Computers & Graphics (2020)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Solving for Muscle Blending Using Data

Dimitar Dinev[a], Wenxian Guo[a], Petr Kadleček[a,b], Ladislav Kavan[a]

[a] *School of Computing, University of Utah, USA*
[b] *Charles University, Prague, Czech Republic*

ARTICLE INFO

Article history:
Received 2020-09-23

Keywords: physics-based simulation, facial animation, anatomical models

ABSTRACT

Modeling of the human face is a challenging yet important problem in computer graphics. Building accurate muscle models for physics-based simulation of the face is a problem that either requires a lot of manual effort or drastic over-parameterization of the muscles to achieve desirable results. In this work, we reduce the number of parameters required to build personalized muscle models by taking into account the blending of the fine muscles and passive tissue when we solve for the muscle activations. We begin by adapting an anatomical template model to a neutral scan of a subject. Then, we solve an inverse physics problem using several scans simultaneously to solve for both the muscle activations and the geometry matrix representing blending of the muscles. Finally, we demonstrate that this geometry matrix can be used on new, previously unseen scans to solve for only the muscle activations. This greatly reduces the number of parameters that must be solved for compared to previous works while requiring no additional manual effort in constructing the muscles.

1. Introduction

The modeling of the human body is a very important problem in computer graphics, with wide applications ranging from the entertainment industry to medicine. In recent years, advances in facial tracking and capture technologies [1, 2, 3] have achieved impressive results and are becoming widely used in the film industry. However, anatomical modelling of the face is particularly difficult, due to the high levels of nonlinear deformations exhibited on the surface of the skin caused by complex interactions between the underlying facial muscles and the passive tissue (e.g. fat).

Traditional approaches to creating facial animations usually rely on direct deformation models which only take into account the surface of the face, such as blendshapes [4]. These techniques are favored for their ease of use and extensive support, but they are not without shortcomings. While they can create believable facial expressions, it is difficult to add physics-based effects to them (i.e., inertia, gravity, collisions). Furthermore, it is possible to produce unrealistic expressions, requiring corrective to fix artifacts.

Physics-based models attempt to remedy these artifacts by attempting to mimic the real-world biomechanics of facial expression generation. Instead of directly using the surface of the face, anatomical face models attempt to simulate the underlying muscle structure which causes the surface deformations [5]. This is a very challenging problem due to the fact that it is difficult to acquire such information for a given subject: all of the muscles and fat lie beneath the skin and require advanced medical imaging machines to observe, making it impractical for general use. Thus, physics-based models usually require significant manual effort to create a muscle structure for a given subject. While this can produce good results which are used in high-budget feature films, it requires many man-hours on the part of technical artists to create a good model for just one subject.

Recent works have attempted to remedy this problem by attempting to infer the underlying muscle structure [6, 7] using only surface scans. This is accomplished by setting up an inverse solve which tries and finds the correct muscle activations that are able to fit a given expression in a scan. This makes the method

more practical for general use, but sacrifices the physical interpretations of the muscles by over-parameterizing them in order to fit the scans. More concretely, the number of parameters is a factor for the number of volumetric elements (i.e., tetrahedra); each element acts as an independent muscle.

In this work, we attempt to alleviate this issue by drastically reducing the number of parameters in the facial muscle model to more closely match the real-world muscles. We accomplish this by observing that any volumetric element (i.e., tetrahedra) in the face can contain several muscles as well as passive tissue. We extract this geometric information into a separate variable from the muscle activations, telling us how all of these elements are blended together. We then solve for the blending variable by simultaneously fitting several different target scans. By separating out this blending information, the muscle activations we solve for are closer to the real-world mechanics of the face; the number of unknowns is a function of number of muscles instead of number of elements. We then show that this blending variable can be reused to successfully fit new, unseen expressions scans of the same subject.

This method retains the advantages of previous methods in that it is still fully automatic and does not require any manual muscle construction, but by extracting the subject-specific muscle geometry from the muscle activations we are able to greatly reduce the space of the activations. A bonus side-effect of this is that the activation-only solves are much faster due to the drastically lower number of unknowns. The muscle activations can also still be used as keyframes to create "volumetric blendshapes" that are responsive to physics-based effects such as wind and changes to our subject's physical parameters (i.e., adding fat). We believe that solving for the muscle geometry is an important step towards having a full personalized physics-based model that is easily obtainable for any given subject, without requiring significant manual effort.

2. Related Work

Facial animation has been an important part of computer graphics research since seminal work in the 1990's [8, 9]. Human facial expressions are extremely important in communication [10], and accurate representations of the human face is vital in both the entertainment industry and the medical field.

Data-Driven Methods. The traditional facial animation technique used by artists is geometry-based blendshapes [4, 11, 12, 13, 14]. This allows for artists to create facial expressions geometrically blending several target expressions. This approach has the advantage of being simple to implement, intuitive for artists, and capable of creating a large variety of facial expressions. However, the linear blending of the surface can produce some non-physical shapes and has difficulty handling phenomena such as contact. Physics-based approaches have been used to augment geometric approaches. The work of [15] augmented the blendshape framework with mass-spring systems, which have been further augmented with a simulated surface [16], or a volumetric layer on top of a blendshape model [17, 18].

Blendshapes are usually created by employing capture techniques to get the target expressions. Facial capture systems have seen a significant increase in quality in recent years and are widely used in the entertainment industry. Marker-based techniques [19] and recently dense markerless techniques [20, 1, 2, 21, 22, 23, 24] can produce a very accurate reconstruction of a facial performance in multiview [25] and even monocular [26, 27] capture setups. Modern systems can also accurately capture traditionally difficult areas such as the eyelids [28], eyes [29, 30], hair [31, 32, 33], teeth [34], lips [35, 36, 37], and skin microstructure [38, 39]. Anatomy-based techniques have also been used to improve the tracking, such as skull transformations [3], jaw tracking [40, 41] and deformation models[42].

Physics-based Approaches. Anatomical modeling of the body is an active area of research with many simulation platforms, such as ArtiSynth [43], SOFA [44], FEBio [45], and Abaqus [46]. Extracting the biomechanical parameters of the biological tissue is an ongoing problem in biomechanics, with many measurements [47] and models [48, 49] being developed. Instead of excising the soft tissue and measuring the forces via torsion or suction tests, computer vision techniques can be applied to estimate material parameters [50, 51, 52, 53] by visually studying the deformation. Similarly, surface electromyography [54] provides a less intrusive option for measuring muscle activations.

Many muscle models have been proposed for full-body animations [55, 56, 57, 58, 59]. Various numerical strategies have been developed, such as Eulerian-on-Lagrangian-based [60, 61] and projective-dynamics-based [62, 63, 64] approaches. Facial muscles are fundamentally different from skeletal muscles such as the biceps brachii [65], posing significant challenges due to their small, thin nature and complicated attachments. Muscle simulation for bodies usually studies the muscles' effects on the bone, while facial simulations try to explain the subtle deformations on the surface of the skin. Building person-specific muscle models has been an active field of research [5], usually requiring significant manual sculpting of the muscles. Similarly, MRI scans have been used to extract muscle information from a subject [66], again requiring significant manual tuning and editing to create a useful model. Muscle models that allow artists to manually sculpt expressions by editing muscle activations have been introduced [67] and further improved [68, 69]. Stavness et al. introduced an automated method for constructing a person's geometry using CT data [70], which is not trivial to obtain. Cong et al. introduced a method to geometrically adapt muscles from a template to a target using only the neutral scans [71], however they do not use target expressions scans to solve for muscle activations. The work of Ichim et al. introduced a physics-based equivalent to blendshapes [6], which overcomes blendshapes' limitations in handling physics-based phenomena such as collisions and gravity. This framework was further extended by adding a rotationally-invariant muscle activation model [7] and by incorporating heterogeneous material parameters and more complicated phenomena such as pre-strain [72]. However, these activations lose their physical meaning, treating each tet as an independent muscle.

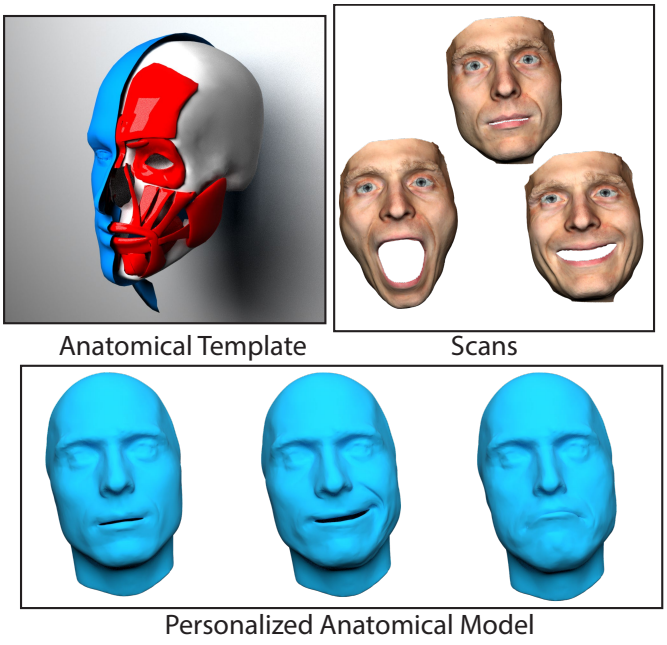

Fig. 1. An animation may be produced by linearly interpolating the activations of two expressions.

3. Method

3.1. Data preparation

We begin with a template face model, which includes the bones (skull and mandible), muscles, and a flesh mesh, shown in Figure 1 (top left). We then capture a series of registered scans of our subject (Figure 1, top right), including a neutral expression which we can use with the method of Anatomy Transfer [73] to transfer the anatomy of the template to our subject, adapting the meshes representing the bones, muscles, and flesh to fit our subject. Tetgen [74] is then used to create a tetrahedral mesh of the flesh (Figure 1, bottom). Similar to [7], we use the tetrahedral flesh mesh and the muscle meshes to find "active" tetrahedra. However, instead of storing a binary "active" or "passive" value, for every tetrahedron we store a list of integers representing the indices associated with all of the muscles that particular tetrahedron intersects with. We use 0 to represent "passive" tissue, and all tetrahedra contain at least this value. Additionally, since the muscles obtained from [73] are only an estimate and can differ from the subject's actual muscle geometry, we also store muscles whose meshes are located in a 1cm radius of the tetrahedron, to account for the potential of these muscles to shift around in a different subject compared to our template. This information is used as an initialization and will be valuable in Section 3.4.

3.2. Forward simulation of face models

We begin with a tetrahedral face mesh of a subject in the rest pose, with p vertices and n tetrahedra. We can describe the activation of a tetrahedron using an augmented finite-element-like elastic potential, such as corotated linear elasticity [7]:

$$E_{act}(\mathbf{x}, \mathbf{A}) = \sum_{i=1}^{n} \min_{\mathbf{R}_i \in \mathrm{SO3}} \frac{\mu_i}{v_i} \|\mathbf{F}_i - \mathbf{R}_i \mathcal{S}(\mathbf{a}_i))\|_{\mathbf{F}}^2 + \frac{\lambda_i}{2v_i} \mathrm{Tr}^2(\mathbf{P} - \mathbf{I}) \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^{p\times 3}$ is a matrix of the mesh vertex positions in the deformed pose, $\mathbf{A} \in \mathbb{R}^{nx6}$ is a global activation vector storing the activation degrees of freedom for all tetrahedra, and $\mathbf{a}_i \in \mathbb{R}^6$ is *ith* row of $\mathbf{A}$ (which we will be solving for in Section 3.3). $\mathbf{F}_i(\mathbf{x}, \mathbf{x}_{rest})$ is the deformation gradient for the i-th tetrahedron (abbreviated as $\mathbf{F}_i$), and matrices $\mathbf{R}_i \in \mathbb{R}^{3\times 3}$ and $\mathbf{P} \in \mathbb{R}^{3\times 3}$ come from the polar decomposition of the deformation gradient $\mathbf{F}_i = \mathbf{R}_i\mathbf{P}_i$. $\mathcal{S} : \mathbb{R}^6 \to \mathbb{R}^{3\times 3}$ is an operator which takes a 6 DoF vector and returns a symmetric matrix (as in to [7]), v_i is the rest volume of i-th tetrahedra, λ_i, μ_i are Lame's first and second parameter, respectively, and $\|\cdot\|_F$ denotes the Frobenius norm. We have added a linear elasticity-based *lambda* term to enforce some notion of volume conservation.

To determine vertex positions of the forward simulation result, we need to solve the following optimization problem, keeping $\mathbf{A}$ constant:

$$\min_{\mathbf{x},\mathbf{b}} \quad E_{act}(\mathbf{x}, \mathbf{A}) + E_{ext}(\mathbf{x}), \quad (2a)$$

$$\text{s.t.} \quad \mathbf{c}(\mathbf{x}, \mathbf{b}) = 0 \quad (2b)$$

where $E_{ext}(\mathbf{x})$ is external energy due to forces such as gravity. The constraints $\mathbf{c}(\mathbf{x}, \mathbf{b})$ correspond to physiological constraints like muscles attaching to the bones. Similar to [7], we model and solve for the jaw kinematics using a 5-DoF rigid transformation model which allows for rotation along only 2 axes and translation along 3 and stacks the five total degrees of freedom in a vector $\mathbf{b} \in \mathbb{R}^5$.

3.3. Inverse model

While the forward problem is simple to solve, obtaining the activation matrix $\mathbf{A}$ is difficult. We want to find an $\mathbf{A}$ matrix such that we are able to explain target scans that we have taken of a subject. We introduce a target energy:

$$E_{target}(\mathbf{x}, \mathbf{t}) = \|\mathbf{M}\mathbf{x} - \mathbf{N}\mathbf{t}\|_{\mathbf{F}}^2 \quad (3)$$

where $\mathbf{x} \in \mathbb{R}^{p\times 3}$ is our tetrahedral mesh's vertices and $\mathbf{t} \in \mathbb{R}^{o\times 3}$ are vertex positions from our target scan with o vertices. The matrices $\mathbf{M} \in \mathbb{R}^{c\times p}$, $\mathbf{N} \in \mathbb{R}^{c\times o}$ are selection matrices that select vertices in $\mathbf{x}$ that correspond to vertices in $\mathbf{t}$, where c is the number of such correspondences. We then set up the following optimization problem to solve for the $\mathbf{A}$ matrix:

$$\min_{\mathbf{x},\mathbf{b},\mathbf{A}} \quad E_{target}(\mathbf{x}, \mathbf{t}) + E_{reg}(\mathbf{x}, \mathbf{A}) \quad (4a)$$

$$\text{s.t.} \quad \nabla_{\mathbf{x}} E_{act}(\mathbf{x}, \mathbf{A}) + \nabla_{\mathbf{x}} E_{ext}(\mathbf{x}) = 0 \quad (4b)$$

$$c(\mathbf{x}, \mathbf{b}) = 0 \quad (4c)$$

where constraint 4b requires quasi-static equilibrium and constraint 4c is the same physiological constraint as in Eq. 2. Since

$p >> o$, the problem is highly overparameterized and regularization $E_{reg}(\mathbf{x}, \mathbf{A})$ is necessary as we will discuss in Section 3.6. Solving this constrained minimization problem will produce a matrix $\mathbf{A}$ that, when used to solve Eq. 2, will produce an $\mathbf{x}$ as close as possible to the target $\mathbf{t}$.

3.4. Muscle Geometry Matrix S

The key difference in our method is the parameterization of $\mathbf{A}$. Previous work parameterized the $\mathbf{A}$ matrix by allowing nine [6] or six [7] degrees of freedom for every tetrahedron and constructing a matrix from those degrees of freedom. This gives the model great flexibility and allows it to explain almost any given target at the cost of losing the physiological meaning of the matrix $\mathbf{A}$: each tetrahedron acts as its own independent "muscle" and each "muscle" can be transformed in any symmetric way. This is far from the real-world situation, where facial expressions are created using only a few muscles.

To reduce the parameter space, first we will change the semantic meaning of $\mathbf{A}$: instead of having an $\mathbf{A}_i \in \mathbb{R}^{3\times3}$ matrix per tetrahedron, we will have an $\mathbf{A}_i \in \mathbb{R}^{3\times3}$ matrix per facial muscle. Since $\mathbf{A}_i$ is a symmetric matrix constructed from 6 variables, we will instead store it as a 6×1 vector $\mathbf{a}_i$. The global activation vector is $\mathbf{a} \in \mathbb{R}^{6m}$, where m is the number of muscles.

These activations are now independent of the tetrahedra. Intuitively, we want tetrahedra to be influenced by nearby muscles. Since facial muscles are close to each other and even overlap, tetrahedra need to be able to combine nearby activations. We introduce a per-tetrahedron vector $\mathbf{s}_i \in \mathbb{R}^m$, where every value s_j represents a weight for how much this tetrahedron is influenced by muscle j. A tetrahedron's full activation $\mathbf{a}_i$ is then composed of

$$\mathbf{a}_i = \sum_j s_{ij}\mathbf{a}_j \tag{5}$$

where s_{ij} is tetrahedron i's weight for muscle j and $\mathbf{a}_j$ is the activation for this same muscle. This makes a tetrahedron's individual activation a linear combination of the muscle activations. In order for this weighted sum to be sensible, we must impose the following constraints on $\mathbf{s}_i$: $0 \le s_{ij} \le 1$, and $\sum_j^m s_{ij} = 1$. Furthermore, we will also enforce spatial sparsity on this vector: we only allow muscles that are within 1cm of this tetrahedron to take non-zero values. This prevents faraway muscles from influencing a tetrahedron (i.e., a muscle on the right side of the face can not directly influence a tetrahedron located on the left side). Finally, using the same operator $\mathcal{S}$ from Eq. 1, we can express our active muscle model as:

$$E_{act}(\mathbf{x}, \mathbf{A}) = \sum_{i=1}^{n} \min_{\mathbf{R}_i \in \mathrm{SO3}} \frac{\mu_i}{v_i} \|\mathbf{F}_i - \mathbf{R}_i \mathcal{S}\left(\sum_j^m s_{ij}\mathbf{a}_j\right)\|_{\mathbf{F}}^2 + \frac{\lambda_i}{2v_i}\mathrm{Tr}^2(\mathbf{P} - \mathbf{I}) \tag{6}$$

where i indexes tetrahedra and j indexes muscles (s_{ij} denotes coefficient j of vector $\mathbf{s}_i$). What we have accomplished here is moving many of the degrees of freedom into a blending matrix $\mathbf{S} \in \mathbb{R}^{n\times m}$ (where n is the number of tetrahedra and m is the number of muscles), which encodes the geometric blending of the muscles. For a single target, this does not accomplish our goal of reducing the degrees of freedom. However, this $\mathbf{S}$ matrix is a *subject-specific* parameter, akin to the stiffnesses λ and μ, in contrast to an *expression-specific* parameter. Once an $\mathbf{S}$ matrix for a particular subject is found, it can be re-used for all expressions generated by the same subject. For new expressions, the only expression-specific parameters are the positions $\mathbf{x}$ and the muscle activations $\mathbf{a}$.

Passive Tissue. To handle passive tissues, we need to make only a small modification to the $\mathbf{S}$ and $\mathbf{a}$ variables. Passive tissue can be modeled by ensuring:

$$\mathcal{S}\left(\sum_j^m s_{ij}\mathbf{a}_j\right) = \mathbf{I}_3 \tag{7}$$

If we append a constant identity vector $\mathbf{a}_\mathbf{I} = [100101]^T$ to the beginning of the global activation vector $\mathbf{a}$ and add a column to the beginning of $\mathbf{S}$ representing the weight corresponding to passive tissue, then a fully passive tetrahedron would have a value of 1 for this weight (and, respecting the constraints, a value of 0 everywhere else) in its corresponding row in the $\mathbf{S}$ matrix. This also allows our model to contain tetrahedra that are partially muscle and partially passive tissue. This expands the dimensions of our $\mathbf{S}$ matrix to $\mathbf{S} \in \mathbb{R}^{n\times(m+1)}$ and gives our model the ability to include in a single tetrahedron multiple muscles as well as passive tissue, similar to what a volume slice in the face would contain.

3.5. Multi-target fitting

In order to properly optimize for a geometry matrix $\mathbf{S}$, we need to use several expressions simultaneously to prevent over-fitting. Each expression has its own $\mathbf{x}$ and $\mathbf{a}$ variables, but all expressions share the same $\mathbf{S}$ variables. This leads to our multi-target inverse optimization problem:

$$\min_{\mathbf{x},\mathbf{b},\mathbf{a},\mathbf{S}} \quad \sum_k^q E_{target}(\mathbf{x}_k, \mathbf{t}_k) + E_{reg}(\mathbf{a}_k, \mathbf{S}) \tag{8a}$$

$$\text{s.t.} \quad \nabla_\mathbf{x} E_{act}(\mathbf{x}_0, \mathbf{a}_0, \mathbf{S}) + \nabla_\mathbf{x} E_{ext}(\mathbf{x}_0) = 0 \tag{8b}$$

$$c(\mathbf{x}_0, \mathbf{b}_0) = 0 \tag{8c}$$

$$\vdots$$

$$\nabla_\mathbf{x} E_{act}(\mathbf{x}_q, \mathbf{a}_q, \mathbf{S}) + \nabla_\mathbf{x} E_{ext}(\mathbf{x}_q) = 0 \tag{8d}$$

$$c(\mathbf{x}_q, \mathbf{b}_q) = 0 \tag{8e}$$

$$0 \le \mathbf{S} \le 1 \tag{8f}$$

$$\mathbf{S}\mathbf{1}_n = \mathbf{1}_m \tag{8g}$$

where q is the number of targets, p is the number of vertices per target, m is the number of muscles, $\mathbf{x} \in \mathbb{R}^{3p\times q}$ represents the positions, $\mathbf{a} \in \mathbb{R}^{6m\times q}$ represents the activations with the different columns representing different targets (i.e., $\mathbf{x}_k, \mathbf{a}_k$ are the positions and activations for target k), and $\mathbf{S} \in \mathbb{R}^{n\times m}$ is the geometry blending matrix described above (recall that n is the number of tetrahedra). The objective term (Eq. 8a) contains all of the target energies and the regularization terms (discussed in Section 3.6). The constraints in Eq. 8b and Eq. 8c are the steady-state and physiological constraints (including jaw kinematics)

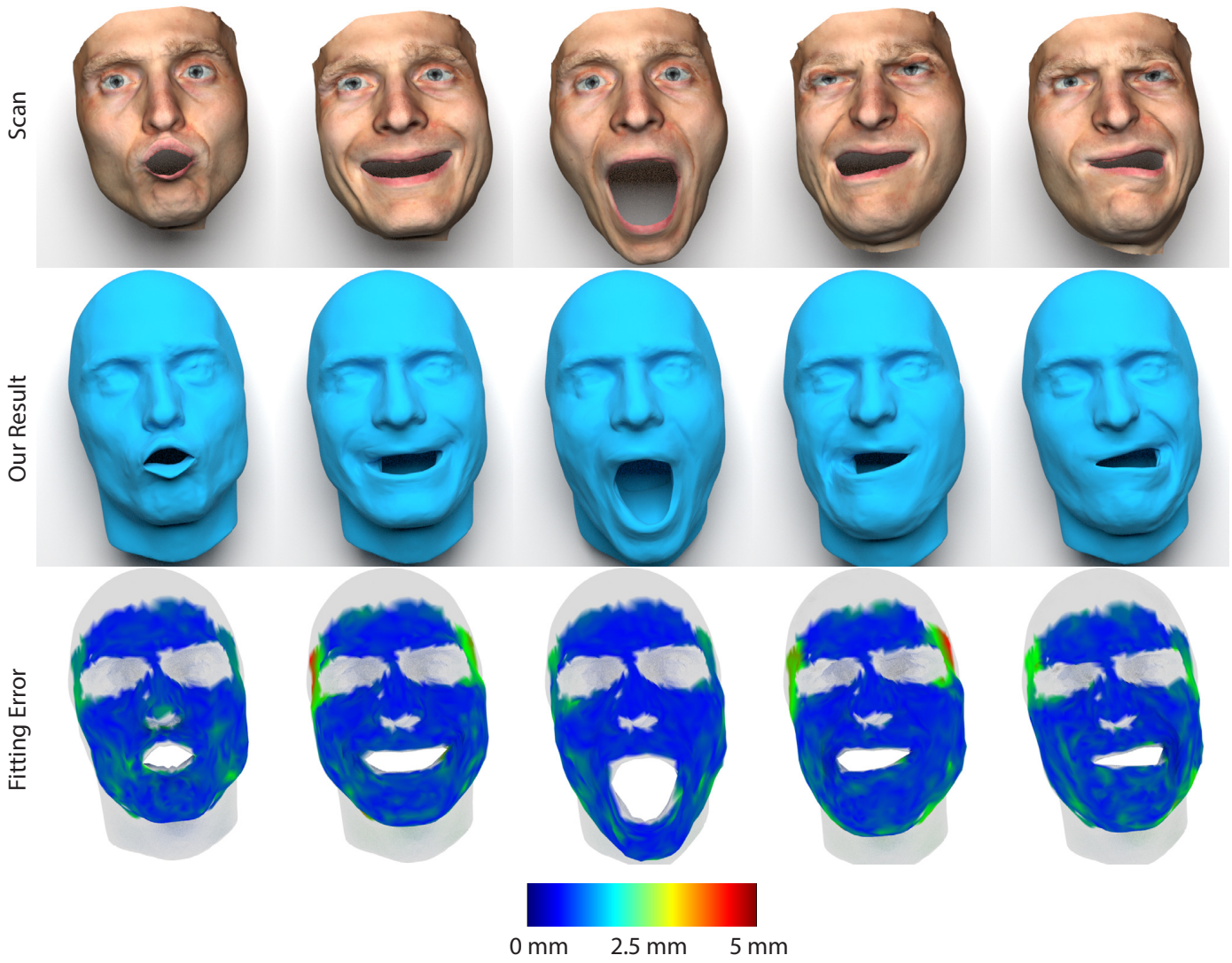

Fig. 2. The input scans used for the multi-target fitting (top), the mesh produced by our method (middle), and a visualization of the point-to-point fitting error (bottom).

for target 0, and each target has such a pair of constraints. Finally, the **S**-matrix constraints discussed in Section 3.4 are represented in Eq. 8f and Eq. 8g, where $\mathbf{1}_n \in \mathbb{R}^n$ and $\mathbf{1}_m \in \mathbb{R}^m$ represents an all-ones vectors.

3.6. Regularization

Eq. 8 is an under-determined problem due to the fact that the target scans and constraints exclude many of the internal vertices. To help solve the optimization problem, we use simple regularizations on the **A** and **S** matrices that push them towards their initial values $\mathbf{A}_i, \mathbf{S}_i$:

$$\frac{w_a}{2}\|\mathbf{A} - \mathbf{A}_i\|_{\mathbf{F}}^2 + \frac{w_s}{2}\|\mathbf{S} - \mathbf{S}_i\|_{\mathbf{F}}^2 \tag{9}$$

The weights w_a, w_s represent the strength of this regularization and are discussed more in Section 3.7. Additionally, we want the activations to be volume-preserving, i.e. having a determinant of one. We add the following regularization to try to enforce this property:

$$\frac{w_v}{2}\sum_{j=1}^{m}(det\left(\mathcal{S}(\mathbf{a}_j)\right) - 1)^2 \tag{10}$$

where $\mathbf{a}_j \in R^6$ represents the block from **a** corresponding to muscle j. Adding Eq. 9 and Eq. 10 gives us the E_{reg} term in Eq. 8.

3.7. Numerical Solution

We solve this optimization problem in Eq. 8 using the IPOPT [75] package and employing a block coordinate descent strategy, alternating between solves for $\mathbf{x}, \mathbf{A}$ and $\mathbf{x}, \mathbf{S}$. Additionally, we use high regularization weights w_a, w_s in Eq. 9 at the beginning and decrease them in subsequent solves. In our results, the **S** matrix was obtained by performing two or three alternating **A** and **S** iterations before decreasing the weights.

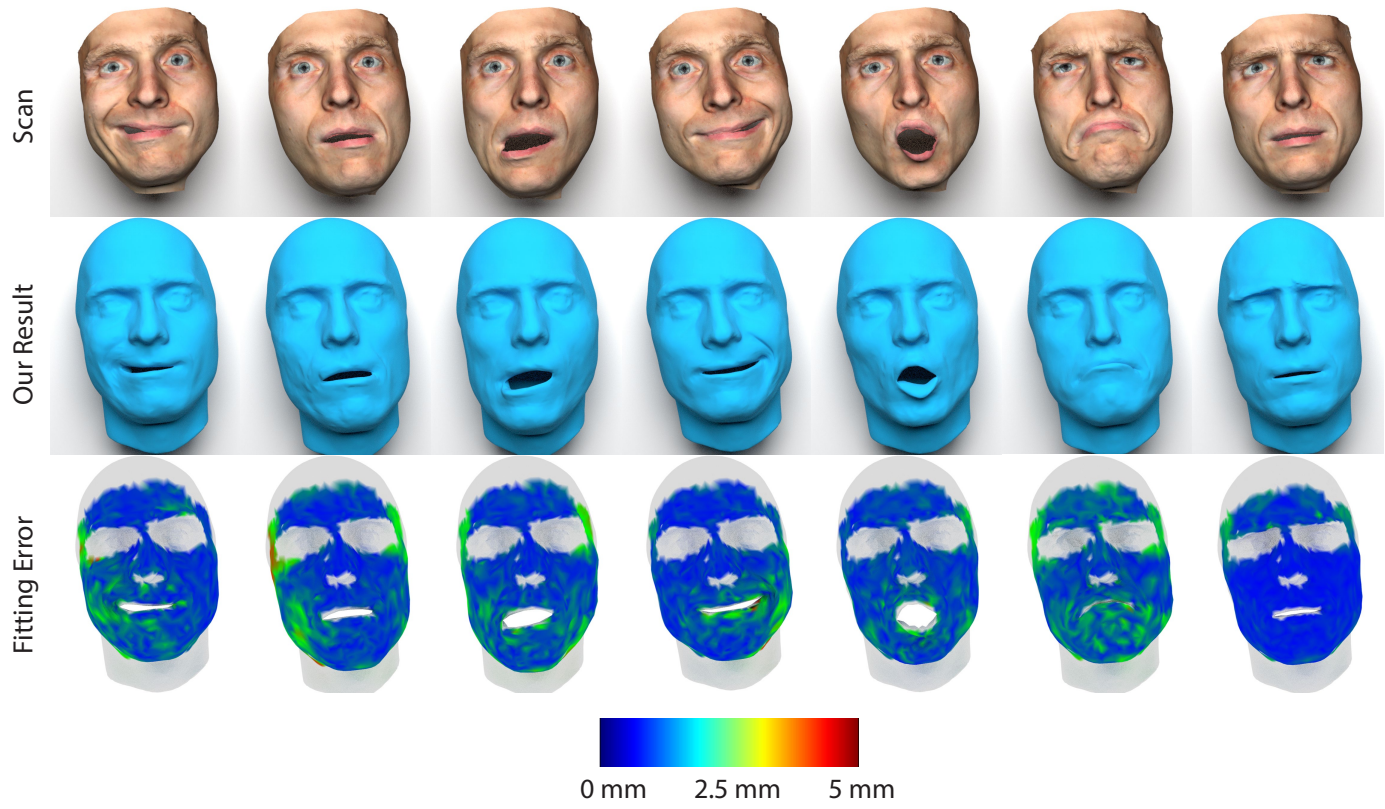

Fig. 3. A set of previously unseen scans scans (top), the mesh produced by our method using only an activation solve (middle), and a visualization of the point-to-point fitting error (bottom).

Once an **S** matrix is obtained for a subject, we can then solve for new expressions using only the **A** solve. For this solve, we do not need to use the strategy of starting with a high regularization weight w_a and then iteratively decrease it; we can use a low weight from the start. This combined with the low dimensionality of the unknown activations makes this solve much faster.

4. Results

4.1. Evaluation

To optimize for the **S** matrix, we used five scans to solve the multi-target optimization problem in Eq. 8, shown in Figure 2 (top). The resulting meshes are shown in Figure 2 (middle) and the fitting error is visualized in Figure 2 (bottom) as the point-to-point distance between our mesh vertices and the corresponding vertices in the scans. We alternated between solving for the global **S** matrix and the target-specific $\mathbf{A}_i$ matrices as described in Section 3.7. Our method is able to fit the target scans quite well, with an average fitting error of 0.85 mm.

After this **S** optimization phase, we can now use the **S** to solve for only the muscle activations **A** for new expressions, which were not included in the multi-target fitting optimization. We tried fitting new scans, shown in Figure 3 (top) by solving only for the activations **A**. Figure 3 (middle) shows the resulting mesh while Figure 3 (bottom) shows the point-to-point error of the fitting. We can see that we are able to fit these expressions, with an average error of 0.90 mm.

To test the generalizability of our method, we also performed the same experiments on another subject. As before, we performed the steps outlined in Section 3.1 using the same parameters as the previous subject (e.g., the 1 cm search radius). In Figure 4 (a), we used 3 scans as targets for the multi-fitting stage and show the result and the fitting error. In Figure 4 (b), we re-used the **S** matrix we solved for to fit new expressions by solving for the activations only.

Instead of doing the multi-target optimization for **S**, we can just use the initial values and only solve for the activations. This results in significant artifacts, shown in Figure 5, caused by an inability of the underlying reduced-space activations to replicate the target expression, sometimes resulting in significant tetrahedral inversions. Our method with the **S** optimization produces a much better result, showing that we cannot just reduce the space of the muscle activations without accounting for the geometry.

Introducing the **S** matrix does not affect the ability of our activations to be used as volumetric blendshapes [6]. Figure 6 shows an animation where we use two expressions as keyframes and linearly interpolate the activations and jaw kinematics of each expression for the in-between frames. Please refer to the accompanying video for a more clear demonstration.

Since we are using a physics-based model, we can also add in physical effects. We can change the physical properties of our

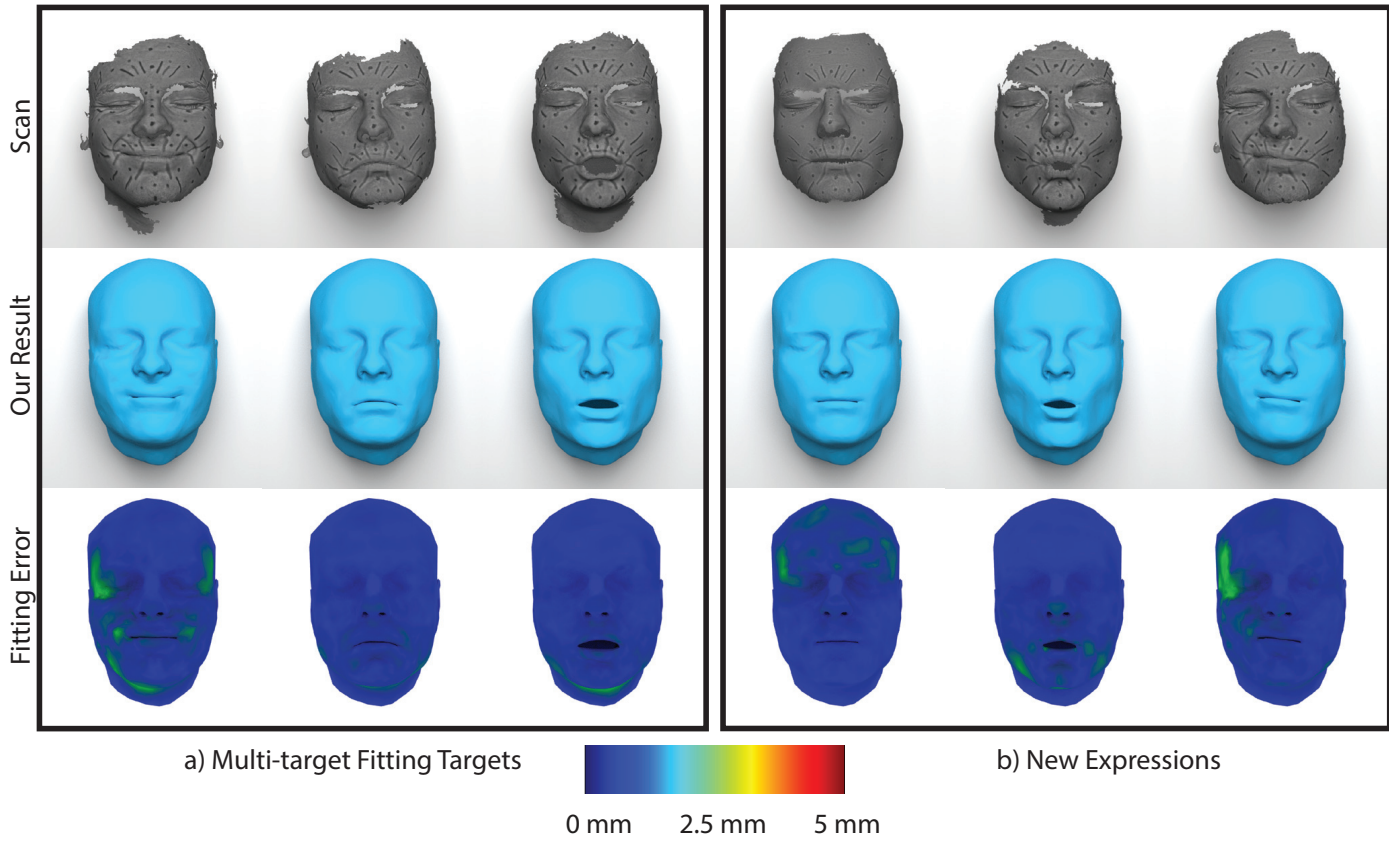

Fig. 4. Results on a subject different from Figure 2 and Figure 3. a) The three expressions that were used to fit the S matrix. b) The resulting fitting on three new expressions.

Fig. 5. Optimizing for only the activations without accounting for the muscle geometry produces artifacts and results in a poor fitting of the input scan.

subject: in Figure 8 we increase the "fatness" of our subject. We accomplish this in a similar vein to the work of [7]: we model fat as a plastic deformation, augmenting $\mathbf{F}(\mathbf{x}, \mathbf{x}_{rest})$ in Eq. 1 with $\mathbf{F}(\mathbf{x}, \mathbf{x}_{rest})\mathbf{F}_p^{-1}$, where $\mathbf{F}_p = f_i\mathbf{I}$ is a scaled identity matrix, with $f_i > 0$ representing the fat multiplier for this tetrahedron. Unlike [7], we do not require a manually made "fat map" as we have optimized for the active/passive ratio during our $\mathbf{S}$ optimization. We choose scalar values f_{min}, f_{max} indicating how much "fatter" ($f_{max} > 1$, $f_{min} = 1$) or "thinner" ($0 < f_{min} < 1, f_{max} = 1$) we want our subject. Figure 8 uses a range of $(0.9, 1.0)$ to $(1.0, 1.2)$ for (f_{min}, f_{max}). For any given tetrahedron, the first weight in its $\mathbf{s}_i$ vector is the ratio of passive tissue, so we can use this weight to grow or reduce fat as desired: $f_i = f_{min}(1 - s_{i0}) + f_{max}s_{i0}$.

Our model is capable of incorporating external forces: in Figure 7 we added a wind force to our keyframed animations in the $+\mathbf{x}$ direction, causing the flesh to deform and jiggle while still making the correct expressions.

4.2. Comparison to Previous Work

Ichim et al. [6] introduced a muscle model that enabled fitting to target scans by allowing every muscle tetrahedron to move independently. While this makes the fitting very close to the target scans, as shown in Figure 9, it does so by constructing a model that intentionally *overfits* to the scan. One drawback of this approach is that it allows non-physical deformations that can arise from artifacts in the capture process. To illustrate this point, we applied some non-physical deformation to one of our scans and test both the Phace method [7] and our method, shown in Figure 10. We can see that our method is not able to fit this deformity due to our more restricted muscle model. Phace [7] is able to (incorrectly) fit this scan artifact. While this scenario is admittedly exaggerated, it illustrates the drawbacks of overfitting to the data. An ideal muscle model is capable of producing *only* the various expressions produced by the facial muscles and is not able to explain non-physical input data.

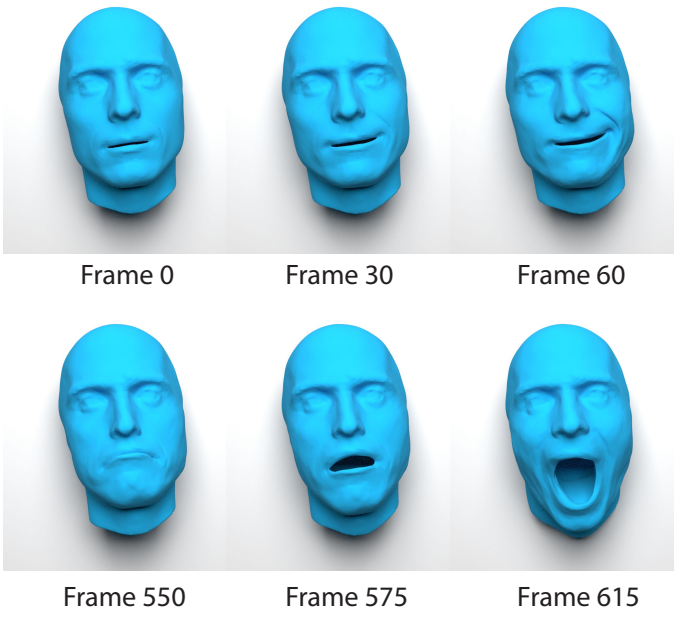

Fig. 6. An animation may be produced by linearly interpolating the activations of two expressions.

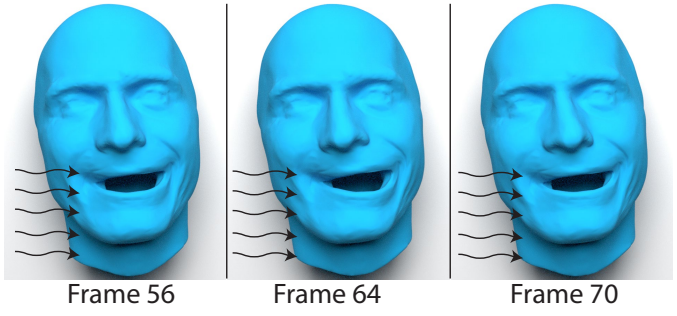

Fig. 7. Applying external forces such as wind causes the flesh to deform accordingly.

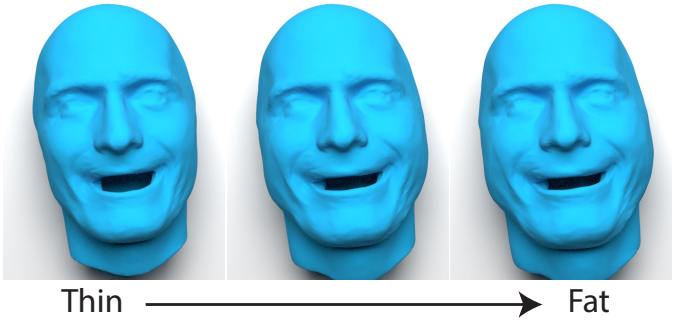

Fig. 8. We can use our model to simulate the accumulation of fat on our subject. Fat is only accumulated in passive tissue, not in muscle.

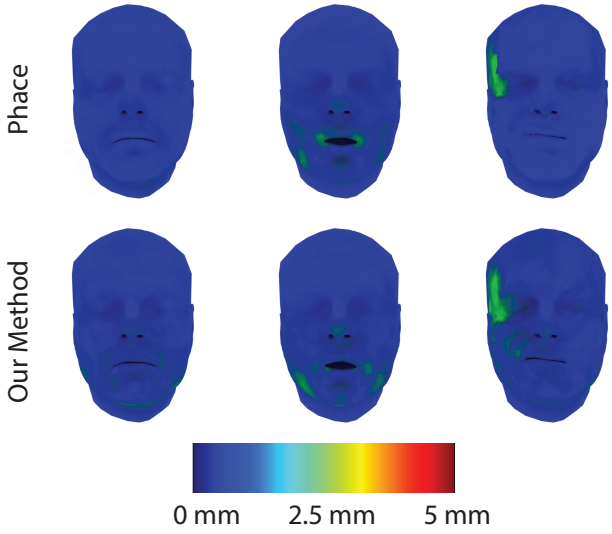

Fig. 9. The fitting error of [7] (top) and our method (bottom). Our method comes close to the fitting of [7], which provides a slightly better fitting at the cost of overfitting (described in Figure 10)

Another benefit of our reduced parameter space for the activations is that we are able to achieve significant speedup when solving for new muscle activations when compared with Phace [7]. This is important because ultimately these activations are used in a volumetric blendshape system, which can require many different expressions to be solved for. For Figure 10 the activation solve for Phace [7] (top) took 5 iterations and 734 ms, (bottom) took 7 iterations and 1100 ms. Our activation solve (top) took 10 iterations and 188 ms, (bottom) took 8 iterations and 148 ms with a mesh resolution of 7,046 vertices and 25,227 tetrahedra on an Intel Core i7-8750 processor. For all of the expressions in Figure 2 and Figure 3, our activation solve took an average of 8.58 iterations and 157.42 ms, while Phace [7] took an average of 6.5 iterations and 998.08 ms.

5. Limitations and Future Work

While we have reduced the parameter space compared to previous work down to more closely match real-world muscle, each muscle still has a 6-DoF activation model. Real muscles can only contract along their fiber directions; an ideal muscle model will only have 1-DoF activation per muscle. This prevents our method from producing readily-controllable muscles that behave in an intuitive way. An interesting future direction is to try and solve for the fiber directions to even further reduce the parameter space towards more realistic muscles as well as introduce anisotropy to the muscle model.

Another limitation of this work is that we use a homogeneous and isotropic material for both our muscle and passive tissue model. In the human body fat and muscle have different material properties. In particular, muscles are anisotropic and have different stiffness properties when they are activated compared to when they are passive. Modeling this difference between active and passive material parameters is a good direction for future work, as well as developing more biomechanically accurate deformation models for both the muscle and the passive tissue.

Our current face model does not contain an eye model or eyelids, so it is incapable of expressing dynamic eye motions, such as the closing of the eyelid or the rotations of the eyeball. The motions of the eye and eyelids are difficult to capture and model, and require special considerations [30]. A more advanced physics-based model that contains complex eye controls is a

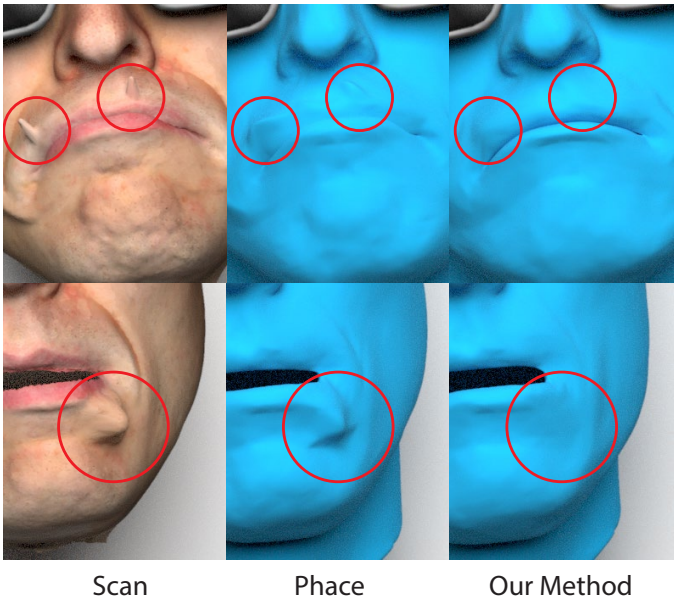

Fig. 10. Previous work allows for fitting of non-physical shapes. We add some artifacts to a scan (left) and demonstrate how the method of [7] is able to fit this (middle). Our muscle model is correctly unable to fit this artifact (right).

promising avenue for future work.

6. Conclusions

In this work, we take an important step towards producing a fully automated, personalized physics-based facial model for a subject. Our main contribution is in reducing the parameter space of the muscle activations in the frameworks of [6, 7] by solving for the blending of muscles and passive tissue for a specific subject. This is important since muscles in the face are close together and their interaction with each other greatly influences the surface deformation that we see on the skin. We believe that good muscle models are important in both the entertainment and medical fields, and physics-based facial models provide many advantages over traditional geometry-based approaches. We hope that this work pushes forward the capabilities of anatomical facial muscle modeling.

Acknowledgements

We thank Dinesh Pai for many insightful discussions. We also thank Thabo Beeler and Derek Bradly for generously providing us with scan data. This material is based upon work supported by the National Science Foundation under Grant Numbers IIS-1617172 and IIS-1622360. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Beeler, T, Bickel, B, Beardsley, P, Sumner, B, Gross, M. High-quality single-shot capture of facial geometry. In: ACM Transactions on Graphics (TOG); vol. 29. ACM; 2010, p. 40.

[2] Beeler, T, Hahn, F, Bradley, D, Bickel, B, Beardsley, P, Gotsman, C, et al. High-quality passive facial performance capture using anchor frames. In: ACM Transactions on Graphics (TOG); vol. 30. ACM; 2011, p. 75.

[3] Beeler, T, Bradley, D. Rigid stabilization of facial expressions. In: ACM Transactions on Graphics (TOG); vol. 33. ACM; 2014, p. 44.

[4] Lewis, JP, Anjyo, K, Rhee, T, Zhang, M, Pighin, FH, Deng, Z. Practice and theory of blendshape facial models. In: Eurographics (State of the Art Reports); vol. 1. 2014, p. 2.

[5] Sifakis, E, Neverov, I, Fedkiw, R. Automatic determination of facial muscle activations from sparse motion capture marker data. In: ACM SIGGRAPH 2005 Papers. 2005, p. 417–425.

[6] Ichim, AE, Kavan, L, Nimier-David, M, Pauly, M. Building and animating user-specific volumetric face rigs. In: SIGGRAPH/Eurographics symposium on Computer animation. 2016, p. 107–117.

[7] Ichim, AE, Kadleček, P, Kavan, L, Pauly, M. Phace: physics-based face modeling and animation. In: ACM Transactions on Graphics (TOG); vol. 36. ACM; 2017, p. 153.

[8] 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. ACM Press/Addison-Wesley Publishing Co.; 1999, p. 187–194.

[9] Terzopoulos, D, Waters, K. Physically-based facial modelling, analysis, and animation. In: The journal of visualization and computer animation; vol. 1. Wiley Online Library; 1990, p. 73–80.

[10] Ekman, P, Rosenberg, EL. What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS). Oxford University Press, USA; 1997.

[11] Ichim, AE, Bouaziz, S, Pauly, M. Dynamic 3d avatar creation from hand-held video input. In: ACM Transactions on Graphics (TOG); vol. 34. ACM New York, NY, USA; 2015, p. 1–14.

[12] Ribera, RBi, Zell, E, Lewis, J, Noh, J, Botsch, M. Facial retargeting with automatic range of motion alignment. In: ACM Transactions on Graphics (TOG); vol. 36. ACM New York, NY, USA; 2017, p. 1–12.

[13] Yoon, SH, Lewis, J, Rhee, T. Blending face details: Synthesizing a face using multiscale face models. In: IEEE computer graphics and applications; vol. 37. IEEE; 2017, p. 65–75.

[14] Li, T, Bolkart, T, Black, MJ, Li, H, Romero, J. Learning a model of facial shape and expression from 4d scans. In: ACM Transactions on Graphics (TOG); vol. 36. ACM; 2017, p. 194.

[15] Ma, WC, Wang, YH, Fyffe, G, Chen, BY, Debevec, P. A blendshape model that incorporates physical interaction. In: Computer Animation and Virtual Worlds; vol. 23. Wiley Online Library; 2012, p. 235–243.

[16] Barrielle, V, Stoiber, N, Cagniart, C. Blendforces: A dynamic framework for facial animation. In: Computer Graphics Forum; vol. 35. Wiley Online Library; 2016, p. 341–352.

[17] Kozlov, Y, Bradley, D, Bächer, M, Thomaszewski, B, Beeler, T, Gross, M. Enriching facial blendshape rigs with physical simulation. In: Computer Graphics Forum; vol. 36. Wiley Online Library; 2017, p. 75–84.

[18] Kim, M, Pons-Moll, G, Pujades, S, Bang, S, Kim, J, Black, MJ, et al. Data-driven physics for human soft tissue animation. In: ACM Transactions on Graphics (TOG); vol. 36. ACM New York, NY, USA; 2017, p. 1–12.

[19] Bhat, KS, Goldenthal, R, Ye, Y, Mallet, R, Koperwas, M. High fidelity facial animation capture and retargeting with contours. In: Proceedings of the 12th ACM SIGGRAPH/eurographics symposium on computer animation. ACM; 2013, p. 7–14.

[20] Bradley, D, Heidrich, W, Popa, T, Sheffer, A. High resolution passive facial performance capture. In: ACM transactions on graphics (TOG); vol. 29. ACM; 2010, p. 41.

[21] Fyffe, G, Jones, A, Alexander, O, Ichikari, R, Debevec, P. Driving high-resolution facial scans with video performance capture. In: ACM Transactions on Graphics (TOG); vol. 34. ACM; 2014, p. 8.

[22] Laine, S, Karras, T, Aila, T, Herva, A, Saito, S, Yu, R, et al. Production-level facial performance capture using deep convolutional neural networks. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. ACM; 2017, p. 10.

[23] von der Pahlen, J, Jimenez, J, Danvoye, E, Debevec, P, Fyffe, G, Alexander, O. Digital ira and beyond: creating real-time photoreal digital actors. In: ACM SIGGRAPH 2014 Courses. 2014, p. 1–384.

[24] Alexander, O, Rogers, M, Lambeth, W, Chiang, JY, Ma, WC, Wang, CC, et al. The digital emily project: Achieving a photorealistic digital actor. In: IEEE Computer Graphics and Applications; vol. 30. IEEE; 2010,

p. 20–31.
[25] Ghosh, A, Fyffe, G, Tunwattanapong, B, Busch, J, Yu, X, Debevec, P. Multiview face capture using polarized spherical gradient illumination. In: ACM Transactions on Graphics (TOG); vol. 30. ACM; 2011, p. 129.
[26] Shi, F, Wu, HT, Tong, X, Chai, J. Automatic acquisition of high-fidelity facial performances using monocular videos. In: ACM Transactions on Graphics (TOG); vol. 33. ACM; 2014, p. 222.
[27] Cao, C, Bradley, D, Zhou, K, Beeler, T. Real-time high-fidelity facial performance capture. In: ACM Transactions on Graphics (TOG); vol. 34. ACM; 2015, p. 46.
[28] Bermano, A, Beeler, T, Kozlov, Y, Bradley, D, Bickel, B, Gross, M. Detailed spatio-temporal reconstruction of eyelids. In: ACM Transactions on Graphics (TOG); vol. 34. ACM; 2015, p. 44.
[29] Bérard, P, Bradley, D, Nitti, M, Beeler, T, Gross, M. High-quality capture of eyes. In: ACM Transactions on Graphics (TOG); vol. 33. ACM New York, NY, USA; 2014, p. 1–12.
[30] Bérard, P, Bradley, D, Gross, M, Beeler, T. Lightweight eye capture using a parametric model. In: ACM Transactions on Graphics (TOG); vol. 35. ACM; 2016, p. 117.
[31] Beeler, T, Bickel, B, Noris, G, Beardsley, P, Marschner, S, Sumner, RW, et al. Coupled 3d reconstruction of sparse facial hair and skin. In: ACM Transactions on Graphics (TOG); vol. 31. ACM; 2012, p. 117.
[32] Luo, L, Li, H, Paris, S, Weise, T, Pauly, M, Rusinkiewicz, S. Multi-view hair capture using orientation fields. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE; 2012, p. 1490–1497.
[33] Hu, L, Ma, C, Luo, L, Li, H. Single-view hair modeling using a hairstyle database. In: ACM Transactions on Graphics (TOG); vol. 34. ACM; 2015, p. 125.
[34] Wu, C, Bradley, D, Garrido, P, Zollhöfer, M, Theobalt, C, Gross, M, et al. Model-based teeth reconstruction. In: ACM Transactions on Graphics (TOG); vol. 35. ACM; 2016, p. 220.
[35] Tian, Yl, Kanade, T, Cohn, J. Robust lip tracking by combining shape, color and motion. In: Proceedings of the 4th Asian Conference on Computer Vision. 2000, p. 1040–1045.
[36] Garrido, P, Zollhöfer, M, Wu, C, Bradley, D, Pérez, P, Beeler, T, et al. Corrective 3d reconstruction of lips from monocular video. In: ACM Transactions on Graphics (TOG); vol. 35. ACM; 2016, p. 219.
[37] Dinev, D, Beeler, T, Bradley, D, Bächer, M, Xu, H, Kavan, L. User-guided lip correction for facial performance capture. In: Computer Graphics Forum; vol. 37. Wiley Online Library; 2018, p. 93–101.
[38] Nagano, K, Fyffe, G, Alexander, O, Barbič, J, Li, H, Ghosh, A, et al. Skin microstructure deformation with displacement map convolution. In: ACM Transactions on Graphics (TOG); vol. 34. ACM New York, NY, USA; 2015, p. 1–10.
[39] Graham, P, Tunwattanapong, B, Busch, J, Yu, X, Jones, A, Debevec, P, et al. Measurement-based synthesis of facial microgeometry. In: Computer Graphics Forum; vol. 32. Wiley Online Library; 2013, p. 335–344.
[40] Zoss, G, Bradley, D, Bérard, P, Beeler, T. An empirical rig for jaw animation. In: ACM Transactions on Graphics (TOG); vol. 37. ACM New York, NY, USA; 2018, p. 1–12.
[41] Yang, W, Marshak, N, Sỳkora, D, Ramalingam, S, Kavan, L. Building anatomically realistic jaw kinematics model from data. In: The Visual Computer; vol. 35. Springer; 2019, p. 1105–1118.
[42] Wu, C, Bradley, D, Gross, M, Beeler, T. An anatomically-constrained local deformation model for monocular face capture. In: ACM Transactions on Graphics (TOG); vol. 35. ACM; 2016, p. 115.
[43] Lloyd, JE, Stavness, I, Fels, S. Artisynth: A fast interactive biomechanical modeling toolkit combining multibody and finite element simulation. In: Soft tissue biomechanical modeling for computer assisted surgery. Springer; 2012, p. 355–394.
[44] Faure, F, Duriez, C, Delingette, H, Allard, J, Gilles, B, Marchesseau, S, et al. SOFA: A Multi-Model Framework for Interactive Physical Simulation. In: Soft Tissue Biomechanical Modeling for Computer Assisted Surgery; vol. 11. Springer; 2012, p. 283–321.
[45] Maas, SA, Ellis, BJ, Ateshian, GA, Weiss, JA. Febio: finite elements for biomechanics. In: booktitle of biomechanical engineering; vol. 134. American Society of Mechanical Engineers Digital Collection; 2012,.
[46] Börgesson, L. Abaqus. In: Developments in geotechnical engineering; vol. 79. Elsevier; 1996, p. 565–570.
[47] Tran, H, Charleux, F, Rachik, M, Ehrlacher, A, Ho Ba Tho, M. In vivo characterization of the mechanical properties of human skin derived from mri and indentation techniques. In: Computer methods in biomechanics and biomedical engineering; vol. 10. Taylor & Francis; 2007, p. 401–407.
[48] Weiss, JA, Maker, BN, Govindjee, S. Finite element implementation of incompressible, transversely isotropic hyperelasticity. In: Computer methods in applied mechanics and engineering; vol. 135. Elsevier; 1996, p. 107–128.
[49] Nazari, MA, Perrier, P, Chabanas, M, Payan, Y. Simulation of dynamic orofacial movements using a constitutive law varying with muscle activation. In: Computer methods in biomechanics and biomedical engineering; vol. 13. Taylor & Francis; 2010, p. 469–482.
[50] Pai, DK, Doel, Kvd, James, DL, Lang, J, Lloyd, JE, Richmond, JL, et al. Scanning physical interaction behavior of 3d objects. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. 2001, p. 87–96.
[51] Pai, DK, Rothwell, A, Wyder-Hodge, P, Wick, A, Fan, Y, Larionov, E, et al. The human touch: measuring contact with real human soft tissues. vol. 37. ACM New York, NY, USA; 2018, p. 1–12.
[52] Bickel, B, Bächer, M, Otaduy, MA, Matusik, W, Pfister, H, Gross, M. Capture and modeling of non-linear heterogeneous soft tissue. vol. 28. ACM New York, NY, USA; 2009, p. 1–9.
[53] Bickel, B, Kaufmann, P, Skouras, M, Thomaszewski, B, Bradley, D, Beeler, T, et al. Physical face cloning. vol. 31. ACM New York, NY, USA; 2012, p. 1–10.
[54] Eskes, M, Balm, AJ, van Alphen, MJ, Smeele, LE, Stavness, I, van der Heijden, F. Simulation of facial expressions using person-specific semg signals controlling a biomechanical face model. vol. 13. Springer; 2018, p. 47–59.
[55] Nedel, LP, Thalmann, D. Real time muscle deformations using mass-spring systems. In: Computer Graphics International, 1998. Proceedings. IEEE; 1998, p. 156–165.
[56] Teran, J, Blemker, S, Hing, V, Fedkiw, R. Finite volume methods for the simulation of skeletal muscle. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association; 2003, p. 68–74.
[57] 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. Eurographics Association; 2005, p. 181–190.
[58] Teran, J, Sifakis, E, Blemker, SS, Ng-Thow-Hing, V, Lau, C, Fedkiw, R. Creating and simulating skeletal muscle from the visible human data set. In: IEEE Transactions on Visualization and Computer Graphics; vol. 11. IEEE; 2005, p. 317–328.
[59] Lee, SH, Sifakis, E, Terzopoulos, D. Comprehensive biomechanical modeling and simulation of the upper body. In: ACM Transactions on Graphics (TOG); vol. 28. ACM; 2009, p. 99.
[60] Fan, Y, Litven, J, Levin, DI, Pai, DK. Eulerian-on-lagrangian simulation. In: ACM Transactions on Graphics (TOG); vol. 32. ACM New York, NY, USA; 2013, p. 1–9.
[61] Fan, Y, Litven, J, Pai, DK. Active volumetric musculoskeletal systems. In: ACM Transactions on Graphics (TOG); vol. 33. ACM New York, NY, USA; 2014, p. 1–9.
[62] Bouaziz, S, Martin, S, Liu, T, Kavan, L, Pauly, M. Projective dynamics: fusing constraint projections for fast simulation. In: ACM Transactions on Graphics (TOG); vol. 33. ACM New York, NY, USA; 2014, p. 1–11.
[63] Saito, S, Zhou, ZY, Kavan, L. Computational bodybuilding: Anatomically-based modeling of human bodies. In: ACM Transactions on Graphics (TOG); vol. 34. ACM; 2015, p. 41.
[64] Kadleček, P, Ichim, AE, Liu, T, Křivánek, J, Kavan, L. Reconstructing personalized anatomical models for physics-based body animation. In: ACM Transactions on Graphics (TOG); vol. 35. ACM; 2016, p. 213.
[65] Blemker, SS, Pinsky, PM, Delp, SL. A 3d model of muscle reveals the causes of nonuniform strains in the biceps brachii. In: booktitle of biomechanics; vol. 38. Elsevier; 2005, p. 657–665.
[66] Wu, T. A computational framework for modelling the biomechanics of human facial expressions. Ph.D. thesis; The University of Auckland; 2013.
[67] Cong, M, Bhat, KS, Fedkiw, R. Art-directed muscle simulation for high-end facial animation. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 2016, p. 119–127.
[68] Lan, L, Cong, M, Fedkiw, R. Lessons from the evolution of an anatomical facial muscle model. In: Proceedings of the ACM SIGGRAPH Digital Production Symposium. ACM; 2017, p. 11.

[69] Bao, M, Cong, M, Grabli, S, Fedkiw, R. High-quality face capture using anatomical muscles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019, p. 10802–10811.
[70] Stavness, I, Nazari, MA, Flynn, C, Perrier, P, Payan, Y, Lloyd, JE, et al. Coupled biomechanical modeling of the face, jaw, skull, tongue, and hyoid bone. In: 3D multiscale physiological human. Springer; 2014, p. 253–274.
[71] Cong, M, Bao, M, Bhat, KS, Fedkiw, R, et al. Fully automatic generation of anatomical face simulation models. In: Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation. ACM; 2015, p. 175–183.
[72] Kadleček, P, Kavan, L. Building accurate physics-based face models from data. In: Proceedings of the ACM on Computer Graphics and Interactive Techniques; vol. 2. ACM New York, NY, USA; 2019, p. 1–16.
[73] Ali-Hamadi, D, Liu, T, Gilles, B, Kavan, L, Faure, F, Palombi, O, et al. Anatomy transfer. In: ACM Transactions on Graphics (TOG); vol. 32. ACM New York, NY, USA; 2013, p. 1–8.
[74] Si, H. Tetgen, a delaunay-based quality tetrahedral mesh generator. In: ACM Transactions on Mathematical Software (TOMS); vol. 41. ACM New York, NY, USA; 2015, p. 1–36.
[75] Wächter, A, Biegler, LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. In: Mathematical programming; vol. 106. Springer; 2006, p. 25–57.