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

Art-Directed Muscle Simulation for High-End Facial Animation

Matthew Cong[†1,2], Kiran S. Bhat[‡2], Ronald Fedkiw[†1,2]

[1]Stanford University
[2]Industrial Light + Magic

Abstract

We propose a new framework for the simulation of facial muscle and flesh that so significantly improves the technique that it allows for immediate mainstream use of anatomically and biomechanically accurate muscle models as a bread and butter technique in a high-end production quality pipeline. The key idea is to create a blendshape system for the muscles that gives the precise directability and controllability required in a high-end production environment. The blendshape muscles are used to drive the underlying anatomically and biomechanically motivated simulation in a way that is unbound by the typical restrictions of a simulation system while still retaining the desirable degree of freedom richness that leads to high quality results. We show that we are able to target production quality facial shapes, whether from scans or an animation system, and illustrate that the resulting nonlinear simulation in-betweens are of higher quality than those obtained from traditional linear blendshapes. We also demonstrate the ability to selectively improve areas on a given blendshape using the results of a simulation, as well as the ability to edit muscle shapes and paths in order to produce directability for animator control. Then, we show how these techniques can be used to transition from one blendshape to another or even track and selectively modify an entire performance. The efficacy of our system is further demonstrated by using it to retarget animation onto new creature models given only a single static rest pose as input.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Researchers have struggled for years with the uncanny valley in facial animation (see e.g. [BGY*13], see [LAR*14] for review). The rendering of still faces under varying lighting conditions has progressed significantly, and recent results are becoming quite realistic climbing their way out of the uncanny valley ([BL05, vdPJD*14, FJA*15]). On the other hand, while facial animation techniques have improved significantly in recent years, various artifacts such as volume preservation, texture stretching, etc. have hindered their ability to shake loose criticisms of the uncanny valley. Similar to the understanding of light interacting with surfaces or the partial differential equations that govern fluids and solids, one would hope that a better understanding of the anatomical and biomechanical aspects of the face would be as beneficial to facial animation as the understanding of subsurface scattering in marble and skin is to rendering [JMLH01], or vorticity and interface merging in fluids is to simulation [FSJ01, EMF02]. However, anatomical and biomechanical muscle models have not as of yet made significant impact on state of the art production technologies for facial animation.

One of the reasons that anatomical and biomechanical facial muscle and flesh models have not made much of an impact on production pipelines is perhaps because it is quite time-consuming to create such a model. Although creating similar models for skeletal muscles and flesh suffers from some of the same issues, significant progress has been made in this area. See for example: [SPCM97, TBNF03, TSB*05, SW06, LST09, AHLG*13, RJD13, CBC15, SZK15, ZHK15] One of the reasons that skeletal muscle animations have been more successful is that these simulations tend to be much more forgiving with respect to producing plausible animations. The recently proposed method in [CBE*15] provides a template muscle model that can be quickly and automatically morphed to any facial scan, alleviating much of the investment required to produce such models. Unfortunately, after utilizing this newly proposed method, even along with a high level of user involvement to create better simulation models, we have been hard pressed to devise scenarios where the resulting simulation can produce results usable for high-end production on more than a special case basis.

Even after constructing an anatomically and biomechanically accurate facial model, one has to set the material parameters (stiff-

[†] e-mail: {mdcong|fedkiw}@cs.stanford.edu
[‡] e-mail: kiran.s.bhat@gmail.com

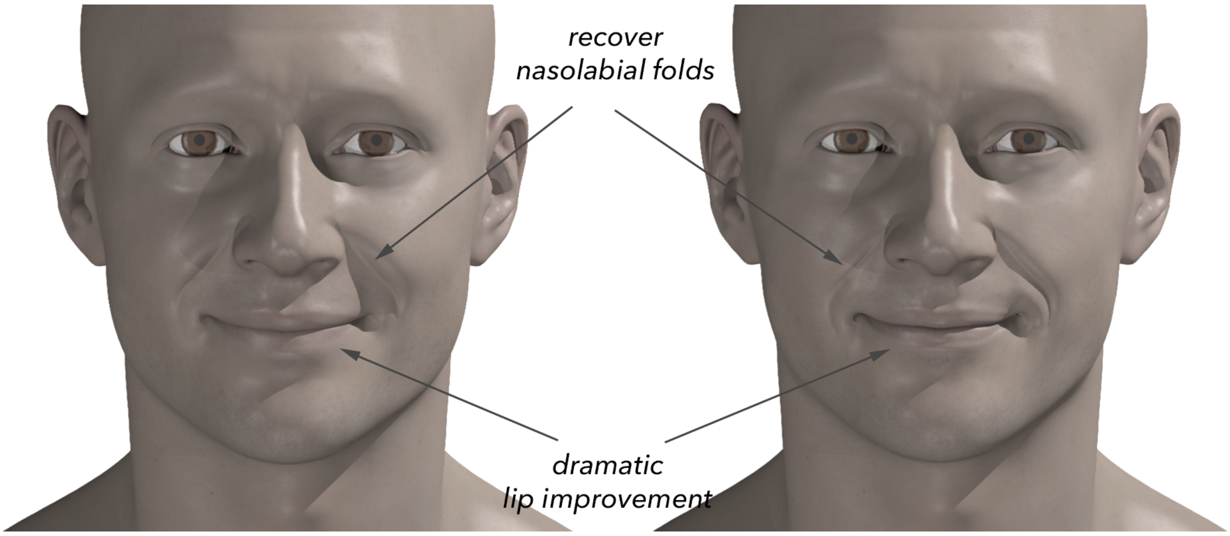

Figure 1: *Left: The smile created by an artist using the blendshape rig. The lips suffer from texture stretching and a lack of volume preservation. Right: The smile generated by our modified muscle tracks. The physically based algorithm produces improved results for the lips. A selective snap technique is used to improve the blendshape by hybridizing it with our physically based results. Here, the nasolabial folds are retained from the blendshape while the mouth and other features come from the simulation.*

ness, incompressibility, etc.) such that realistic results are obtained. It is extremely difficult to get the underlying cranium and jaw collision geometry, thin muscle sheets and attachments, muscle fascia, skin hysteresis, etc. all correct in a manner that allows one to simply actuate muscles and subsequently obtain production quality results. Instead, one has to continuously tinker with elasticity parameters, muscle activations, muscle attachments and shapes, muscle fibers, etc. in order to significantly improve the model on a case-by-case basis. While the state of the art simulation results would have been useful to the movie industry a decade ago, these results still lag behind the modern state of the art production results obtained using enveloping, skinning, and computer vision techniques. Even though in certain scenarios, such as colliding against a creature's tusk to obtain in-betweens, muscle-based simulation techniques can be useful in a production setting, they still lack the quality and directability required to be used as a bread and butter tool in a production environment.

One of the significant advantages of our approach is that it mostly leverages existing algorithms. The key idea is actually rather simple, straightforward, and intuitive once it is seen in action. Instead of tweaking parameters, anatomy, attachment points, etc. in the hopes that the muscle will somehow contract and move in a way that subsequently drives the flesh and then the surface of the face towards a desired configuration, we simply tell the muscles where to go creating a blendshape system for muscles that drives the underlying simulation framework in a direction that achieves the desired result. This bears some similarity to the method proposed by [SNF05] that solved an inverse problem to push the face as close as possible to a desired shape. However, we have seen that the vast majority of desirable shapes produced in high-end production are unobtainable using this method, even with extensive tweaking of muscle attachments and shapes, bone geometry, material parameters, etc. Although one could expand the parameters of [SNF05] to include elasticity parameters, cranium and jaw geometry, muscle positions, etc., this becomes an unwieldy inverse problem. Instead, we significantly simplify the inverse problem by using a blendshape muscle system to unhinge from physical parameters and restrictions while deforming the muscles in the desired directions. Then, these muscle tracks are used as drivers for the anatomically and biomechanically motivated simulation. Thus, we still retain the richness of simulation while gaining the directability of a blendshape system.

2. Related Work

High-quality facial geometry and performance data can be acquired using techniques such as [BBB*10, BHB*11]. The resulting facial performance can be augmented with additional wrinkles [BBA*07], eye geometry [BBN*14], and/or eyelid detail [BBK*15]. The captured data can then be used to create corresponding facial expressions on other meshes using a variety of methods. One such method is given by [NN01] which uses correspondences to create vertex motion vectors to transfer animation to a target mesh. Another such method is deformation transfer [SP04]. More recently, [XCLT14] proposed a method for editing a performance and transferring large and fine-scale details to a target mesh. Many of these methods require a set of correspondences between the source and the target face meshes which can either be manually-specified or automatically generated using an algorithm such as iterative closest point [BM92] or non-rigid iterative closest point [ARV07].

The acquired facial geometry is often used to build a blendshape facial rig [LAR*14] which can be highly complex in a production environment [SL14]. The blendshape rig can be edited by an artist to achieve a desired pose [LA10]. There has also been some work on incorporating physics into blendshape rigs. For example, [MWF*11] proposed a technique for improving shape blending by interpolating the spring rest lengths of mass-spring systems on

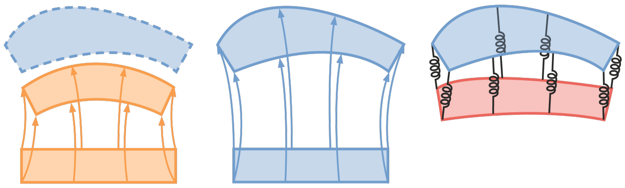

Figure 2: *Left: A muscle moves from its neutral pose to a deformed state (orange). The tracks are shown as arrows. An artist modified muscle position is shown in blue. Middle: New tracks are generated going from the neutral to the modified muscle position. Right: Zero-length springs are used to drag the simulated muscle (red) towards the kinematically animated muscle on tracks.*

the source and target shapes. Once such a blendshape rig is built, it can then be transferred to an arbitrary target mesh using [LWP10] which optimizes the blendshape rig and its weights to match a series of training shapes. The method proposed in [SSW*10] can be used to transfer over underlying muscle and joint data as well.

3. Simulation Framework

Given a high resolution face mesh in the static rest pose, we construct a simulatable anatomical face model by morphing an anatomically and biomechanically accurate muscle and flesh template model following the approach in [CBE*15]. Some hand adjustments, such as manual curve drawing and fine-tuning meshing parameters, are applied in conjunction with their approach in order to obtain the highest quality model possible with the goal of using it in a production environment. The resulting simulatable face model contains a cranium and jaw represented as both an explicit triangulated surface and an implicit level set volume. The jaw joint is defined by the endpoints of the left and right condyle sliding tracks and can rotate around the axis that connects the jaw along those two tracks. The soft tissue is represented by a single conforming tetrahedralized flesh mesh and 49 B-spline solid facial muscles. The facial muscles are embedded into the tetrahedralized flesh mesh by calculating the fraction of overlap between each muscle and each tetrahedron of the flesh mesh and storing the fraction locally in the tetrahedron. Each tetrahedron with a nonzero overlap is then assigned a fiber direction per muscle based on the overlapping portion of the muscle B-spline solid. This is accomplished by first uniformly distributing a number of sample points in each tetrahedron with a nonzero overlap. Then, for each tetrahedron, we compute the sample points interior to the muscle B-spline solid and assign each of these sample points the direction of the muscle fiber field at the sample point's location. The per-muscle fiber direction for the tetrahedron is then obtained by averaging the sampled fiber directions and normalizing the result. Dirichlet boundary conditions are used to specify cranium and jaw attachments of the muscles and flesh.

The tetrahedralized flesh mesh is simulated using a quasi-incompressible Mooney-Rivlin constitutive model with an anisotropic muscle response as in [TBNF03, TSB*05]. In this constitutive model, the magnitude of the force applied by an activated muscle in a given mesh configuration is weighted by a scalar muscle activation value. For a positive muscle activation value, each tetrahedron with a nonzero overlap with the muscle contracts along its assigned fiber direction for that muscle. Thus, for a set of muscles, we have a vector of muscle activations that controls the forces applied by the muscles to the tetrahedralized flesh mesh. Given a set of control parameters containing both the muscle activations and the cranium and jaw configuration, we obtain a steady state deformation of the tetrahedralized flesh mesh by simulating with the quasistatic framework of [TSIF05]. Animations are obtained by varying the control parameters over time. We can include inertial effects, which are neglected in the quasistatic solver, via a full dynamic simulation with the same control parameters (see e.g. [SSIF07]).

The original high resolution face mesh is embedded in the neutral pose of the tetrahedralized flesh mesh by finding the surface triangle of the tetrahedralized flesh mesh closest to each vertex of the high resolution face mesh and computing the barycentric weights for the vertex along with a displacement if it does not lie exactly on the surface of the tetrahedralized flesh mesh. In cases where the lips or eyelids are arbitrarily close together, it is often useful to compute this embedding in a deformed configuration where the lips are apart and the eyelids are open. Using this embedding allows us to deform the high resolution face mesh during a simulation by interpolating vertex positions from the deformed tetrahedralized flesh mesh.

Given a pre-existing blendshape rig for a high resolution face mesh obtained via sculpting or capture, we compute the control parameters that best match each blendshape using the inverse activations solver of [SNF05]. The target landmark positions used as input are obtained by selecting 480 points on the high-resolution face mesh in the neutral pose, computing their barycentric coordinates, and interpolating target positions from the deformed blendshape pose of the high-resolution mesh. We denote the muscle activations obtained from this inverse activations solve as the *raw activations* and the resulting shape as the *raw activations pose* throughout the paper. Then, to determine in-betweens for this pose, we interpolate between the control parameters of the neutral pose and the control parameters of the raw activations pose. Since we interpolate control parameters instead of surface mesh positions, every vertex of the tetrahedralized flesh mesh and the high resolution face mesh follows a nonlinear path. Since the muscles are located inside the tetrahedralized flesh mesh, we can also obtain a set of nonlinear muscle tracks for each B-spline solid control point using barycentric interpolation. Thus, for each expression, we are able to generate a set of muscle tracks that each muscle follows from the neutral pose to the raw activations pose. See Figure 2 (Left).

4. Directability

The raw activations pose can be art-directed by editing the nonlinear tracks for the B-spline solid control points. To do this, we edit the last point on the nonlinear track for every control point effectively resculpting the muscle shape. Then, assuming that the shape of the muscle in the neutral pose does not change, we modify the intermediate points on the tracks by linearly interpolating the displacements of the endpoints throughout the tracks. In general, one can edit any point along the tracks allowing for the art-direction of

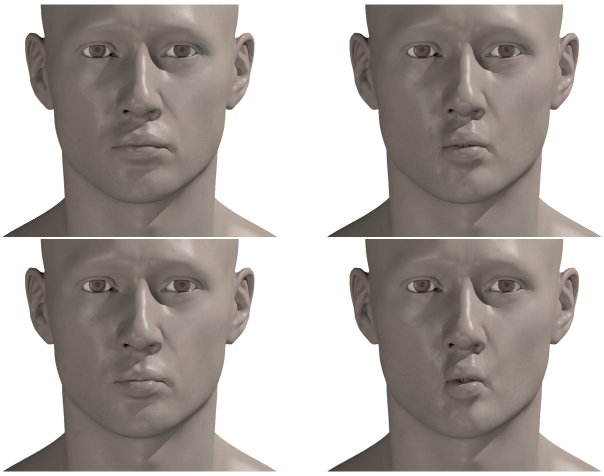

Figure 3: *Upper Left: We fully activate the incisivus labii superioris and incisivus labii inferioris muscles to pull the corners of the mouth towards the medial plane of the face. Even though the muscles were carefully adjusted based on anatomical references, the resulting pose only exhibits a small deformation. Upper Right: We create muscle tracks from the aforementioned simulation and move the endpoints of the tracks closer to the medial plane of the face. Simulating with these modified tracks pulls the corners of the mouth closer to the middle of the face causing the upper and lower lip to buckle outwards exposing the teeth. Bottom Right: We create even more deformed muscle tracks. Bottom Left: We use the inverse solver to find the muscle activations that best match the pose obtained from the tracks. The inverse solver fails to obtain the pose.*

in-betweens while preserving the initial and final poses. See Figure 2 (Left and Middle).

Editing the raw activations pose and/or modifying the tracks causes the muscle to be shortened or lengthened. In order to obtain the proper tension in the muscle, we compute new activation values. To do this, we rely on a set of activation vs. length curves that are precomputed for each muscle. These are precomputed, starting from the neutral shape, by activating a single muscle incrementally from 0 to 1 holding all other control parameters constant. After each activation increment, we measure and record the length of the curve that goes down the centerline of the muscle in the parameter space of the plane transverse to the muscle. This length vs. activation data is inverted to build an activation vs. length function, which can be edited if desired to modify the relationship between activation and muscle length. Then, given modified muscle tracks, we compute the length of our centerline curve to obtain the appropriate muscle activation that adds the proper tension to the muscle as it follows the tracks. We have found that using the proper muscle tension yields significantly improved results, since the flesh shortens properly instead of bunching up.

Given the newly created tracks and modified activation values, we compute a path for the muscle that is usable as a driver for the simulation. This is accomplished by identifying the vertices of the flesh mesh that lie within the neutral pose of the B-spline solid muscle and moving them along the tracks to the final target pose. In order to do this, we evaluate the B-spline solid muscle in the neutral representation to obtain a tetrahedralized volume representation of the muscle that can be used to barycentrically interpolate locations of vertices belonging to the flesh mesh. Then, periodically along the tracks, we simply re-evaluate the B-spline solid muscle, compute its tetrahedra, and use the barycentric weights from the neutral pose to interpolate new locations for the vertices of the flesh mesh. This gives us usable tracks for every vertex of the flesh mesh that is contained in a muscle.

During simulation, the target location for each vertex of the flesh mesh inside a muscle is kinematically animated to move along its predetermined track. Then, we connect zero-length springs between the vertices of the flesh mesh and their kinematically animated targets in order to drag the vertices associated with our B-spline solid muscles along our modified muscle tracks (see Figure 2, Right). Simultaneously, we use the modified muscle activation values in order to add the proper tension to the muscle. Although this may at first seem forced and would cause obvious visual artifacts if applied to the surface of the face, the resulting simulated flesh mesh does an excellent job of smoothing the results. The ability to naively modify muscle tracks while still obtaining a plausible result for the face surface is rather compelling, especially when realizing that the alternative requires an artist to carefully sculpt every triangle on the face in a plausible manner. Figure 3 demonstrates our method being used to art-direct a lip pucker expression. See also Figure 4.

It is significantly quicker to obtain high-quality results by editing the muscle tracks and resimulating instead of editing the face surface. This is because the muscle tracks have significantly fewer vertices and the simulation enforces constraints such as volume preservation that are difficult to explicitly sculpt on the face surface. In fact, the edits to the muscle tracks in Figure 3 and Figure 4 were made by a computer science graduate student (with very little experience in 3D modeling) in only a few minutes. In addition, we have found that simulations using modified muscle tracks incur a negligible overhead over simulations using only the raw activations.

5. Expression Targeting

In addition to providing directability, the muscle tracks can also be used to target expressions represented by either an artist sculpted shape or a facial scan. Although one could accomplish this by manually adjusting the muscle tracks for every muscle, such a process can quickly become tedious and time consuming due the complex interactions of different muscles involved in forming certain expressions. Thus, we propose an iterative algorithm that automatically computes the muscle tracks that drive the simulation to match an expression target. In order to do this, we define a *snap* as the per-vertex displacement of the high resolution face mesh that deforms the high resolution face mesh to match an expression target. If the expression target is topologically identical to the high resolution face mesh, the snap is simply the displacement of each vertex of the high resolution face mesh from its current position to its position on the expression target. If the expression target has a different topology, we first build a correspondence between the high resolution face mesh and the expression target using a method such as [BM92, SP04, CBE*15].

As outlined in Section 3, we first use the inverse activations

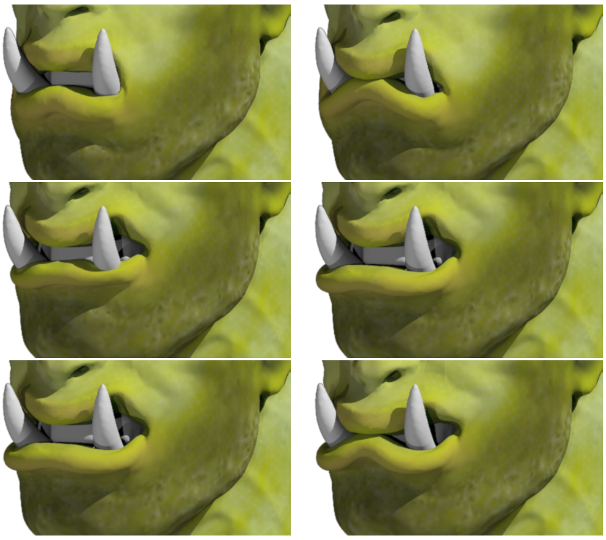

Figure 4: *Upper Left: Collisions between the Orc's mouth and the tusks result in the mouth being open in the neutral pose. In order to close the mouth, we fully activate the orbicularis oris; however, this is insufficient to close the mouth completely. Upper Right: Thus, we edit the shape of the orbicularis oris at the end of the tracks to form a mouth closed configuration. Middle Left: Collisions between the lower lip and the tusks are resolved using a penalty based formulation, which can often be problematic especially during quasistatic simulation. Notice the poor deformation of the lip near the tusk. Middle Right: We resculpt the muscle tracks for the lower orbicularis oris to maintain some contact with the tusks without overly stressing the collision penalty forces. Notice the more aesthetic collision. Bottom Left: The mouth does not stay closed as the orc smiles. Bottom Right: We edit the muscle tracks to maintain a closed mouth throughout the simulation.*

solver to compute the raw activation pose and tracks for every muscle in our model. Recall, the high resolution face mesh is deformed along with the tetrahedralized flesh mesh. Even the best simulation models will struggle to match interesting target poses. Thus, we compute the snap from the deformed high resolution face mesh to the expression target. Using these snap displacements as boundary conditions, we utilize the volumetric morphing algorithm outlined in [AHLG*13, CBE*15] to morph the endpoints of the muscle tracks for every muscle. As a consequence of using penalty forces to resolve self-collisions, the deformed high resolution face mesh may exhibit self-intersections. Intersections between the upper and lower lips are especially problematic when the lips are separating because this causes the endpoints of the muscle tracks for muscles in the upper lip to be influenced by the displacements of the lower lip and vice versa. We avoid this by solving the Poisson equation outlined in the aforementioned volumetric morphing algorithm on the deformed tetrahedralized flesh mesh instead of a background grid to obtain per-vertex displacements. Dirichlet boundary conditions are enforced on the outer surface of the tetrahedralized flesh mesh with values obtained by interpolating the snap displacements from the deformed high resolution face mesh. The displacements for the endpoints of the muscle tracks for every muscle are obtained by barycentrically interpolating from vertices of the deformed tetrahedralized flesh mesh. The intermediate points of the muscle tracks for every muscle in our model are also morphed by interpolating the displacements at the track endpoints as per Section 4. Then we re-simulate using the modified muscle tracks to drive every muscle in order to obtain a pose that more closely matches the blendshape and reduces the norm of the snap displacements. This entire process of simulating, computing the snap, morphing to obtain new tracks, and re-simulating, etc. can be iterated to further reduce the norm of the snap displacements towards convergence. Figure 5 demonstrates our approach for a lip funneler shape from a production blendshape rig. We have found that this iterative approach converges to a pose that matches the blendshape much more closely in just a few iterations.

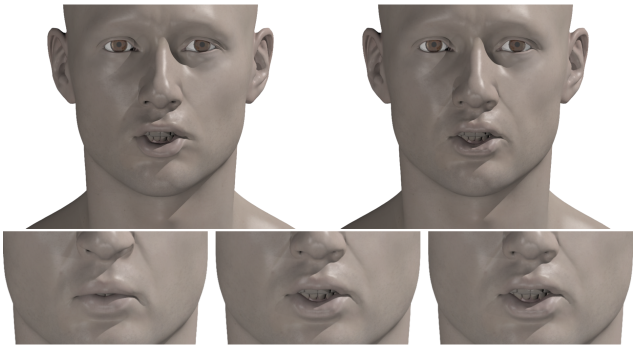

Figure 5: *Upper Left: Blendshape target. Upper Right: Converged result. Bottom Row: Raw activations followed by successive iterations of snap, morph, and simulate.*

Note that the deformed high resolution mesh in the converged pose will typically not match the blendshape exactly since it is subject to the physical constraints of the face model. This is quite helpful in many cases because nonphysical volume gains or losses that are extremely common in linear blendshapes can be filtered out while maintaining the defining characteristics of the expression. However, if one does wish to exactly match the blendshape, we simply compute a *final snap* that displaces the high resolution face mesh from its converged pose to the blendshape target. This final snap can be added to the muscle simulation as a perturbation blendshape, linearly interpolating its effect from 0 to 1 on the way to the pose.

Although one could use the final snap to exactly match a pose, it is often more useful to selectively blend between the converged simulation shape and the results of the final snap allowing an artist, for example, to accentuate wrinkles and folds that are sculpted on the blendshape while preserving desirable effects such as volume conservation obtained via simulation. In order to facilitate this, we utilize an interactive editor. First, the artist draws lines on the blendshape in order to select features. The artist also paints triangles in order to specify a smooth falloff region. Then, this is used as input into a Poisson equation solver over the painted area with the discretization outlined in [CBE*15]. Dirichlet boundary conditions are used to set a value of 1 on the artist-drawn feature lines and a value of 0 on the boundary of the painted triangular region. Furthermore, an editable spline curve is used to scale the per-vertex area weight in order to obtain new weights which enable better control of falloff and blending. The final snap is then modified by the per-

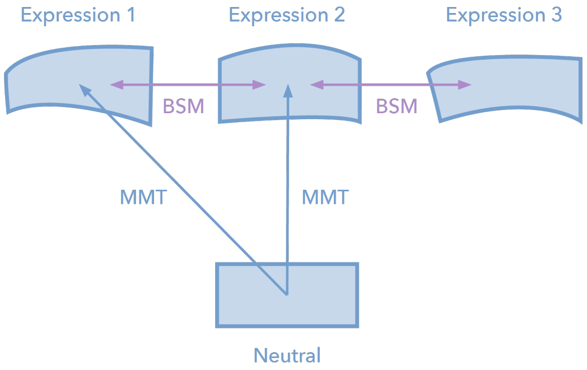

Figure 6: *Given modified muscle tracks (MMT) from the Neutral to Expression 1 and from the Neutral to Expression 2, we create in-betweens between Expression 1 and 2 using blendshape muscles (BSM). This can be done for every pair of expressions.*

vertex interpolation weights in order to selectively use or discard its contribution. See Figure 1.

6. Cranium and Jaw

In addition to muscle activations, the inverse activations solver also yields transformations for both the cranium and jaw. Typically, these are sufficient. However, it may be difficult to match some of the more extreme and/or potentially unphysical poses given the jaw's physical joint limits. In these scenarios, we ignore the Dirichlet boundary conditions specifying flesh attachments to the cranium and jaw allowing the subsequently computed muscle tracks to separate the flesh from the cranium and jaw in an inconsistent manner. Then, in a second pass, we use the cranium and jaw attachments as landmarks to optimize the cranium placement and jaw articulation in a manner that minimizes the displacements between the flesh and the flesh's attachments to these rigid bones. If desired, one could relax the anatomical constraints of the jaw joint (e.g. maximum opening angle) during this step in order to further minimize these displacements. In fact, using a sequence of poses, one can use such a process in order to better determine the proper anatomical constraints for the jaw joint. Finally, we reattach the flesh to the cranium and jaw and resimulate the flesh volume using the resulting cranium and jaw frames along with the modified muscle tracks. If desired, one could iterate this process by re-running the expression targeting algorithm, recomputing the cranium and jaw transforms, and resimulating to further refine the converged pose.

7. Blendshape Muscles

So far, we have discussed how to create and refine a set of nonlinear modified muscle tracks that drives every muscle of the face from the neutral pose towards a target expression pose. In addition, we have discussed ways that our method can be used to selectively improve/modify the target pose. Next we consider dealing with two such target poses and the transition from one to the other. Although one might attempt this transition by varying the strength of springs that attach to the muscles from both target poses, we instead apply the standard blendshape framework to the underlying muscles creating a blendshape muscle system. Whereas applying blendshapes to the high-resolution face mesh results in kinematically driven transitions, using the same kinematic framework on the muscle shapes does not incur the same drawbacks because the simulation dynamics cleanse/soften the effects of kinematics.

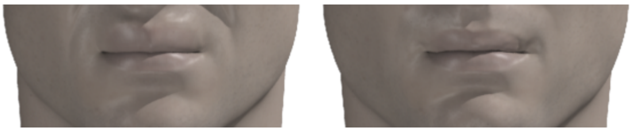

Figure 7: *Left: The upper lip is incorrectly stretched in the artist's blendshape performance. Right: Our blendshape muscles produce a highly improved result. See the video for the rest of the performance and our corresponding modified muscle tracks.*

Given the muscle shapes corresponding to two expression targets, we linearly interpolate between the muscle shapes of the initial expression target and the muscle shapes of final expression target to obtain a set of linear muscle tracks for every muscle in the face. See Figure 6. Then, we simulate using these muscle tracks from the initial expression target to the final expression target in order to obtain in-betweens. Note that even though our muscle tracks are linear, the resulting motion obtained via simulation is nonlinear, exhibiting effects such as collision, contact, and volume preservation. In contrast, linear blendshapes are limited to linear motion between two target expressions. Note that the initial and final simulation poses will not necessarily match the initial and final expression targets respectively. While we have already generated a final snap for the initial expression target, we also compute a new final snap that accounts for the difference between the final state of the simulation mesh and the final expression target. This final snap for the transition is different from the final snap obtained when simulating from the neutral pose to the final expression target since the path of the muscle is different. Now, if we wish to have the simulation exactly hit both expression targets, we simply add a linear interpolation between the final snaps for the initial and final expression targets to the simulation results as a blendshape, similar to what was done in Section 5 when targeting a single expression. Similarly, we may selectively edit either one of these final snaps.

In order to track an entire performance, we simply treat every pair of frames as outlined above, creating a blendshape muscle performance for every muscle and subsequently using this performance to drive the simulation. See Figure 6. Each frame and/or each transition can be art-directed and/or cleaned as outlined in Sections 4 and 5. One can also re-time the performance of the blendshape muscles in order to additionally edit the performance. Figure 8 shows in-betweens obtained using our blendshape muscles. In addition, the last example in our video shows a sixteen second artist-directed blendshape performance taken from a facial animation stress test used in an industry setting which is subsequently captured by modified muscle tracks using our blendshape muscle system. Figure 7 shows a frame from the video.

8. Retargeting

Given a high resolution face mesh for a target creature as input, we build a corresponding simulatable anatomical face model in which

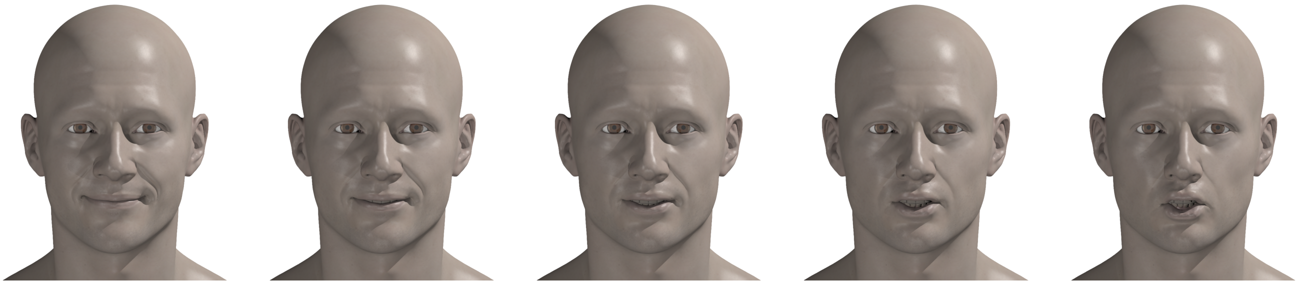

Figure 8: *Here, we use blendshape muscles in order to drive our simulation from the smile pose (left) to the funneler pose (right).*

the given high resolution face mesh is embedded as discussed in Section 3. The easiest way to retarget the performance from the actor to the target creature is to simulate the target creature using the raw activations of the actor in order to obtain corresponding raw activation poses for the target creature. However, since the raw activations already struggle to match interesting poses on the actor, the resulting performance on the target creature will also struggle to achieve interesting poses. Thus, we also transfer the modified muscle tracks (computed as per Section 5) that drive the actor from the neutral pose towards an expression target. This is accomplished using deformation transfer [SP04] which computes the deformation gradient for each source triangle and solves a least squares problem to map this deformation gradient to every corresponding target triangle subject to the constraint that vertices shared between the triangles are transformed to the same location. The deformation transfer algorithm was optimized using the surface-based deformation gradient formulation of [BPGK06] which significantly reduces the size of the least squares problem. Although applying deformation transfer to the high resolution face mesh of an actor in order to transfer expression targets to a target creature can result in visual artifacts and often requires hand tuning of the correspondences in order to improve the results, applying the same framework to the muscle shapes does not incur these drawbacks because the simulation takes into account differences in internal anatomy, volume preservation, etc. Moreover, the muscles shapes of the actor and target creature are both mapped from the same template and therefore have a straightforward one-to-one correspondence.

As noted in Section 4, we may evaluate the B-spline solid muscle in order to compute its tetrahedra given the control points. The surface of this tetrahedralized volume can be used to render the muscles as shown in Figure 10. We also use this triangulated surface in order to transfer the muscle tracks from the actor to the target creature. In order to obtain source displacements on the actor, we compute the displacement of the triangulated muscle surface from the raw activations pose to the position of the muscle at the end of its modified tracks. These displacements are used as inputs into the deformation transfer algorithm in order to obtain a corresponding set of displacements for the triangulated surface of the target creature's muscle in its raw activations pose. These new displacements on the target creatures' muscle triangulated surface are used along with the morphing algorithm outlined in [AHLG*13, CBE*15] to morph all interior muscle tetrahedra as well. Then, the intermediate points of the muscle tracks are also morphed by interpolating displacements from the track endpoints. Finally, the creature is simulated using these modified muscle tracks to obtain a new retargeted pose.

If desired, the final snap can also be transferred from the actor to the creature via deformation transfer. Recall, that the method of [CBE*15] first snaps the surface mesh of the template to the high resolution surface mesh of the target model. Thus, this step puts templates with the same topology on both the actor's high resolution face mesh and the target creature's high resolution face mesh. Thus, we simply project the high resolution face mesh of the actor onto its nearby deformed template mesh and then use the resulting barycentric weights in order to redraw the actor's mesh on the template mesh that was snapped to the target creature. This places the actor's mesh and the creature's mesh in close proximity so that we can use the standard version of the deformation transfer algorithm of [SP04] in order to transfer (selectively edited) final snaps to the creature. Note that although one could simulate transferred raw activations on the creature and apply final snaps to these, the results from the raw activations are so poor that they cannot be used to create better blendshape poses. Figure 9 shows retargeted performances using our modified muscle tracks and selective snaps for cleaning blendshape poses.

Now that we have modified muscle tracks for the creature taking it from the neutral pose to a target expression pose, as well as final snaps, we can apply the blendshape muscle framework outlined in Section 7 to transition between two such poses on the creature allowing us to obtain a blendshape muscle performance for the creature. Of course, the blendshape muscle performance can also be art-directed, cleaned, and re-timed. Several examples of this are shown in the video.

Although a straightforward application of deformation transfer can be used to transfer our modified muscle tracks to a new creature, one has to exercise caution if one desires net translational or rotational effects that are not typically considered in deformation transfer algorithms. For translation, one can compute the net translation that the modified muscle tracks make to the raw activations pose and transfer this net translation to the creature properly rescaled based on scale/size differences. The situation can be exacerbated when the raw activations perform poorly. In fact, if the raw activations perform poorly enough, one can instead use a linear blendshape muscle to connect the various expressions with the neutral shape ignoring the raw activations pose altogether. However in such instances, it is still desirable to obtain jaw motion. This is

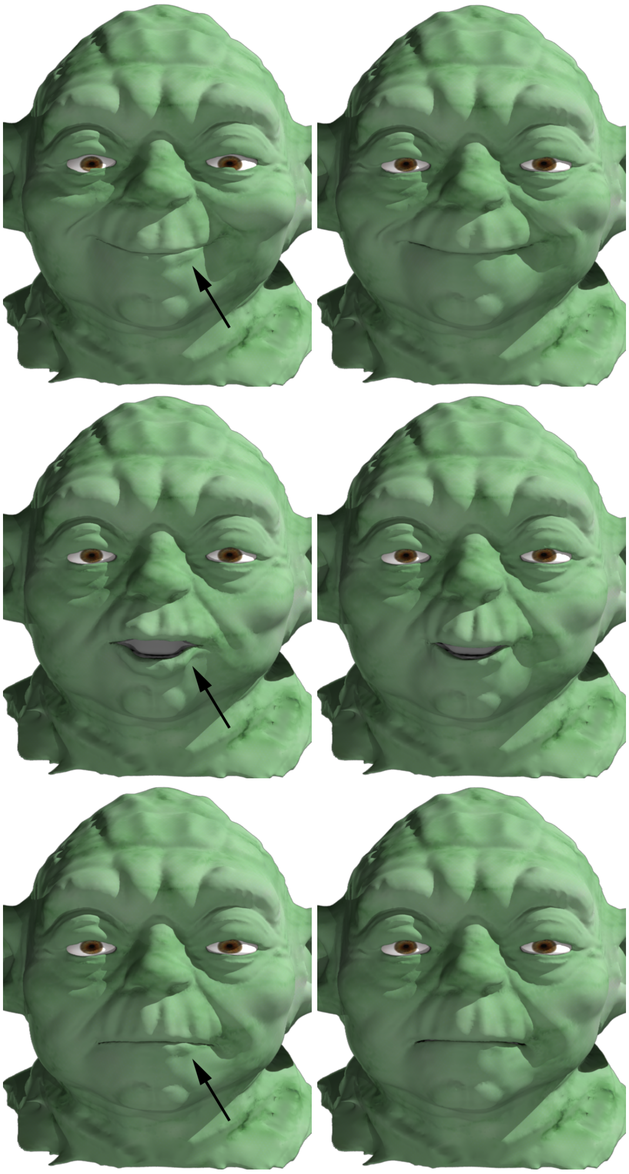

Figure 9: *For the sake of comparison, we show full deformation transfer results (left) as compared to our modified muscle tracks (right) for a smile (top), funneler (middle), and grimace (bottom). Note that there is ghosting of a bottom lip from the actor in the deformation transfer results. It is difficult to craft correspondences for every region of the face in order to avoid such issues. The video also shows that deformation transfer gives incorrect movements in the eyes and forehead regions, which are not present utilizing our method.*

accomplished by parenting all the muscles, or sub-muscles, that are near the jaw to the jaw so that they move rigidly in the jaw's frame. Then, we treat this rigidly rotated pose as a jaw deformed neutral pose in order to properly capture rotations and translations that would otherwise be missed using blendshape muscles starting from the unrotated jaw neutral pose. The last example in our video shows a sixteen second artist-directed blendshape performance retargeted to a new creature. We also show the modified muscle tracks for our captured version of the source performance as well as for the target creature. See Figure 10.

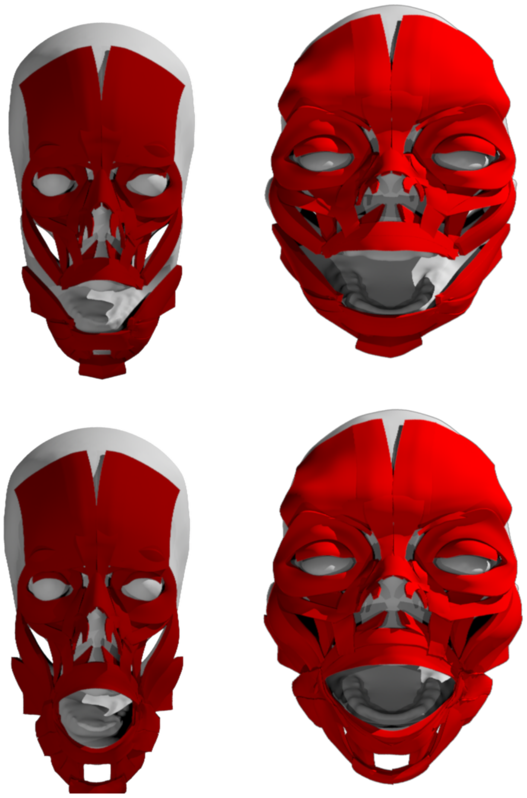

Figure 10: *Top: Jaw deformed neutral poses for an artist-directed performance and a target creature. Bottom Left: The modified muscle tracks obtained using our blendshape muscle system. Bottom Right: The same frame retargeted to a creature.*

9. Discussion and Conclusions

Albeit our new paradigm for muscle simulation greatly improves upon the state of the art in muscle simulation, admittedly, one may at first feel that the improvements our system makes to a state of the art blendshape system are comparably smaller. However, the problems with the uncanny valley stem from the fact that it is difficult to ascertain exactly what changes need to be made in order to climb out of it. When our system identifies previously unidentified issues with state of the art blend shapes or blendshape systems and additionally allows us to correct them, we feel that our contribution, as well as the potential for future contributions in this vein, is rather high. In fact, many experts in this area, who are working on classical and state of the art approaches to facial animation, have expressed significant enthusiasm upon seeing our results.

Acknowledgements

Research supported in part by ONR N00014-13-1-0346, ONR N00014-11-1-0707, ONR N-00014-11-1-0027, and ARL AHPCRC W911NF-07-0027. Computing resources were provided in part by ONR 00014-05-1-0479. We would like to thank Jane E and Michael Bao for help with the paper and the video. We would also like to thank Brice Criswell, Sean Comer, John Doublestein, Chris Havreberg, Scott Jones, Michael Koperwas, and Cary Phillips for helpful discussions as well as Industrial Light + Magic for assets. M.C. was supported in part by an NDSEGF.

References

[AHLG*13] Ali-Hamadi D., Liu T., Gilles B., Kavan L., Faure F., Palombi O., Cani M.-P.: Anatomy transfer. In *ACM SIGGRAPH Asia 2013 papers* (2013), SIGGRAPH ASIA '13, pp. 188:1–188:8.

[ARV07] Amberg B., Romdhani S., Vetter T.: Optimal step nonrigid ICP algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* (June 2007), pp. 1–8.

[BBA*07] Bickel B., Botsch M., Angst R., Matusik W., Otaduy M., Pfister H., Gross M.: Multi-scale capture of facial geometry and motion. In *ACM SIGGRAPH 2007 Papers* (2007), SIGGRAPH '07.

[BBB*10] Beeler T., Bickel B., Beardsley P., Sumner B., Gross M.: High-quality single-shot capture of facial geometry. *ACM Trans. Graph. (SIGGRAPH Proc.) 29*, 3 (2010), 40:1–40:9.

[BBK*15] Bermano A., Beeler T., Kozlov Y., Bradley D., Bickel B., Gross M.: Detailed spatio-temporal reconstruction of eyelids. *ACM Trans. Graph. 34*, 4 (2015), 44:1–44:11.

[BBN*14] Bérard P., Bradley D., Nitti M., Beeler T., Gross M.: High-quality capture of eyes. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 223:1–223:12.

[BGY*13] Bhat K. S., Goldenthal R., Ye Y., Mallet R., Koperwas M.: High fidelity facial animation capture and retargeting with contours. In *ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2013), pp. 7–14.

[BHB*11] Beeler T., Hahn F., Bradley D., Bickel B., Beardsley P., Gotsman C., Sumner R. W., Gross M.: High-quality passive facial performance capture using anchor frames. *ACM Trans. Graph. (SIGGRAPH Proc.) 30*, 4 (2011), 75:1–75:10.

[BL05] Borshukov G., Lewis J. P.: Realistic human face rendering for the matrix reloaded. In *ACM Siggraph 2005 Courses* (2005), ACM, p. 13.

[BM92] Besl P. J., McKay N. D.: Method for registration of 3-D shapes. In *Robotics-DL tentative* (1992), International Society for Optics and Photonics, pp. 586–606.

[BPGK06] Botsch M., Pauly M., Gross M., Kobbelt L.: Deformation transfer for detail-preserving surface editing. *Vision, Modeling, and Visualization* (2006), 357–364.

[CBC15] Comer S., Buck J., Criswell B.: Under the scalpel - ILM's digital flesh workflows. In *ACM SIGGRAPH 2015 Talks* (2015), ACM.

[CBE*15] Cong M., Bao M., E J. L., Bhat K. S., Fedkiw R.: Fully automatic generation of anatomical face simulation models. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2015), pp. 175–183.

[EMF02] Enright D., Marschner S., Fedkiw R.: Animation and rendering of complex water surfaces. *ACM Trans. Graph. (SIGGRAPH Proc.) 21*, 3 (2002), 736–744.

[FJA*15] Fyffe G., Jones A., Alexander O., Ichikari R., Debevec P.: Driving high-resolution facial scans with video performance capture. *ACM TOG 34*, 1 (2015).

[FSJ01] Fedkiw R., Stam J., Jensen H.: Visual simulation of smoke. In *Proc. of ACM SIGGRAPH 2001* (2001), pp. 15–22.

[JMLH01] Jensen H. W., Marschner S., Levoy M., Hanrahan P.: A practical model for subsurface light transport. In *Proc. of SIGGRAPH 2001* (2001), pp. 511–518.

[LA10] Lewis J., Anjyo K.-i.: Direct manipulation blendshapes. *IEEE Computer Graphics and Applications*, 4 (2010), 42–50.

[LAR*14] Lewis J. P., Anjyo K., Rhee T., Zhang M., Pighin F., Deng Z.: Practice and Theory of Blendshape Facial Models. In *Eurographics 2014 - State of the Art Reports* (2014), Lefebvre S., Spagnuolo M., (Eds.), The Eurographics Association.

[LST09] Lee S.-H., Sifakis E., Terzopoulos D.: Comprehensive biomechanical modeling and simulation of the upper body. *ACM Trans. Graph. 28* (2009), 99:1–99:17.

[LWP10] Li H., Weise T., Pauly M.: Example-based facial rigging. *ACM Trans. Graph. (SIGGRAPH Proc.) 29* (2010), 32:1–32:6.

[MWF*11] Ma W.-C., Wang Y.-H., Fyffe G., Barbič J., Chen B.-Y., Debevec P.: A blendshape model that incorporates physical interaction. In *SIGGRAPH Asia 2011 Posters* (2011), ACM.

[NN01] Noh J., Neumann U.: Expression cloning. In *Proc. of ACM SIGGRAPH* (2001), Fiume E., (Ed.), ACM Press, pp. 277–288.

[RJD13] Rose R., Jutan M., Doublestein J.: Blockparty 2: Visual procedural rigging for film, tv, and games. In *ACM SIGGRAPH 2013 Talks* (2013), ACM.

[SL14] Seo J., Lewis J.: Developing interactive facial rigs in production environment. In *ACM SIGGRAPH 2014 Talks* (2014), ACM, p. 36.

[SNF05] Sifakis E., Neverov I., Fedkiw R.: Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph. (SIGGRAPH Proc.) 24*, 3 (2005).

[SP04] Sumner R., Popović J.: Deformation transfer for triangle meshes. In *ACM Trans. on Graph. (Proc. ACM SIGGRAPH)* (2004), vol. 23, pp. 399 – 405.

[SPCM97] Scheepers F., Parent R. E., Carlson W. E., May S. F.: Anatomy-based modeling of the human musculature. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), ACM Press/Addison-Wesley Publishing Co., pp. 163–172.

[SSIF07] Sifakis E., Shinar T., Irving G., Fedkiw R.: Hybrid simulation of deformable solids. In *Proc. of ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2007), pp. 81–90.

[SSW*10] Seo J., Seol Y., Wi D., Kim Y., Noh J.: Rigging transfer. *Computer Animation and Virtual Worlds 21*, 3-4 (2010), 375–386.

[SW06] Smith J., White J.: Blockparty: Modular rigging encoded in a geometric volume. In *ACM SIGGRAPH 2006 Sketches* (2006), ACM.

[SZK15] Saito S., Zhou Z.-Y., Kavan L.: Computational bodybuilding: Anatomically-based modeling of human bodies. *ACM Trans. Graph. 34* (2015), 41:1–41:12.

[TBNF03] Teran J., Blemker S., Ng V., Fedkiw R.: Finite volume methods for the simulation of skeletal muscle. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2003), pp. 68–74.

[TSB*05] Teran J., Sifakis E., Blemker S. S., Ng-Thow-Hing V., Lau C., Fedkiw R.: Creating and simulating skeletal muscle from the visible human data set. *IEEE Trans. on Vis. and Comput. Graph. 11*, 3 (2005), 317–328.

[TSIF05] Teran J., Sifakis E., Irving G., Fedkiw R.: Robust quasistatic finite elements and flesh simulation. *Proc. of the 2005 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2005), 181–190.

[vdPJD*14] 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), ACM, p. 1.

[XCLT14] Xu F., Chai J., Liu Y., Tong X.: Controllable high-fidelity facial performance transfer. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 42.

[ZHK15] Zhu L., Hu X., Kavan L.: Adaptable anatomical models for realistic bone motion reconstruction. *Comput. Graph. Forum 34* (2015), 459–471.