High-Quality Face Capture Using Anatomical Muscles

Michael Bao[1,2] Matthew Cong[2,†] Stéphane Grabli[2,†] Ronald Fedkiw[1,2]
[1]Stanford University [2]Industrial Light & Magic
[1]{mikebao,rfedkiw}@stanford.edu [†]{mcong,sgrabli}@ilm.com

Abstract

Muscle-based systems have the potential to provide both anatomical accuracy and semantic interpretability as compared to blendshape models; however, a lack of expressivity and differentiability has limited their impact. Thus, we propose modifying a recently developed rather expressive muscle-based system in order to make it fully-differentiable; in fact, our proposed modifications allow this physically robust and anatomically accurate muscle model to conveniently be driven by an underlying blendshape basis. Our formulation is intuitive, natural, as well as monolithically and fully coupled such that one can differentiate the model from end to end, which makes it viable for both optimization and learning-based approaches for a variety of applications. We illustrate this with a number of examples including both shape matching of three-dimensional geometry as as well as the automatic determination of a three-dimensional facial pose from a single two-dimensional RGB image without using markers or depth information.

1. Introduction

Muscle simulation-based animation systems are attractive due to their ability to preserve important physical properties such as volume conservation as well as their ability to handle contact and collision. Moreover, utilizing an anatomically motivated set of controls provides a straightforward way of extracting out semantic meaning from the control values. Unfortunately, even though [42] was able to automatically compute muscle activation values given sparse motion capture data, muscle-based animation models have proven to be significantly less expressive and harder to control than their blendshape-based counterparts [30].

Recently, [14] introduced a novel method that significantly improved upon the expressiveness of muscle-based animation systems. They introduced the concept of "muscle tracks" to control the deformation of the underlying musculature. This concept gives the muscle simulation enough expressiveness to target arbitrary shapes, which allowed it be used in high-quality movie productions such as *Kong: Skull Island* where it was used both to aid in the creation of blendshapes and to offer physically-based corrections to artist-created animation sequences [15, 29]. While [14] alleviates the problems of muscle-based simulation in regards to expressiveness and control, the method is geared towards generative computer graphics problems, and is thus not amenable for estimating a facial pose from a two-dimensional image as is common for markerless performance capture. One could iterate between solving for a performance using blendshapes and then using a muscle-based solution to correct the blendshapes; however, this iterative method is lossy as the muscle simulation does not have access to the raw data and may thus hallucinate details or erase details of the performance.

In this paper, we extend [14] by combining the ease of use and differentiability of traditional blendshape models with expressive, physically-plausible muscle track simulations in order to create a differentiable simulation framework that can be used interchangeably with traditional blendshape models for facial performance capture and animation. Instead of relying on a non-differentiable per-frame volumetric morph to drive the muscle track deformation as in [14], we instead create a state-of-the-art blendshape model for each muscle, which is then used to drive its volumetric deformation. Our model maintains the expressiveness of [14] while preserving crucial physical properties. Furthermore, our new formulation is differentiable from end to end, which allows it to be used to target three-dimensional facial poses as well as two-dimensional RGB images. We demonstrate that our blendshape muscle tracks method shows significant improvements in anatomical plausibility and semantic interpretability when compared to state-of-the-art blendshape-based methods for targeting three-dimensional geometry and two-dimensional RGB images.

2. Related Work

Face Models: Although our work does not directly address the modeling part of the pipeline, it relies on having a pre-existing model of the face. For building a realistic digital double of an actor, multi-view stereo techniques can be used to collect high-quality geometry and texture information in a variety of poses [5, 6, 16]. Artists can then use this data to create the final blendshape model. In state-of-the-art models, the deformation model will include non-linear skinning/enveloping in addition to linear blendshapes to achieve more plausible deformations [30]. On the other hand, more generalized digital face models would be more useful in cases where the target actor is not known beforehand. Such models include the classic Blanz and Vetter model [8], the Basel Face Model (BFM) [36, 37], FaceWarehouse [11], and the Large Scale Facial Model (LSFM) [9]. Recent models such as the FLAME model [31] have begun to introduce non-linear deformations by using skinning and corrective blendshapes. These models tend to be geared towards real-time applications and as a result have a low number of vertices.

Face Capture: A more comprehensive review of facial performance capture techniques can be found in [53]. To date, marker based techniques have been the most popular for capturing facial performances for both real-time applications and feature films. Helmet mounted cameras (HMCs) are often used to stereo track a sparse set of markers on the face. These markers are then used as constraints in an optimization to find blendshape weights [7]. In many real-time applications, pre-applied markers are generally not an option so 2D features [10, 12, 50], depth images [12, 26, 49], or low-resolution RGB images [48] are often used instead. More recently, methods using neutral networks have been used to reconstruct face geometry [24, 41] and estimate facial control parameters [25, 27]. Analysis-by-synthesis techniques have also been explored for capturing facial performances [38].

Face Simulation: [42] was one of the first to utilize quasistatic simulations to drive the deformation of a 3D face, especially for motion capture. There has also been interest in using quasistative simulations to drive muscle deformations in the body [23, 45, 46]. However, in general, facial muscle simulations tend to be less expressive than their artist-driven blendshape counterparts. More recently, significant work has been done to make muscle simulations more expressive [14, 21]. While these methods can be used to target data in the form of geometry, it is unclear how to cleanly transfer these methods to target non-geometry data such as two-dimensional RGB images. Other work has been done to try to introduce physical simulations into the blendshape models themselves [3, 4, 22, 28]; however, these works do not focus on the inverse problem.

3. Blendshape Model

As discussed in Section 2, there are many different types of blendshape models that exist and we refer interested readers to [30] for a more thorough overview of existing literature. We focus on the state-of-the-art hybrid blendshape deformation model that is the basis of our method introduced in Section 6. A hybrid blendshape model refers to a deformation model that uses both linear blendshapes and linear blend skinning to deform the vertices of the mesh. Our model contains a single 6-DOF joint for the jaw. We can succinctly write the model given the blendshape parameters b and joint parameters j as

$$x(b, j) = T(j)(n + Bb) \tag{1}$$

where n is the neutral shape, B is the blendshape deltas matrix, and $T(j)$ contains the linear blend skinning matrix, *i.e.* a transformation matrix due to a change in the jaw joint, for each vertex. Note that $n + Bb$ is often referred to as the *pre-skinning* shape and Bb as the *pre-skinning* displacements. More complex animation systems include corrective shapes and intermediate controls and thus we let w denote a broader set of animator controls which we treat as our independent variable rewriting Equation 1 as

$$x(w) = T(j(w))(n + Bb(w)) \tag{2}$$

where $j(w)$ and $b(w)$ may include non-linearities such as non-linear corrective blendshapes.

4. Muscle Model

We create an anatomical model of the face consisting of the cranium, jaw, and a tetrahedralized flesh mesh with embedded muscles for a given actor using the method of [13]. Since we desire parity with the facial model used to deform the face surface, we define the jaw joint as a 6-DOF joint equivalent to the one used to skin the face surface in Section 3. Traditionally, face simulation models have been controlled using a vector of muscle activation parameters which we denote as a. We use the same constitutive model for the muscle as [45, 46] which consists of an isotropic Mooney-Rivlin term, a quasi-incompressibility term, and an anisotropic passive/active muscle response term. The finite-volume method [45, 47] is used to compute the force on each vertex of the tetrahedralized flesh mesh given the current 1st Piola-Kirchoff stress computed using the constitutive model and the current deformation gradient. Some vertices of the flesh mesh X^C are constrained to kinematically follow along with the cranium/jaw and the steady state position is implicitly defined as the positions of the unconstrained flesh mesh vertices X^U which make the sum of all relevant forces identically 0, *i.e.* $f(X^C, X^U) = 0$.

One can decompose the forces to be a sum of the finite-volume forces and collision penalty forces

$$f_{\text{fvm}}(X^C, X^U, a) + f_{\text{collisions}}(X^C, X^U) = 0. \tag{3}$$

One can further break down the finite-volume forces into the passive force f_p and active force f_a. Then using the fact the the active muscle response is scaled linearly by the muscle activation a [51], we can rewrite the finite-volume force as

$$f_{\text{fvm}}(X^C, X^U, a) = f_{\text{p}}(X^C, X^U) + a f_{\text{a}}(X^C, X^U). \quad (4)$$

We refer interested readers to [42, 45, 46, 47] for derivations of the aforementioned forces and their associated Jacobians with respect to the flesh mesh vertices. Given a vector of muscle activations and cranium/jaw parameters, Equation 3 can be solved using the Newton-Raphson method to compute the unconstrained flesh mesh vertex positions X^U.

5. Muscle Tracks

The muscle tracks simulation introduced by [14] modifies the framework described in Section 4 such that the muscle deformations are primarily controlled by a volumetric morph [1, 13] rather than directly using muscle activation values. [14] first creates a correspondence between the neutral pose n of the blendshape system and the outer boundary surface of the tetrahedral mesh X^b. Then, given a blendshape target expression $x^*(b, j)$ with surface mesh displacements $x^* - n$, [14] creates target displacements for the outer boundary of the tetrahedral mesh δX^b. Using δX^b as Dirichlet boundary conditions, [14] solves a Poisson equation for the displacements $\delta X = X - X_0$, *i.e.* $\nabla^2 \delta X = 0$, where X_0 are the rest-state vertex positions consistent with the neutral pose n. Neumann boundary conditions are used on the inner boundary of the tetrahedral mesh. Afterwards, zero-length springs are attached between the tetrahedralized flesh mesh vertices interior to each muscle and their corresponding target locations resulting from the Poisson equation. The muscle track force resulting from the zero-length springs for each muscle m has the form

$$f_{\text{tracks},m} = K_m(M_m - I_m X^U) \quad (5)$$

where K_m is the per-muscle spring stiffness matrix, I_m is the selector matrix for the flesh mesh vertices interior to the muscle, and M_m are the target locations resulting from the volumetric morph. Thus the expanded quasistatics equation can be written as

$$f_{\text{fvm}} + f_{\text{collisions}} + f_{\text{tracks}} = 0 \quad (6)$$

where f_{tracks} includes Equation 5 for every muscle. Since the activation values a are no longer specified manually, they must be computed automatically given the final post-morph shape of a muscle to reintroduce the effects of muscle tension into the simulation. [14] barycentrically embeds a piecewise linear curve into each muscle and uses the length of that curve to determine an appropriate activation value.

6. Blendshape-Driven Muscle Tracks

The morph from Section 5 was designed in the spirit of the computer graphics pipeline, and as such, does not allow for the sort of full end-to-end coupling that facilitates differentiability, inversion, and other typical inverse problem methodologies. Thus, our key contribution is to replace the morphing step with a blendshape deformation in the form of Equation 1 to drive the muscle volumes and their center-line curves thereby creating a direct functional connection between the animator controls w and the muscle tracks target locations M_m and activation values a.

For each muscle, we create a tetrahedralized volume M_m^0 and piecewise linear center-line curve C_m^0 in the neutral pose. Furthermore, for each blendshape in the face surface model, we use the morph from [14] to create a corresponding shape for each muscle's tetrahedralized volume M_m^k and center-line curve C_m^k, where k is used to denote the kth blendshape. Alternatively, one could morph and subsequently simulate as in Section 5 using tracks in order to create M_m^k and C_m^k. In addition, we assign skinning weights to each vertex in M_m^0 and C_m^0 and assemble them into linear blend skinning transformation matrices T_m^M and T_m^C. This allows us to write

$$M_m(b, j) = T_m^M(j) \left(M_m^0 + \sum_k M_m^k b_k \right) \quad (7)$$

$$C_m(b, j) = T_m^C(j) \left(C_m^0 + \sum_k C_m^k b_k \right) \quad (8)$$

which parallel Equation 1. Notably, we are able to obtain Equations 7 and 8 in part because we solve the Poisson equation on the pre-skinning neutral as compared to [14] which uses the post-skinning neutral. In addition, this better prevents linearized rotation artifacts from diffusing into the tetrahedralized flesh mesh. Finally, we can write the length of each center-line curve as

$$L(C_m(b, j)) = \sum_i ||C_{m,i}(b, j) - C_{m,i-1}(b, j)||_2 \quad (9)$$

where $C_{m,i}(b, j)$ is the ith vertex of the piecewise linear center-line curve for the mth muscle.

To justify our approach, we can write the linear system to solve the Poisson equation as $A^U(X_0)\delta X = A^C(X_0)Bb$ where $A^U(X_0)$ is the portion of the Laplacian matrix discretized on the tetrahedralized volume at rest using the method of [52] for the unconstrained vertices. Similarly, $A^C(X_0)$ is the portion for the constrained vertices post-multiplied by the linear correspondence between the neutral pose n of the blendshape system and the outer boundary of the tetrahedral mesh X^b. Equivalently, we may write

$$A^U(X_0)\delta X = \sum_k A^C(X_0) B e_k b_k \quad (10)$$

(where e_k are the standard basis vectors) which is equivalent to doing k solves of the form

$$A^U(X_0)\delta X_k = A^C(X_0)Be_k \tag{11}$$

and then subsequently summing both sides to obtain $\delta X = \sum_k \delta X_k b_k$. That is, the linearity of the Poisson equation allows us to precompute its action for each blendshape and subsequently obtain the exact result on any combination of blendshapes by simply summing the results obtained on the individual blendshapes.

In summary, for each of the k blendshapes, we solve a Poisson equation (Equation 11) to precompute M_m^k and C_m^k, and then given animator controls w which yield b and j, we obtain M_m and C_m via Equations 7 and 8. This replaces the morphing step allowing us to proceed with the quasistatic muscle simulation using tracks driven entirely by the animator parameters w.

7. End-to-End Differentiability

In this section, we outline the derivative of the simulated tetrahedral mesh vertex positions with respect to the blendshape parameters b and jaw controls j that parameterize the simulation results as per Section 6. The derivative of b and j with respect to the animator controls w depend on the specific controls and can be post-multiplied. If one cares about the resulting vertices of a rendered mesh embedded in or constrained to the tetrahedral mesh, then this embedding, typically linear, can be pre-multiplied.

Although the constrained nodes X^C typically only depend on the joint parameters, one may wish, at times, to simulate only a subset of the tetrahedral flesh mesh. In such instances, the constrained nodes can appear on the unsimulated boundary which in turn can be driven by the blendshape parameters b; thus, we write $X^C(b, j)$ and concatenate it with $X^U(b, j)$ to obtain $X(b, j)$ for the purposes of this section. The collision forces only depend on the nodal positions, and we may write $f_{\text{collisions}}(X(b, j))$. The finite volume force depends on both the nodal positions and activations, and the activations are determined from an activation-length curve where the length is given in Equation 9. Our precomputation makes C_m only a function of b and j and notably independent of X, and so we may write $a_m(L_m(C_m(b, j)))$ combining the activation length curve with Equations 8 and 9. We stress that the activations are independent of the positions, X. Thus, we may write $f_{\text{fvm}}(X(b, j), C(b, j))$. Similarly, we may write $f_{\text{tracks}}(X(b, j), M(b, j))$. Therefore, all the forces in Equation 6 are a function of X, C, and M which are in turn a function of b and j.

Using the aforementioned dependencies, we can take the total derivative of the forces $f_T = f_{\text{fvm}} + f_{\text{collisions}}$ in Equation 3 with respect to a single blendshape parameter b_k to obtain $(\partial f_T/\partial X)(\partial X/\partial b_k) + (\partial f_T/\partial C)(\partial C/\partial b_k) = 0$ which is equivalent to $(\partial f_T/\partial X)(\partial X/\partial b_k) + (\partial f_{\text{fvm}}/\partial C)(\partial C/\partial b_k) = 0$ since $f_{\text{collisions}}$ is independent of C. Since our activations are still independent of X just as they were in [42], $\partial f_T/\partial X$ here is identical to that discussed in [42], and thus their quasistatic solve can be used to determine $\partial X/\partial b_k$ by solving $(\partial f_T/\partial X)(\partial X/\partial b_k) = -(\partial f_{\text{fvm}}/\partial C)(\partial C/\partial b_k)$. To compute the right hand side, note that $\partial C/\partial b_k$ can be obtained from Equation 8. To obtain $\partial f_{\text{fvm}}/\partial C$, we compute $\partial f_{\text{fvm}}/\partial C = (\partial f_{\text{fvm}}/\partial a)(\partial a/\partial L)(\partial L/\partial C)$. $\partial f_{\text{fvm}}/\partial a$ are simply the active forces f_a in Equation 4, $\partial a/\partial L$ is the local slope of the activation length curve, and $\partial L/\partial C$ is readily computed from Equation 9. The $\partial X/\partial j_k$ are determined similarly.

One may take a similar approach to Equation 6, obtaining $\partial X/\partial b_k$ by solving $(\partial f_T/\partial X)(\partial X/\partial b_k) = -(\partial f_{\text{fvm}}/\partial C)(\partial C/\partial b_k) - (\partial f_{\text{tracks}}/\partial M)(\partial M/\partial b_k)$. We stress that the coefficient matrix $\partial f_T/\partial X$ of the quasistatic solve is now augmented by $\partial f_{\text{tracks}}/\partial X$ (see Equation 5) and is the same quasistatic coefficient matrix in [14]. $\partial f_{\text{tracks}}/\partial M$ and $\partial M/\partial b_k$ are obtained from Equations 5 and 7 respectively. Again, the $\partial X/\partial j_k$ are found similarly. In summary, finding $\partial X/\partial b_k$ and $\partial X/\partial j_k$ involves solving the same quasistatics problem of [42] with the slight augmention to the coefficient matrix from [14] merely with different right hand sides. Although this requires a quasistatic solve for each b_k and j_k, they are all independent and can thus be done in parallel.

8. Experiments

We use the Dogleg optimization algorithm [35] as implemented by the Chumpy autodifferentation library [33] in order to target our face model to both three-dimensional geometry and two-dimensional RGB images to demonstrate the efficacy of our end-to-end fully differentiable formulation. Other optimization algorithms and/or applications may similarly be pursued. Our nonlinear least squares optimization problems generally have the form

$$\min_w ||F^* - F(x_R(w))||_2^2 + \lambda||w||_2^2 \tag{12}$$

where w are the animator controls that deform the face, $x(w)$ are the positions of the vertices on the surface of the face deformed using the full blendshape-driven muscle simulation system as described in Section 6, $F(x_R(w))$ is a function of those vertex positions, and F^* is the desired output of that function. $R(\theta)$ and t are an additional rigid rotation and translation, respectively where θ represents Euler angles, *i.e.* $x_R(w) = R(\theta)x(w) + t$. We use a standard L2 norm regularization on the animator controls $||w||_2^2$, where λ is set experimentally to avoid overfitting.

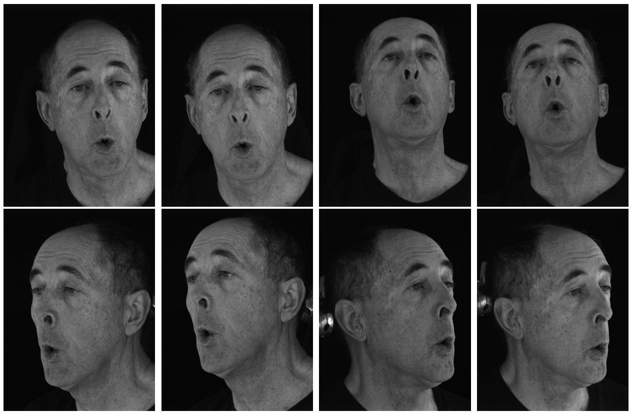
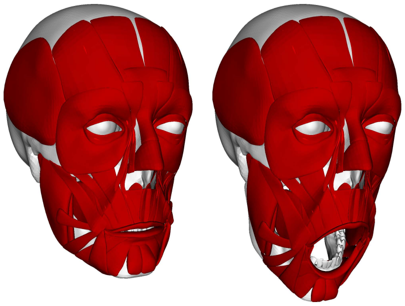


Figure 1. The eight viewpoints used to reconstruct the facial geometry for a particular pose.

8.1. Model Creation

The blendshape system is created from the neutral pose n as well as FACS-based expressions [17] using the methods of [5, 6]. Eight black and white cameras from varying viewpoints (see Figure 1) are used to reconstruct the geometry of the actor. Artists clean up these scans and use them as inspiration for a blendshape model and to calibrate the linear blend skinning matrices for the face surface (see Figure 3). Of course, any reasonable method could be used to create the blendshape system. A subset of the full face surface model with 52,228 surface vertices is used in the optimization.

We use the neutral pose of the blendshape system and the method of [13] to create the tetrahedral flesh mesh X_0, tetrahedral muscle volumes M_m^0, and muscle center line curves C_m^0 by morphing them from a template asset. Our simulation mesh has 302,235 vertices and 1,470,102 tetrahedra. We use 60 muscles with a total of 50,710 vertices and 146,965 tetrahedra (some tetrahedra are duplicated between muscles due to overlap). The linear blend skinning weights used to form T_j on the face surface are propagated to the surface of the tetrahedral mesh and used as boundary conditions in a Poisson equation solve again as in [1, 13] to obtain linear blend skinning weights throughout the volumetric tetrahedral mesh as well as for the muscles and center-line curves, thus defining skinning transformation matrices T_m^M and T_m^C. Figure 2 shows the muscles in the neutral pose M_m^0 as well as the result after skinning with the jaw open, *i.e.* Equation 7 with all b_k identically 0.



(a) Neutral (b) Jaw Open

Figure 2. The underlying anatomical model of the face in the neutral pose as well as the jaw open pose using linear blend skinning.

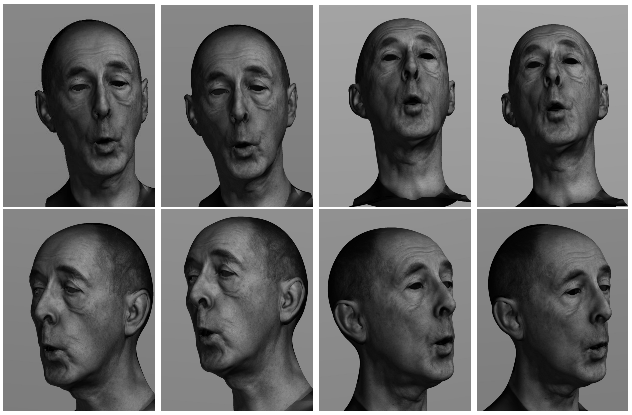
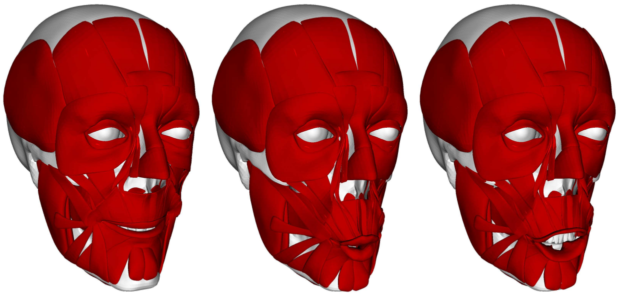


Figure 3. The geometry reconstructed by applying the multi-view stereo algorithm described in [5, 6] to the input images shown in Figure 1.

Finally, for each shape in the blendshape system, we solve a Poisson equation (Equation 11) for the vertex displacements δX_k which are then transferred to the muscle volumes and center-line curves to obtain M_m^k and C_m^k. This allows us full use of Equations 7 and 8 parameterized by the blendshapes b_k. Figure 4 shows some examples of the muscles evaluated using Equation 7 for a variety of expressions.

8.2. Targeting 3D Geometry

Oftentimes, one has captured a facial pose in the form of three-dimensional mesh; however, this data is generally noisy, and it is desirable to convert this data into a lower dimensional representation. Using a lower dimensional representation facilitates editing, extracting semantic informa-



(a) Smile (b) Pucker (c) Funneler

Figure 4. The anatomical model of the face peforming a variety of expressions using only the blendshape deformation from Equation 7.

tion, and performing statistical analysis. In our case, the lower dimensional representation is the parameter space of the blendshape or simulation model.

In general, extracting a lower dimensional representation from an arbitrary mesh requires extracting a set of correspondences between the mesh and the face model. However, for simplicity, we assume that the correspondence problem has been solved beforehand and that each vertex of the incoming mesh captured by a system using the methods of [5, 6] has a corresponding vertex on our face surface. We can thus use an optimization problem in the form of Equation 12 to solve for the model parameters where F^* are the vertex positions of the target geometry, and $F(x) = x$ is the identity function.

While a rigid alignment between the F^* and the neutral mesh n, *i.e.* $R(\theta)$ and t, is created as a result of [5, 6], we generally found it to be inaccurate. As a result, we also allow the optimization to solve for θ and t as well. Our optimization problem for targeting three-dimensional geometry thus has the form

$$\min_{w,\theta,t}||F^* - x_R(w)||_2^2 + \lambda||w||_2^2 \qquad (13)$$

where $\lambda = 1 \times 10^{-6}$ is set experimentally.

We demonstrate the efficacy of our method on a pose where the actor has his mouth slightly open and is making a pucker shape. We compare the results of targeting three-dimensional geometry when it is driven using simulation via the blendshape muscle tracks as described in Section 6 versus when it is driven using the pure blendshape model described in Section 3. Traditionally, pucker shapes have been difficult for activation-muscle based simulations to hit. See Figure 5. Although neither inversion quite captures the tightness of the mouth's pucker, the muscle simulation results demonstrate how the simulation's volume preservation property significantly improves upon the blendshape results where the top and bottom lips seem to shrink. This property is also useful in preserving the general shape of the philtrum; the blendshape models's inversion causes the part of the philtrum near the nose to incorrectly bulge significantly. Furthermore, the resulting muscle activation values are easier to draw semantic meaning from due to their sparsity and anatomical meaning as seen in Figure 6.

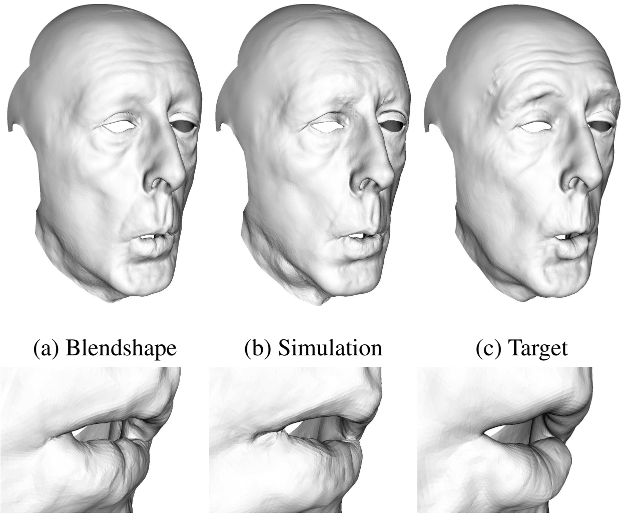

(a) Blendshape (b) Simulation (c) Target

Figure 5. We target the geometry shown in (c) using purely blendshapes shown in (a) versus the blendshape driven muscle simulation model shown in (b). While neither method exactly matches target geometry, in general, we found that the simulation results preserve key physical properties such as volume preservation around the lips. A close-up of the lips is shown in the bottom row where it is more apparent how the pure blendshape inversion has significant volume loss around the lips.

Note that errors in the method of [5, 6] in performing multi-view reconstruction will cause the vertices of the target geometry to contain noise and potentially be in physically implausible locations. Additionally, errors in finding correspondences between the target geometry and the face surface will result in an inaccurate objective function. Furthermore, there is no guarantee that our deformation model $x(w)$ is able to hit all physically attainable poses even when the capture and correspondence are perfect. This demonstrates the efficacy of introducing physically-based priors into the optimization. Additional comparisons and results are shown in the supplementary material and video.

8.3. Targeting Monocular RGB Images

To further demonstrate the efficacy of our approach, we consider facial reconstruction from monocular RGB images. The images were captured using an 100mm lens attached to an ARRI Alexa XT Studio running at 24 frames-per-second with an 180 degree shutter angle at ISO 800.

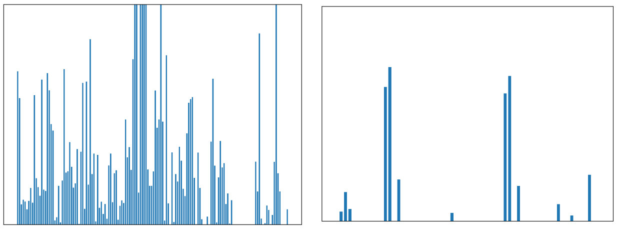

(a) Blendshape Weights (b) Muscle Activations

Figure 6. The blendshape solve results in blendshape weights that are dense, overdialed, and hard to decipher. Whereas all 129 (of 146; shapes for the neck, etc. were not used) blendshapes used have non-zero values, only 13 of the available 60 muscles have non-zero activation values. The top four most activated muscles are related to the frontalis indicating that the eyebrows are raised [43]. The activations of the incisivus labii superioris and orbicularis oris muscles are also among the top activated muscles properly indicating a compression of the lips [20, 43].

We refer to images captured by the camera as the "plates." The original plates have a resolution of 2880×2160, but we downsample them to 720×540. The camera was calibrated using the method of [19] and the resulting distortion parameters are used to undistort the plate to obtain F^*.

$F(x)$ renders the face geometry in its current pose with a set of camera, lighting, and material parameters. We use a simple pinhole camera with extrinsic parameters determined by the camera calibration step. The rigid transformation of the face is determined by manually tracking features on the face in the plate. The face model is lit with a single spherical harmonics light with 9 coefficients γ, see [39], and is shaded with Lambertian diffuse shading. Each vertex i also has an RGB color c_i associated with it. We solve for γ and all c_i using a non-linear least squares optimization of the form

$$\min_{\gamma,c} ||F^* - F(x_R(0), \gamma, c)||_2^2 + \lambda ||S(c)||_2^2 \tag{14}$$

where the per-vertex colors is regularized using $S(c) = \sum_i \sum_{j \in N(i)} c_i - c_j$ where $N(i)$ are the neighboring vertices of vertex i. This lighting and albedo solve is done as a preprocess on a neutral or close to neutral pose with $\lambda = 2500$ set experimentally. OpenDR [34] is used to differentiate $F(x)$ to solve Equation 14; however, any other differentiable renderer (*e.g.* [32]) can be used instead. Then we assume that γ and c stay constant throughout the performance. See Figure 7.

We solve for the parameters w in two steps. Given curves around the eyes and lips on the three-dimensional neutral face mesh, a rotoscope artist draws corresponding curves on the two-dimensional film plate. Then, we solve for an initial guess $\hat{w}$ by solving an optimization problem of the form

$$\min_{\hat{w}} ||E_1(\hat{w})||_2^2 + \lambda_1 ||\hat{w}||_2^2 \tag{15}$$

where $\lambda_1 = 3600$ is set experimentally. $E_1(\hat{w})$ is the two-dimensional Euclidean distance between the points on the rotoscoped curves on the plate and the corresponding points on the face surface $x(w)$ projected into the image plane. See Figure 8. We then use $\hat{w}$ to initialize a shape from shading solve

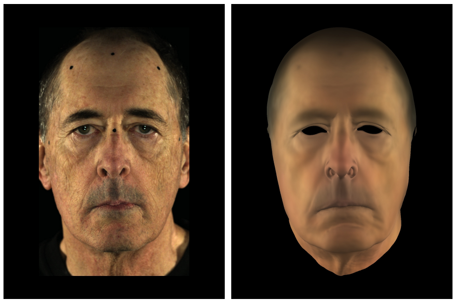
$$\min_{w} ||E_2(w)||_2^2 + \lambda_1 ||E_1(w)||_2^2 + \lambda_2 ||w - \hat{w}||_2^2 \tag{16}$$

to determine the final parameters w where $\lambda_1 = 1 \times 10^{-4}$ and $\lambda_2 = 1$ are set experimentally. Here, $E_2 = G(F^* - F(x_R(w), \gamma, c))$ is a three-level Gaussian pyramid of the per-pixel differences between the plate and the synthetic render.

We demonstrate the efficacy of our approach on 66 frames of a facial performance. As in Section 8.2, we compare the results of solving Equations 15 and 16 using $x(w)$ driven by a simulation model versus a blendshape model. In particular, we choose four frames with particularly challenging facial expressions (frames 1112, 1160, 1170) as well as capture conditions such as motion blur (frame 1134). We note that a significant portion of the facial expression is captured using the rotoscoped curves and the shape-from-shading step primarily helps to refine the expression and the contours of the face. Both E_1 and E_2 (Equations 15 and 16) require end-to-end differentiability through our blendshape driven method. See Figure 9. While the general expressions are similar, we note that the simulation's surface geometry tends to be more physically plausible due the simulation's ability to preserve volume, especially around the lips. This regularization is especially prominent on frame 1134. As shown in supplementary material, the resulting muscle activation values are also comparatively sparser which leads to an increased ability to extract semantic meaning out of the performance. Additional comparisons and results are shown in the supplementary material and video.

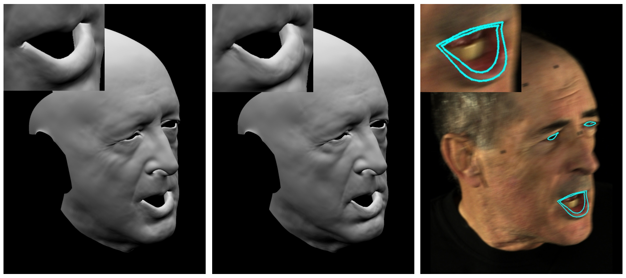
9. Conclusion and Future Work

Although promising anatomically based muscle simulation systems have existed for some time and have had the ability to target data as in [42], they have lacked the high-end efficacy required to produce compelling results. Al-



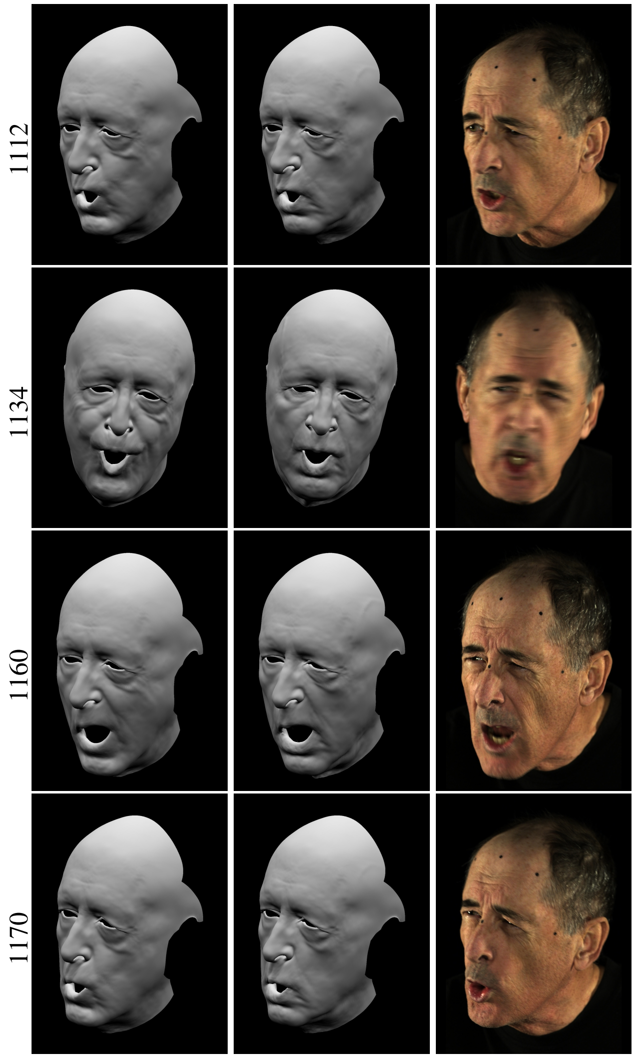
(a) Plate (b) Lighting/Albedo

Figure 7. Before estimating the facial pose, we first estimate lighting and albedo on a neutral or close to neutral pose.



(a) Blendshapes (b) Simulation (c) Roto Curves

Figure 8. We use rotoscoped curves on the plate to solve for an initial estimate of the face pose.



(a) Blendshapes (b) Simulation (c) Plate

Figure 9. We target the raw image data using our face model $x(w)$ using both simulation and blendshapes on a number of frames of an actor's performance. Both sets of results suffer from some depth ambiguity due to only using monocular two-dimensional data in the optimization.

though the recently proposed [14] does produce quite compelling results, it requires a full face shape as input and is not differentiable. In this paper, we alleviated both of the aforementioned difficulties, extending [14] with end-to-end differentiability and a morphing system driven by blendshape parameters. This blendshape-driven morph removes the need for a full face surface mesh as a pre-existing target. We demonstrate the efficacy of our approach by targeting three-dimensional geometry and two-dimensional RGB images. To the best of our knowledge, we are the first to use quasistatic simulation of a muscle model to target RGB images. We note that methods such as [42] could be used in the optimizations presented in this paper (as outlined in the second to last paragraph of Section 7); however, the resulting simulation results would be less expressive and would not be able to effectively reproduce the desired expressions.

Although the computer vision community expends great efforts in regards to identifying faces in images, segmenting them cleanly from their surroundings, and even identifying their shape, semantic understanding of what such faces are doing or intend to do or feel is still in its infancy consisting mostly of preliminary image labeing and annotation. The ability to express a facial pose or image using a muscle activation basis provides an anatomically-motivated way to extract semantic information. Even without extensive model calibration, our anatomical model's muscle activations have shown to be useful for extracting anatomically-based smenatic information. This is a promising avenue for future work. Additionally, muscle activations could also be used as a basis for statistical/deep learning instead of semantically meaningless combinations of blendshape weights.

Finally, one of the more philosophical questions in deep learning seems to revolve around what should or should not be considered a "learning crime" (drawing similarities to variational crimes [44]). For example, in [2], the authors learn a perturbation of linear blend skinning as opposed to the whole shape, assuming that the perturbation is lower-dimensional, spatially correlated, and/or easier to learn. The authors in [18, 40] use spatially correlated networks for spatially correlated information under the assumption, once again, that this leads to a network that is easier to train and generalizes better. It seems that adding strong priors, domain knowledge, informed procedural methods, etc. to generate as much of a function as possible before training a network to learn the rest is often considered prudent. Our anatomically-based physical simulation system incorporates physical properties such as volume preservation, contact, and collision so that a network would not need to learn or explain them; instead the network only needs to learn what further perturbations are required to match the data.

Acknowledgements

Research supported in part by ONR N00014-13-1-0346, ONR N00014-17-1-2174, ARL AHPCRC W911NF-07-0027, and generous gifts from Amazon and Toyota. In addition, we would like to thank both Reza and Behzad at ONR for supporting our efforts into computer vision and machine learning, as well as Cary Phillips, Kiran Bhat, and Industrial Light & Magic for supporting our efforts into facial performance capture. M. Bao was supported in part by The VMWare Fellowship in Honor of Ole Agesen. We would also like to thank Paul Huston for his acting and Jane Wu for her help in preparing the supplementary video.

References

[1] Dicko Ali-Hamadi, Tiantian Liu, Benjamin Gilles, Ladislav Kavan, François Faure, Olivier Palombi, and Marie-Paule Cani. Anatomy transfer. *ACM Transactions on Graphics (TOG)*, 32(6):188, 2013.

[2] Stephen W Bailey, Dave Otte, Paul Dilorenzo, and James F O'Brien. Fast and deep deformation approximations. *ACM Transactions on Graphics (TOG)*, 37(4):119, 2018.

[3] V Barrielle and N Stoiber. Realtime performance-driven physical simulation for facial animation. In *Computer Graphics Forum*. Wiley Online Library, 2018.

[4] Vincent Barrielle, Nicolas Stoiber, and Cédric Cagniart. Blendforces: A dynamic framework for facial animation. In *Computer Graphics Forum*, volume 35, pages 341–352. Wiley Online Library, 2016.

[5] Thabo Beeler, Bernd Bickel, Paul Beardsley, Bob Sumner, and Markus Gross. High-quality single-shot capture of facial geometry. In *ACM Transactions on Graphics (ToG)*, volume 29, page 40. ACM, 2010.

[6] Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W Sumner, and Markus Gross. High-quality passive facial performance capture using anchor frames. In *ACM Transactions on Graphics (TOG)*, volume 30, page 75. ACM, 2011.

[7] Kiran S Bhat, Rony Goldenthal, Yuting Ye, Ronald Mallet, and Michael Koperwas. High fidelity facial animation capture and retargeting with contours. In *Proceedings of the 12th ACM SIGGRAPH/eurographics symposium on computer animation*, pages 7–14. ACM, 2013.

[8] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[9] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3d morphable model learnt from 10,000 faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5543–5552, 2016.

[10] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 3d shape regression for real-time facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):41, 2013.

[11] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014.

[12] Yen-Lin Chen, Hsiang-Tao Wu, Fuhao Shi, Xin Tong, and Jinxiang Chai. Accurate and robust 3d facial capture using a single rgbd camera. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3615–3622. IEEE, 2013.

[13] Matthew Cong, Michael Bao, Jane E, Kiran S Bhat, and Ronald Fedkiw. Fully automatic generation of anatomical face simulation models. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 175–183. ACM, 2015.

[14] Matthew Cong, Kiran S Bhat, and Ronald Fedkiw. Art-directed muscle simulation for high-end facial animation. In *Symposium on Computer Animation*, pages 119–127, 2016.

[15] Matthew Cong, Lana Lan, and Ronald Fedkiw. Muscle simulation for facial animation in kong: Skull island. In *ACM SIGGRAPH 2017 Talks*, page 21. ACM, 2017.

[16] Paul Debevec. The light stages and their applications to photoreal digital actors. *SIGGRAPH Asia*, 2(4), 2012.

[17] Paul Ekman. Facial action coding system (facs). *A human face*, 2002.

[18] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. *arXiv preprint arXiv:1803.07835*, 2018.

[19] Janne Heikkila and Olli Silven. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112. IEEE, 1997.

[20] Mi-Sun Hur. Anatomical features of the incisivus labii superioris muscle and its relationships with the upper mucolabial fold, labial glands, and modiolar area. *Scientific reports*, 8(1):12879, 2018.

[21] Alexandru-Eugen Ichim, Petr Kadleček, Ladislav Kavan, and Mark Pauly. Phace: physics-based face modeling and animation. *ACM Transactions on Graphics (TOG)*, 36(4):153, 2017.

[22] Alexandru Eugen Ichim, Ladislav Kavan, Merlin Nimier-David, and Mark Pauly. Building and animating user-specific volumetric face rigs. In *Symposium on Computer Animation*, pages 107–117, 2016.

[23] Geoffrey Irving, Joseph Teran, and Ronald Fedkiw. Invertible finite elements for robust simulation of large deformation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 131–140. Eurographics Association, 2004.

[24] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 1031–1039. IEEE, 2017.

[25] Amin Jourabloo and Xiaoming Liu. Large-pose face alignment via cnn-based dense 3d model fitting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4188–4196, 2016.

[26] Vahid Kazemi, Cem Keskin, Jonathan Taylor, Pushmeet Kohli, and Shahram Izadi. Real-time face reconstruction from a single depth image. In *3D Vision (3DV), 2014 2nd international conference on*, volume 1, pages 369–376. IEEE, 2014.

[27] Hyeongwoo Kim, Michael Zollhöfer, Ayush Tewari, Justus Thies, Christian Richardt, and Christian Theobalt. Inversefacenet: Deep monocular inverse face rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4625–4634, 2018.

[28] Yeara Kozlov, Derek Bradley, Moritz Bächer, Bernhard Thomaszewski, Thabo Beeler, and Markus Gross. Enriching

facial blendshape rigs with physical simulation. In *Computer Graphics Forum*, volume 36, pages 75–84. Wiley Online Library, 2017.

[29] Lana Lan, Matthew Cong, and Ronald Fedkiw. Lessons from the evolution of an anatomical facial muscle model. In *Proceedings of the ACM SIGGRAPH Digital Production Symposium*, page 11. ACM, 2017.

[30] John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H Pighin, and Zhigang Deng. Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)*, 1(8), 2014.

[31] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics (TOG)*, 36(6):194, 2017.

[32] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018.

[33] M Loper. Chumpy autodifferentation library. `http://chumpy.org`, 2014.

[34] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014.

[35] MLA Lourakis and Antonis A Argyros. Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment? In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1526–1531. IEEE, 2005.

[36] Marcel Lüthi, Thomas Gerig, Christoph Jud, and Thomas Vetter. Gaussian process morphable models. *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[37] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *Advanced video and signal based surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 296–301. Ieee, 2009.

[38] Frédéric Pighin, Richard Szeliski, and David H Salesin. Resynthesizing facial animation through 3d model-based tracking. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 143–150. IEEE, 1999.

[39] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500. ACM, 2001.

[40] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. *arXiv preprint arXiv:1807.10267*, 2018.

[41] Matan Sela, Elad Richardson, and Ron Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1585–1594. IEEE, 2017.

[42] Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *Acm transactions on graphics (tog)*, 24(3):417–425, 2005.

[43] Susan Standring. *Gray's anatomy e-book: the anatomical basis of clinical practice*. Elsevier Health Sciences, 2015.

[44] Gilbert Strang. Variational crimes in the finite element method. In *The mathematical foundations of the finite element method with applications to partial differential equations*, pages 689–710. Elsevier, 1972.

[45] Joseph Teran, Sylvia Blemker, V Hing, and Ronald Fedkiw. Finite volume methods for the simulation of skeletal muscle. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 68–74. Eurographics Association, 2003.

[46] Joseph Teran, Eftychios Sifakis, Silvia S Blemker, Victor Ng-Thow-Hing, Cynthia Lau, and Ronald Fedkiw. Creating and simulating skeletal muscle from the visible human data set. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):317–328, 2005.

[47] Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. Robust quasistatic finite elements and flesh simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 181–190. ACM, 2005.

[48] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016.

[49] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. In *ACM transactions on graphics (TOG)*, volume 30, page 77. ACM, 2011.

[50] Chenglei Wu, Derek Bradley, Markus Gross, and Thabo Beeler. An anatomically-constrained local deformation model for monocular face capture. *ACM Transactions on Graphics (TOG)*, 35(4):115, 2016.

[51] Felix E Zajac. Muscle and tendon properties models scaling and application to biomechanics and motor. *Critical reviews in biomedical engineering*, 17(4):359–411, 1989.

[52] Wen Zheng, Bo Zhu, Byungmoon Kim, and Ronald Fedkiw. A new incompressibility discretization for a hybrid particle mac grid representation with surface tension. *Journal of Computational Physics*, 280:96–142, 2015.

[53] M Zollhöfer, J Thies3 P Garrido, D Bradley4 T Beeler4 P Pérez, M Stamminger, and M Nießner3 C Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. *STAR*, 37(2), 2018.