf{m})\|^2$. Both of these terms are convex quadratic functions that can be easily embedded in the Projective Dynamics framework.

Flesh thickness. Our flesh thickness model is based on statistical information from a forensic study [DGCV*06]. We start from a sparse set of 16 skull landmarks containing the mean and variance of flesh thickness at this point, and then linearly interpolate these values over the entire skull. Specifically, for each non-landmark skull vertex, we find three closest landmarks, with closeness measured using geodesic distance on the skull. The mean and variance are then interpolated linearly, using the inverse geodesic distances as blending weights. The resulting mean thicknesses are visualized in Figure 3 (left). Regions such as the craniocervical junction and the teeth do not have flesh thickness measurements (in these regions, we set the mean to zero and the standard deviation to infinity). For each skull vertex j, we introduce an energy term:

$$E_j^{\text{thickness}}(\mathbf{x}) = \frac{1}{\sigma_j^2}\|\mathbf{n}_j^{\mathsf{T}}(\mathbf{H}_j\mathbf{x} - \mathbf{T}_j\mathbf{x}) - \mu_j\|^2 \tag{2}$$

where σ_j is the standard deviation, $\mathbf{n}_j$ is the skull normal, $\mathbf{H}_j$ is the selector of the skull vertex and $\mathbf{T}_j$ selector of the corresponding skin vertex, and μ_j is the mean flesh thickness. The term $E_j^{\text{thickness}}(\mathbf{x})$ encourages realistic placement of the skull inside the head, see Figure 5. We combine all of the face-specific priors into:

$$E^{\text{prior}} = E^{\text{faceLike}} + E^{\text{faceDist}} + \tau\sum_j E_j^{\text{thickness}} \tag{3}$$

For notational brevity we drop the argument $\mathbf{x}$ which appears in all the terms. The parameter $\tau \geq 0$ expresses the relative confidence in the flesh thickness prior.

For a given set of correspondences, the final volumetric deformation problem can be expressed as the minimization of:

$$E^{\text{total}} = E^{\text{planeDist}} + \alpha E^{\text{attach}} + \beta E^{\text{ARAP}} + \gamma E^{\text{prior}}, \tag{4}$$

where we assume that each energy type is summed over all elements, e.g., $E^{\text{ARAP}}(\mathbf{x}) = \sum_i E_i^{\text{ARAP}}(\mathbf{x})$, with i summing over all tetrahedra. The weights $\alpha \geq 0, \beta \geq 0, \gamma \geq 0$ are used to guide the registration process. The key idea is to start with high regularization (high values of α, β, γ) to obtain an initial guess and progressively reduce the regularization as our correspondences are becoming more and more accurate. Specific parameter values used in our experiments can be found in Section 5.

In terms of numerical optimization, we minimize E^{total} using the local/global solver of Projective Dynamics [BML*14]. We slightly modify the solver in order to handle constraints using Lagrange multipliers, which allows us to avoid collision constraints in a more efficient way, as described in Section 3.5. We denote the final result as $\mathbf{x}_{\text{neutral}}$, see the third column of Figure 7.

3.2. Registration of actor's facial expressions

In the previous section we showed how to deform the volumetric template into $\mathbf{x}_{\text{neutral}}$, which corresponds to the scan of our actor in neutral expression. In this section, we describe how to deform $\mathbf{x}_{\text{neutral}}$ to align with the other expression scans. Specifically, we use 10 expressions such as smile, frown, kiss, sneer, etc. The key difference from the previous section is that the deformation from $\mathbf{x}_{\text{neutral}}$ to the target expression must be physiologically plausible, i.e., achievable by a normal human subject under normal conditions. For example, in Section 3.1 it is accepted to deform the bones, because we are explaining individual subject-specific differences. However, in the next stage the bones must remain rigid, because now we are explaining only shape differences due to facial motion of a given human subject.

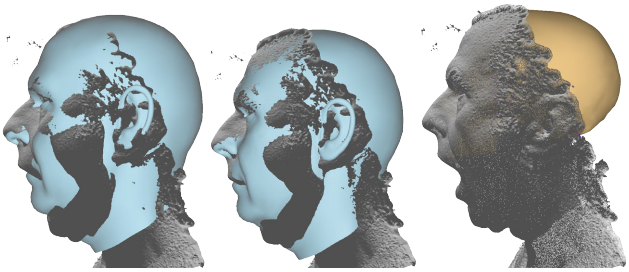
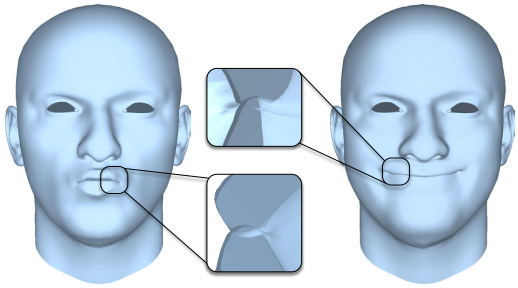
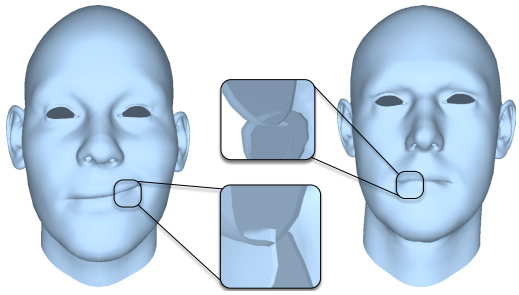


Figure 5: Rigid stabilization using the skull mesh and skin thicknesses. The standard skin registration approach (left) does not compute the correct rigid registration of a mouth open scan, as compared to the skull-based approach (middle and right).



(a) FaceShift [WBLP11] scan registration.



(b) Deformation transfer [SP04].

Figure 6: Most previous methods do not handle self-collisions.

For each facial expression of our actor (Figure 7) we manually find approximate corresponding blendshape weights. This is not too difficult because the actors were instructed to assume specific expressions, which are combinations of only a few blendshapes. We use deformation transfer [SP04] to bootstrap the expression registration process. Assuming a given facial expression, for each triangle of the template surface mesh (2D), we compute the deformation gradient, i.e., the 3D linear transformation between the rest pose and the template expression, using the cross product of the edges to determine the normal, as in [SP04]. Next, we select all surface

tetrahedra from the neutral pose ($\mathbf{x}_{\text{neutral}}$) and define an energy term

$$E_k^{\text{defTransfer}} = \|\mathbf{F}_k - \mathbf{F}_k^{\text{target}}\|_F^2, \tag{5}$$

which attracts the deformation gradients $\mathbf{F}_k$ of all surface tets k of the neutral face ($\mathbf{x}_{\text{neutral}}$) to the deformation gradients $\mathbf{F}_k^{\text{target}}$ calculated from the template model.

Because the template blendshape model explains only the surface, the terms $E_k^{\text{defTransfer}}$ are defined only for tetrahedra adjacent to the boundary. To propagate the surface deformation to the entire volumetric shape, we apply the E^{ARAP} term discussed in Section 3.1 to all of the tets. This term ensures that the surface deformation is distributed throughout the entire volume. During this volumetric deformation, we need to account for the fact that most biological soft tissues are nearly incompressible [WMG96]. We capture this behavior with a new term E^{volume} that is analogous to the ARAP term, except that the projection on $SO(3)$ is replaced with projection of $SL(3)$ – the group of matrices with determinant 1, i.e., volume preserving linear maps. This leads to the objective

$$E^{\text{defTransfer}} + \mu E^{\text{ARAP}} + \lambda E^{\text{volume}}, \tag{6}$$

where the μ and λ are Lamé parameters approximating the elasticity of the flesh. We minimize Equation 6 using Projective Dynamics, keeping the vertices corresponding to the bones fixed (they do not appear as degrees of freedom in the optimization problem). We open the jaw manually by estimating the rigid transformation of the jaw corresponding to the given expression. We denote the result as $\mathbf{x}_{\text{init}}$, which serves as volumetric initialization for the subsequent fitting.

Next, we need to take the actual expression scan into account. As shown by Beeler and Bradley [BB14], it is advantageous to start the fitting process with "rigid stabilization", guided by areas of the skin that are close to the skull and thus not significantly affected by facial expressions. We use an energy analogous to Equation 2, where the mean is set to the actual flesh thickness in $\mathbf{x}_{\text{neutral}}$ and the variance is left out, because at this point we are no longer trying to model variations among different human subjects. We denote this modified objective as $\tilde{E}^{\text{thickness}}$. We find the optimal transformation $\mathbf{T}$ as a composition of rotation, translation, and uniform scale such that $\tilde{E}^{\text{thickness}}(\mathbf{T}\mathbf{x}_{\text{init}})$ is minimized. The uniform scale takes care of the fact that the expression scan from multi-view stereo is in arbitrary units of length.

The resulting "rigidly stabilized" state $\mathbf{T}\mathbf{x}_{\text{init}}$ contains a good estimate of the bone positions and a good initialization of the skin. We are therefore ready to launch the ICP process to account for the subtleties of flesh deformations, while keeping the bones fixed. The deformation energy is analogous to Equation 4:

$$E^{\text{exp-total}} = E^{\text{planeDist}} + \alpha E^{\text{attach}} + \mu E^{\text{ARAP}} + \lambda E^{\text{volume}} \tag{7}$$

Similarly to Section 3.1, the attachment term E^{attach} is found in a semi-automatic way using [SLC11]. Differently from Equation 4, we drop the E^{prior} term because at this stage we are already committed to a given actor. For the same reason, we include the E^{volume} term to enforce incompressibility of the soft tissues.

3.3. Volumetric facial rigging

The expression registration process described in Section 3.2 results in plausible volumetric shapes $\mathbf{x}_{\text{expression},l}$, where l indexes the individual facial expressions. Interpreting $\mathbf{x}_{\text{neutral}}$ (Section 3.1) as the rest pose, we can compute deformation gradients for all tets, mapping from $\mathbf{x}_{\text{neutral}}$ to $\mathbf{x}_{\text{expression},l}$. For each expression, we stack the deformation gradients of all tets into a matrix $\mathbf{H}_l$. Let us denote the vector of blendshape weights for the l-th expression as α_l. These blendshape weights are copied from the template blendshapes and ensure that our volumetric blendshapes will have the same semantics as the template blendshapes. This has the desired consequence that the user intuitively understands how each parameter affects the shape of the face, e.g., that $\alpha_{l,6}$ lowers the right mouth corner etc.

Our next task is to find the volumetric blendshapes. A volumetric blendshape is a collection of deformation gradients for all tets in the face model. Even in the traditional surface case [LWP10], we do not observe the blendshapes directly, because each facial expression $\mathbf{x}_{\text{expression},l}$ is composed of several blendshapes. We find our volumetric blendshapes through a process similar to Example-based Facial Rigging [LWP10] adapted to the volumetric case. Specifically, we solve for volumetric blendshapes $\mathbf{V}_m$ by minimizing:

$$\sum_l \left\| \left(\mathbf{I} + \sum_m \mathbf{V}_m \alpha_{l,m} \right) - \mathbf{H}_l \right\|_F^2 + \kappa \sum_m \|\mathbf{V}_m - \tilde{\mathbf{V}}_m\|_F^2 \tag{8}$$

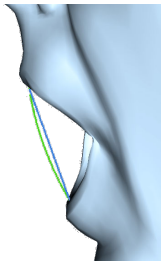
where the addition of stacked identity matrices $\mathbf{I}$ ensures that if all $\alpha_{l,m} = 0$, we obtain the neutral face, corresponding to all deformation gradients equal to identities. In other words, the $\alpha_{l,m}$ are not coefficients of an affine combination, but rather scaling factors of individual blendshapes, interpreted as differences from the neutral pose. In the second term, the $\tilde{\mathbf{V}}_m$ are volumetric blendshapes obtained from deformation transfer of template blendshapes, i.e., minimizing Equation 6. The second term including its weighting coefficient $\kappa \geq 0$ expresses a prior, which is necessary because the first (data) term does not specify the volumetric blendshapes uniquely (in all of our experiments we use $\kappa = 10^{-4}$). This is because we use only a small set of expressions which could be generated by many different volumetric blendshapes. Therefore, we use the second (regularization) term that picks a unique solution – the one that is as close as possible to deformation-transferred template blendshapes.

3.4. Animation

We create new facial animations using a time-varying sequence of blendshape weights $\mathbf{w}(t)$ and rigid head motion $\mathbf{R}(t) \in SE(3)$; the latter specifies the position and orientation of the skull. Even though the jaw motion could be also controlled explicitly, we continue to rely on the blendshape model, which is compatible with standard animation workflows, i.e., the jaw motion is implicitly controlled via blendshape weights instead of explicit control via rigid transformations or a kinematic rig (used by Sifakis and colleagues [SNF05]). The rigidity of the jaw bone will be enforced in the volumetric-blendshape blending process, described below. Our input sequences of the time-varying $\mathbf{w}$ and $\mathbf{R}$ parameters can be either directly keyframed by artists or captured from human subjects using tracking software such as FaceShift [WBLP11].

The blendshape weights can be used to blend the deformation gradients from the individual volumetric blendshapes linearly, $\mathbf{F}^{\text{target}} = \mathbf{I} + \sum_m \mathbf{V}_m \alpha_m$, as in Equation 8. However, it is a well-known fact that linear blending of matrices is prone to artifacts,

especially when the blended transformations contain larger rotations [SD92]. This problem can be avoided by using the polar decomposition method introduced by Shoemake and Duff [SD92]. Specifically, if we have a set of 3×3 matrices $\mathbf{M}_1, \ldots, \mathbf{M}_n$, we first find their polar decompositions, i.e., $\mathbf{M}_i = \mathbf{R}_i\mathbf{S}_i$, where $\mathbf{R}_i$ is a rotation and $\mathbf{S}_i$ is symmetric. The rotations $\mathbf{R}_i$ are then blended non-linearly using quaternions [Sho85]; the "stretch" matrices $\mathbf{S}_i$ are blended linearly, as they correspond to the non-rigid component of the transformation. Finally, the blended rotations and stretch components are multiplied together to create the final result. This approach avoids the loss of volume associated with linear blending of rotations. If the input transformations are pure rotations, as is the case for tets corresponding to the jaw, the blended result will also be a pure rotation, guaranteeing that the jaw bone remains rigid as expected. See the figure on the right for an example: the blue curve is the path of a linearly interpolated vertex for a mouth opening sequence, while the green curve is the path using nonlinear interpolation.



In theory, Equation 8 should be revised for polar decomposition-based blending. In practice, the computation of polar decomposition inside the objective would require more complicated numerical solution procedures and therefore, we continue to rely on Equation 8. This linear approximation seems to be sufficient for the purpose of determining volumetric blendshapes.

If we denote the deformation gradients computed by polar-decomposition-blending as $\mathbf{F}^{\text{target}}$, we can create a "targeting" energy term:

$$E^{\text{target}}(\mathbf{x}) = \|\mathbf{F}(\mathbf{x}) - \mathbf{F}^{\text{target}}\|_F^2 \tag{9}$$

where $\mathbf{F}$ is a linear function of $\mathbf{x}$ [SB12]. This energy specifies that all deformation gradients $\mathbf{F}$ of the unknown mesh state $\mathbf{x}$ are attracted to $\mathbf{F}^{\text{target}}$. Intuitively speaking, the E^{target} term serves the same purpose as muscle activations in full anatomical models [SNF05], however, without the need of modeling the geometry and mechanics of individual muscles. While we avoid the intricacies of full anatomical modeling, we retain the possibility of introducing additional energy potentials and constraints. For example, dynamic effects can be easily added using an "inertial" term $E^{\text{inertia}}(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\mathbf{y})^{\mathsf{T}}\mathbf{M}(\mathbf{x}-\mathbf{y})$, where $\mathbf{M}$ is the mass matrix and $\mathbf{y}$ is state predicted by Newton's first law, i.e., motion without the presence of forces. This term is equivalent to the variational Implicit Euler formulation used in Projective Dynamics [BML*14]. Perhaps even more useful is the ability to add constraints due to collisions with the face itself, e.g., lips-lips or lips-teeth collisions, or external objects. Our approach to handling contact involves a modification of the Projective Dynamics solver which is described in the following section.

Stronger inertial or contact forces can result in shapes with deformation gradients significantly departing from the targeting term E^{target}. In order to preserve realistic behavior of the soft tissues even in these large deformations, we add the $\mu E^{\text{ARAP}} + \lambda E^{\text{volume}}$ terms, as in Equation 6. This has a natural biomechanic interpretation as the elasticity of passive soft tissues [TSIF05]. Intuitively, if there is, e.g., a large external force acting on the cheek, this force is propagated through the entire musculoskeletal system. For tets corresponding to the skull and the jaw, we use stiffness high enough to prevent any visible deformations of the bones (specifically, we use $\mu = 1000$).

3.5. Collisions

Our collision processing mechanism is based on point-to-plane constraints which are dynamically instanced as needed to resolve collisions, analogous to classical collision resolution approaches [MZS*11]. To detect inter-penetrations, we use a fast bounding box sequence intersection algorithm [ZE00] for the broad phase, and an AABB tree built in the rest pose. For efficiency, only certain pairs of regions of the face are checked against collisions (e.g., lips against lips, lips against skull, skin against external objects). When colliding with external objects, our current implementation assumes these external objects are fixed, e.g., directly controlled via keyframing. In either case, if we detect a collision, i.e., a vertex penetrating a tetrahedron, we find the closest surface point where the vertex needs to move in order to resolve the collision. To facilitate sliding, we create a constraint which requires the offending vertex to align with a tangent plane at the closest surface point. In case of both self-collisions and external collisions, this can be expressed as affine equality constraint $\mathbf{C}_i\mathbf{x} = \mathbf{d}_i$, where i indexes contact points. We append all of the collision constraints together: $\mathbf{C}\mathbf{x} = \mathbf{d}$. The main challenge in efficient collision processing is the fact that the collision constraints $\mathbf{C}\mathbf{x} = \mathbf{d}$ are frequently changing.

The original Projective Dynamics paper [BML*14] proposes two options. The first is to directly add energy terms penalizing violation of the collision constraints. Unfortunately, this requires re-computing the factorization of the global step matrix, resulting in significant computational overheads. The second option is to add these constraints for all vertices in the system and pre-factorize only once, because changing the target positions or planes of the constraints affects only the right hand sides. The undesired side-effect is that these constraints affect the behavior of the system even if there are no collisions. The collision constraints are always present in the system, and even if they are not active, they attract the vertices towards their current locations. In practice, this introduces additional damping, slowing down convergence in the quasi-static case and creating artificial viscosity in the dynamic case.

To avoid these drawbacks, we propose a new method, motivated by the observation that the number of colliding vertices is typically small, because the collision resolution process is invoked each iteration. The key idea is to apply the Schur complement [Jac13, YCP16] to reuse the pre-computed factorization without introducing any artificial damping. First, recall that the global step of Projective Dynamics solves a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{A}$ is a constant symmetric positive definite matrix. Therefore, Projective Dynamics pre-computes a sparse Cholesky factorization of $\mathbf{A}$ that allows calculating $\mathbf{A}^{-1}\mathbf{b}$ very efficiently as long as $\mathbf{A}$ is not changing.

We propose to incorporate our frequently changing collision constraints $\mathbf{C}\mathbf{x} = \mathbf{d}$ using Lagrange multipliers. This leads to the KKT system, named after the famous Karush-Kuhn-Tucker optimality conditions [NW06]:

$$\begin{bmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix} \tag{10}$$

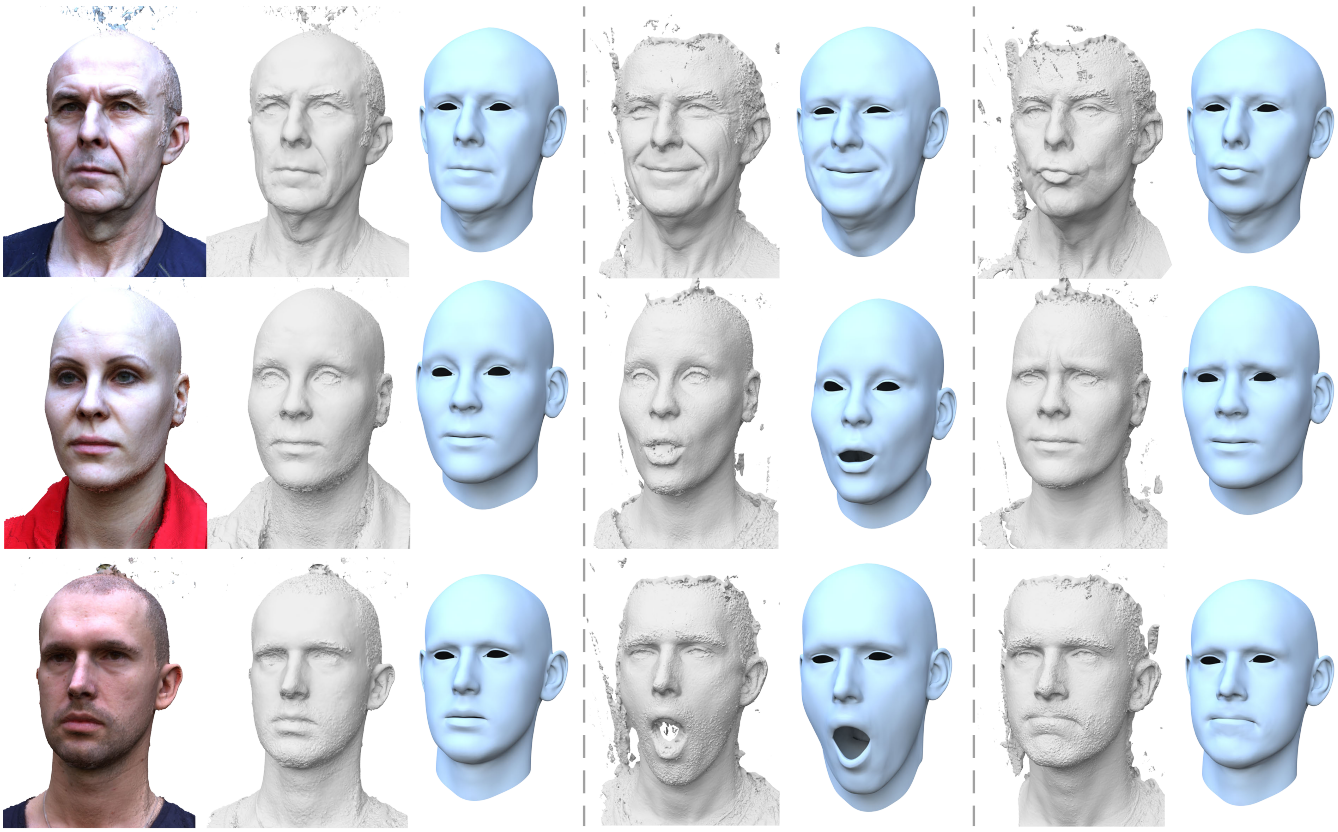


Figure 7: Registration of 3D scans of our test subjects: neutral pose (left) and two facial expressions (middle, right).

One possible way to solve this system while taking advantage of the existing factorization of $\mathbf{A}$ would be using low-rank updates [CDHR08]. Unfortunately, in our case the cost of low-rank updates is comparable or even greater to the cost of factorizing the KKT system from scratch. Instead, we propose to solve for the Lagrange multipliers using the Schur complement of Equation 10: $\mathbf{C}\mathbf{A}^{-1}\mathbf{C}^{\mathsf{T}}\boldsymbol{\lambda} = \mathbf{C}\mathbf{A}^{-1}\mathbf{b} - \mathbf{d}$. The matrix $\mathbf{C}\mathbf{A}^{-1}\mathbf{C}^{\mathsf{T}}$ is dense but small, because we assume the number of rows of $\mathbf{C}$ is small; in our simulations, it is typically less than 50. The solve for $\boldsymbol{\lambda}$ is therefore fast even with dense linear algebra. Having found $\boldsymbol{\lambda}$, we can compute the solution $\mathbf{x} = \mathbf{A}^{-1}(\mathbf{b} - \mathbf{C}^{\mathsf{T}}\boldsymbol{\lambda})$.

4. Corrective blendshapes

In some cases, physics-based facial animation may not be desirable, e.g., in 3D game engines which require extremely fast animation algorithms. In this case, our approach can be used as an automatic method to generate corrective blendshapes, which is a common way to address the problems of linear blendshape models [LAR*14]. We focus on the basic case of quadratic blendshapes, even though higher-order methods are also possible. The key idea is to sample activations of every pair of blendshapes. For each pair, we sample activations of each of the two blendshapes; we use four steps for the first weight: $0.25, 0.5, 0.75, 1$ and five for the second one: $0, 0.25, 0.5, 0.75, 1$, leading to a total of 20 samples per pair. We denote the final sequence of $20\binom{b}{2}$ blendshape weights samples as $\mathbf{w}_1, \mathbf{w}_2, \ldots$, where the number of blendshapes in our case is $b = 29$. For each of them we synthesize a realistic face shape using our method, as described in Section 3, and denote the coordinates of the resulting skin vertices as $\mathbf{p}_1, \mathbf{p}_2, \ldots$. Our goal is to explain these example face shapes $\mathbf{p}_k$ using the quadratic blendshape model. This task can be formulated as an optimization problem:

$$\underset{\mathbf{m},\mathbf{u}_i,\mathbf{v}_{ij}}{\arg\min} \sum_k \left\| \mathbf{m} + \sum_i w_{k,i}\mathbf{u}_i + \sum_i \sum_j w_{k,i} w_{k,j} \mathbf{v}_{ij} - \mathbf{p}_k \right\|^2 \qquad (11)$$

where $\mathbf{m}$ is the mean, corresponding to neutral facial expression, $\mathbf{u}_i$ are traditional linear blendshapes and $\mathbf{v}_{ij}$ are the quadratic blendshapes. We find the optimal $\mathbf{m}, \mathbf{u}_i, \mathbf{v}_{ij}$ by solving a linear least squares problem.

5. Implementation and results

The geometric search data structures and algorithms used for registration and collision detection are based on CGAL. Our optimization framework is an extension of the open-source ShapeOp [DDB*15]. Numerical linear algebra is handled using Eigen. Our current prototype runs on the CPU, parallelized using OpenMP. We benchmark the performance on a consumer laptop with a 2.5 GHz Intel Core i7 processor and 16GB of main memory. In our experiments, the

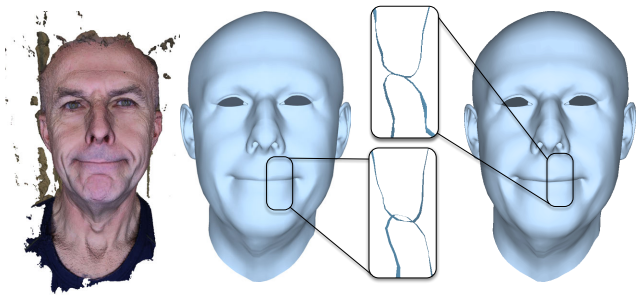


Figure 8: Our collision handling (right) avoids inter-penetrations during expression registration.

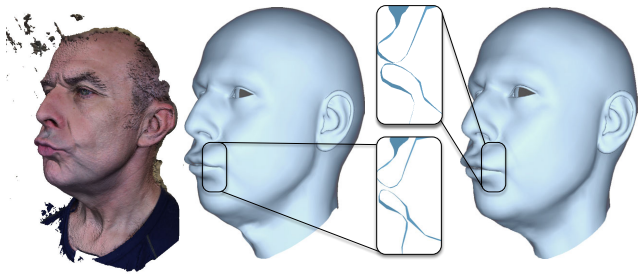


Figure 9: Volume preservation allows us to achieve more natural expression registration (right). To the left is the result without volume preservation.

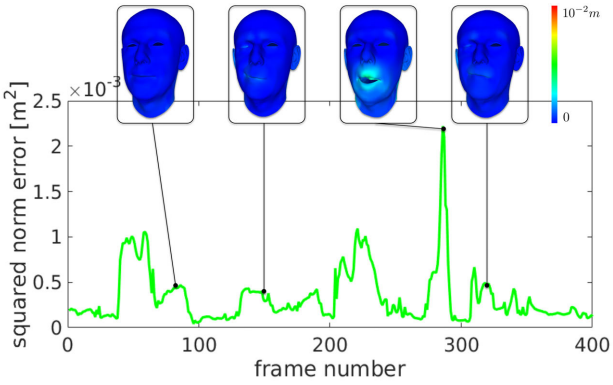


Figure 10: The difference between the blendshape animation and our physically simulated animation, expressed as the squared norm error between each mesh for each frame of a sequence. Note that the spikes appear when large non-linear motion is present (e.g., frame 280), or when collisions are present (e.g. frames 90, 155, 330).

animation converged using 6 iterations per frame. The timing per frame ranges from 500ms if no collisions are detected up to 1200ms when the lips collide heavily (about 80 collision constraints at a time, like in the *chewing* sequences shown in the supplementary video). The template volumetric model has 7366 vertices and 14600 triangles for the skin surface, 8947 vertices and 36654 tetrahedra for the flesh, 6760 vertices and 29888 tetrahedra for the bones. We use the same anatomical template for all of our actors.

Registration. For registration of the neutral face expression (Section 3.1), we used the following parameters: $\alpha = 10^1, \beta = 10^1, \gamma = 10^{-2}, \tau = 10^1$. We captured three different human subjects, all of them experienced actors. The input neutral scans and our resulting registered templates are shown in Figure 7 (left). In addition to the neutral expression, for each actor we also captured 10 facial expressions and executed the expression fitting algorithm described in Section 3.2 with parameters $\mu = 10^2$ and $\lambda = 10^3$. The results for two different expressions can be seen in Figure 7 (middle and right). Our registration technique takes advantage of collision constraints to avoid self-penetrations, see Figure 8. Similarly, the volume preservation terms used in the expression registration process help us avoid unnatural deformations, as shown in Figure 9. Because the inside of the mouth is not visible and therefore not captured by 3D scanning methods, previous techniques that do not account for incompressibility of the flesh can deform the lips into unnaturally thin shapes. Furthermore, volume preservation helps to establish the lip contact surface, which is difficult to determine using optical methods due to occlusions.

Animation. We invite the reader to watch the accompanying video, showing facial animation sequences generated by our system. In particular, certain types of facial expressions frequently produce self-intersections of the lips with traditional blendshape models. Our method successfully removes these inter-penetrations while departing from the original blendshape model as little as possible, see Figure 10.

In addition to traditional facial motion driven purely by muscle activations, our method allows incorporating external forces. In Figure 11 (left), as well as in the accompanying video, we show a talking sequence with part of the bottom lip held fixed. Our simulator can also naturally deliver dynamic effects, including stylized animations such as shockwave propagation through the skin or making the nose more heavy while swinging the head, see Figure 11 (middle). Perhaps even more entertaining are collisions with external objects, such as the boxer glove in Figure 11 (right). Note that the nose bridge does not deform due to the presence of the bone in this region, unlike the rest of the nose.

Corrective blendshapes. We use 8120 samples corresponding to activating all pairs of blendshapes at different activation levels (Section 4), resulting in 406 quadratic blendshapes which require additional 65MB of memory (in addition to 7.7MB for the linear

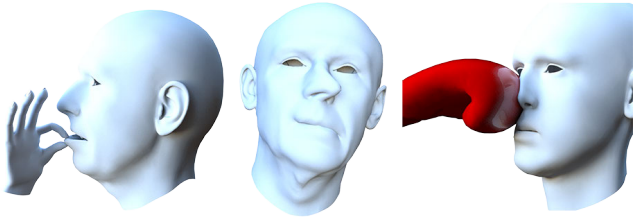


Figure 11: Our method allows us to incorporate external forces and dynamic effects.

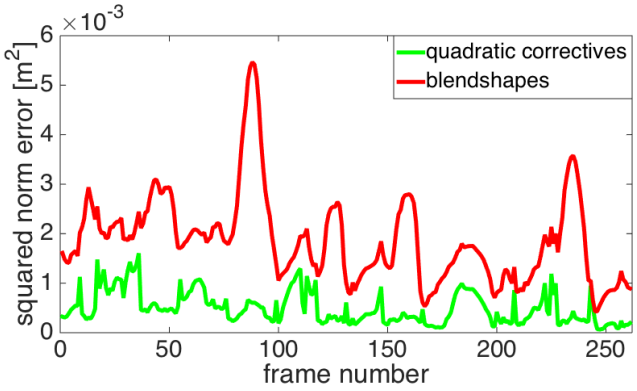


Figure 12: Error decrease when using blendshapes against our trained quadratic corrective blendshapes on an animation sequence.

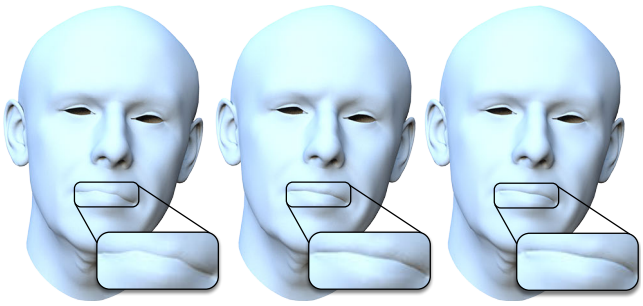


Figure 13: An example of the handling of self-collisions via corrective blendshapes. From left to right: linear blendshapes, quadratic correctives, simulation.

blendshapes). The runtime increases from 1ms for linear-only blendshapes to 8ms, which is acceptable even in real-time applications such as games. To compare the accuracy of quadratic vs. linear blendshapes, we measured for each frame of an animation sequence the error between the full simulated model and an approximation computed by 1) linear and 2) quadratic blendshapes. The resulting plot is shown in Figure 12. The quadratic blendshapes significantly reduce the error compared to the linear ones. Even though we cannot guarantee collision-free results, the quadratic blendshape model is quite effective in avoiding visible self-penetrations, as demonstrated in Figure 13. A limitation of quadratic blendshapes is the fact that they are not able to capture previously unseen external forces, such as collisions with external objects.

6. Conclusion

We introduced a method for creating personalized volumetric face rigs that combine the intuitive control of blendshapes with the improved realism of physics-based simulation. Specifically, our face animation supports volume preservation, avoids self-collisions, and enables dynamic effects due to external forces. These improvements in animation quality come at the cost of increased computation time. To alleviate this performance loss, we show how the simulated face model can be used to automatically create corrective blendshapes. While these cannot guarantee the same level of accuracy as the full simulation model, significant quality improvements are achieved with a low computational overhead compared to the initial blendshape model.

Building a volumetric face rig based on high-resolution surface scans requires advanced registration algorithms to mitigate errors caused by the inherent limitations of the optical 3D scanning process, such as occlusions. We show how the same underlying optimization framework used for animation can be applied effectively for volumetric registration as well. This unification of representation and optimization leads to a simple and robust implementation based on existing open-source software.

As the quest for more realism continues, we believe that reducing the complexity of facial rigging will be crucial for wide-spread adoption in computer gaming, movie production, VR and avatar-based online communication. Interesting future challenges lie in further simplifications of the acquisition process, in building more advanced volumetric priors for effective model reconstruction, and in more efficient simulation methods for realtime animation of volumetric face rigs.

7. Acknowledgements

We thank the anonymous reviewers for their feedback and constructive criticism. We would also like to thank Sofien Bouaziz, Matthew Cong, Ron Fedkiw, Eftychios Sifakis, and Peter Shirley for valuable discussions and feedback. This project was supported in part by NSF awards IIS-1622360 and IIS-1350330 and a gift from Activision. Furthermore, we would love to acknowledge the help received from the actors who accepted to be scanned for the purpose of this project: Peter Ender, Jördis Wölk, and Michael Schönert, as well as Anton Rey for the coordination and acting advice.

References

[ABF*07] AMBERG B., BLAKE A., FITZGIBBON A., ROMDHANI S., VETTER T.: Reconstructing high quality face-surfaces using model based stereo. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (2007), IEEE. 3

[ARL*10] ALEXANDER O., ROGERS M., LAMBETH W., CHIANG J.-Y., MA W.-C., WANG C.-C., DEBEVEC P.: The digital emily project: Achieving a photorealistic digital actor. *Computer Graphics and Applications, IEEE* (2010). 3

[BB14] BEELER T., BRADLEY D.: Rigid stabilization of facial expressions. *ACM Transactions on Graphics (TOG)* (2014). 3, 4, 6

[BBB*10] BEELER T., BICKEL B., BEARDSLEY P., SUMNER B., GROSS M.: High-quality single-shot capture of facial geometry. *ACM Transactions on Graphics (TOG)* (2010). 4

[BBK*15] BERMANO A., BEELER T., KOZLOV Y., BRADLEY D., BICKEL B., GROSS M.: Detailed spatio-temporal reconstruction of eyelids. *ACM Transactions on Graphics (TOG)* (2015). 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)* (2014). 2, 3, 4, 5, 7

[BSC16] BARRIELLE V., STOIBER N., CAGNIART C.: Blendforces, a dynamic framework for facial animation. *Comput. Graph. Forum* (2016). 3

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co. 3, 4

[BWP13] BOUAZIZ S., WANG Y., PAULY M.: Online modeling for realtime facial animation. *ACM Transactions on Graphics (TOG)* (2013). 3

[CBB*15] CONG M., BAO M., BHAT K. S., FEDKIW R., ET AL.: Fully automatic generation of anatomical face simulation models. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2015), ACM. 3

[CDHR08] CHEN Y., DAVIS T. A., HAGER W. W., RAJAMANICKAM S.: Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software (TOMS)* (2008). 8

[CPSS10] CHAO I., PINKALL U., SANAN P., SCHRÖDER P.: A simple geometric model for elastic deformations. In *ACM Transactions on Graphics (TOG)* (2010), ACM. 4

[CWZ*14] CAO C., WENG Y., ZHOU S., TONG Y., ZHOU K.: Facewarehouse: a 3d facial expression database for visual computing. *Visualization and Computer Graphics, IEEE Transactions on* (2014). 3

[DDB*15] DEUSS M., DELEURAN A. H., BOUAZIZ S., DENG B., PIKER D., PAULY M.: ShapeopâĂŤa robust and extensible geometric modelling paradigm. In *Modelling Behaviour*. Springer, 2015. 2, 8

[DGCV*06] DE GREEF S., CLAES P., VANDERMEULEN D., MOLLEMANS W., SUETENS P., WILLEMS G.: Large-scale in-vivo caucasian facial soft tissue thickness database for craniofacial reconstruction. *Forensic science international* (2006). 4, 5

[DLG*13] DICKO A.-H., LIU T., GILLES B., KAVAN L., FAURE F., PALOMBI O., CANI M.-P.: Anatomy transfer. *ACM Transactions on Graphics (TOG)* (2013). 3

[GVWT13] GARRIDO P., VALGAERTS L., WU C., THEOBALT C.: Reconstructing detailed dynamic face geometry from monocular video. *ACM Trans. Graph.* (2013). 3

[IBP15] ICHIM A. E., BOUAZIZ T., PAULY M.: Dynamic 3d avatar creation from hand-held video input. *ACM Trans. Graph. (Proc. SIGGRAPH)* (2015). 3

[Jac13] JACOBSON A.: *Algorithms and interfaces for real-time deformation of 2d and 3d shapes*. PhD thesis, ETH, 2013. 7

[JKSH13] JACOBSON A., KAVAN L., SORKINE-HORNUNG O.: Robust inside-outside segmentation using generalized winding numbers. *ACM Transactions on Graphics (TOG)* (2013). 4

[KHS03] KÄHLER K., HABER J., SEIDEL H.-P.: Reanimating the dead: reconstruction of expressive faces from skull data. In *ACM Transactions on Graphics (TOG)* (2003), ACM. 3

[LAR*14] LEWIS J. P., ANJYO K., RHEE T., ZHANG M., PIGHIN F. H., DENG Z.: Practice and theory of blendshape facial models. In *Eurographics (State of the Art Reports)* (2014). 1, 8

[LBOK13] LIU T., BARGTEIL A. W., O'BRIEN J. F., KAVAN L.: Fast simulation of mass-spring systems. *ACM Transactions on Graphics* (2013). Proceedings of ACM SIGGRAPH Asia 2013, Hong Kong. 3

[LSP08] LI H., SUMNER R. W., PAULY M.: Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum* (2008). 4

[LWP10] LI H., WEISE T., PAULY M.: Example-based facial rigging. In *ACM Transactions on Graphics (TOG)* (2010), ACM. 2, 6

[MHHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *Journal of Visual Communication and Image Representation* (2007). 3

[MWF*12] MA W.-C., WANG Y.-H., FYFFE G., CHEN B.-Y., DEBEVEC P.: A blendshape model that incorporates physical interaction. *Computer Animation and Virtual Worlds* (2012). 3

[MZS*11] MCADAMS A., ZHU Y., SELLE A., EMPEY M., TAMSTORF R., TERAN J., SIFAKIS E.: Efficient elasticity for character skinning with contact and collisions. In *ACM Transactions on Graphics (TOG)* (2011), ACM. 7

[NW06] NOCEDAL J., WRIGHT S.: *Numerical optimization*. Springer Science & Business Media, 2006. 7

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on* (2001), IEEE. 4

[SB12] SIFAKIS E., BARBIC J.: Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses* (2012), ACM. 4, 7

[SD92] SHOEMAKE K., DUFF T.: Matrix animation and polar decomposition. In *Proceedings of the conference on Graphics interface* (1992). 7

[Sho85] SHOEMAKE K.: Animating rotation with quaternion curves. In *ACM SIGGRAPH computer graphics* (1985), ACM. 7

[SLC11] SARAGIH J. M., LUCEY S., COHN J. F.: Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision* (2011). 4, 6

[SNF05] SIFAKIS E., NEVEROV I., FEDKIW R.: Automatic determination of facial muscle activations from sparse motion capture marker data. In *ACM Transactions on Graphics (TOG)* (2005), ACM. 3, 6, 7

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. In *ACM Transactions on Graphics (TOG)* (2004), ACM. 3, 5

[SWTC14] SHI F., WU H.-T., TONG X., CHAI J.: Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM Transactions on Graphics (TOG)* (2014). 3

[SZGP05] SUMNER R. W., ZWICKER M., GOTSMAN C., POPOVIĆ J.: Mesh-based inverse kinematics. In *ACM transactions on graphics (TOG)* (2005), ACM. 3

[TSIF05] TERAN J., SIFAKIS E., IRVING G., FEDKIW R.: Robust quasistatic finite elements and flesh simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2005), ACM. 7

[VCL*06] VANDERMEULEN D., CLAES P., LOECKX D., DE GREEF S., WILLEMS G., SUETENS P.: Computerized craniofacial reconstruction using ct-derived implicit surface representations. *Forensic science international* (2006). 3

[WBLP11] WEISE T., BOUAZIZ S., LI H., PAULY M.: Realtime performance-based facial animation. In *ACM Transactions on Graphics (TOG)* (2011), ACM. 3, 4, 5, 6

[WMG96] WEISS J. A., MAKER B. N., GOVINDJEE S.: Finite element implementation of incompressible, transversely isotropic hyperelasticity. *Computer methods in applied mechanics and engineering* (1996). 6

[YCP16] YEUNG Y.-H., CROUCH J., POTHEN A.: Interactively cutting and constraining vertices in meshes using augmented matrices. *ACM Trans. Graph.* (2016). 7

[ZE00] ZOMORODIAN A., EDELSBRUNNER H.: Fast software for box intersections. In *Proceedings of the sixteenth annual symposium on Computational geometry* (2000), ACM. 7

[ZHK15] ZHU L., HU X., KAVAN L.: Adaptable anatomical models for realistic bone motion reconstruction. *Comput. Graph. Forum* (2015). 3

[Zyg16] ZYGOTE: Zygote body, 2016. [Online; accessed 6-March-2016]. URL: https://zygotebody.com. 4