Physically-based Muscles and Fibers Modeling from Superficial Patches

F. Turchet[1,2] O. Fryazinov[2] and S. C. Schvartzman[1]

[1]MPC London ,[2]Bournemouth University

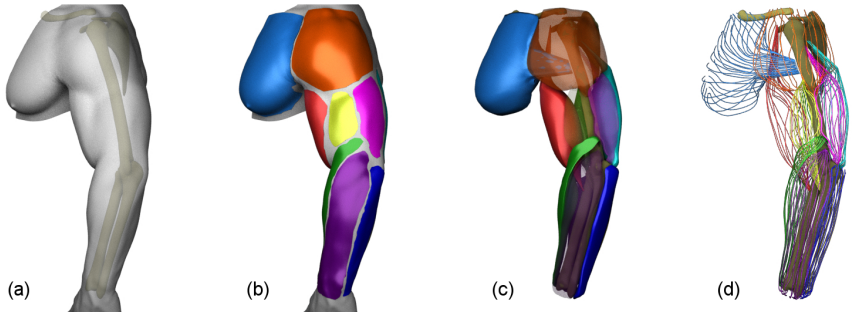

Figure 1: *Overview of our method. Input skin mesh and skeleton (a); sketched patches of superficial muscles (b); inflated output muscles (opacity used to show underlying structures) (c); associated fibers (d).*

Abstract

We propose a novel approach for the generation of volumetric muscle primitives and their associated fiber field, suitable for simulation in computer animation. Muscles are notoriously difficult to sculpt because of their complex shapes and fiber architecture, therefore often requiring trained artists to render anatomical details. Moreover, physics simulation requires these geometries to be modeled in an intersection-free rest state and to have a spatially-varying fiber field to support contraction with anisotropic material models. Inspired by the principles of computational design, we satisfy these requirements by generating muscle primitives automatically, complete with tendons and fiber fields, using physics based simulation of inflatable 3D patches which are user-defined on the external mesh of a character.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

1. Introduction and Motivation

In recent years, physics-based muscle simulation has become increasingly popular in the Animation and Visual Effects industry. A detailed muscle system typically consists of five components: anatomically correct muscle geometries; a fiber field defined on them; a constitutive model supporting fiber activation for contraction; a skeleton model on which the muscles attach via constraints and a physics solver that computes the dynamics of the system. One of the most time consuming steps of a character simulation pipeline is the initial setup of the input model. In fact, muscle geometries can be generated in various ways: sculpted per character by trained artists, created from MRI and CAT scans data as in [JBE*16] or transferred from an existing template. The sculpting process is time consuming because it requires ad-hoc work for each character, the data scans are not always available and the fitting of templates to new characters presents challenges when dealing with non-humanoid or fantastic creatures which have atypical features, body proportions and body mass distribution. Moreover the result is generally not simulation-ready because of the interpenetrations present in the input meshes.

In this work we reduce the complexity of the sculpting process by introducing methods that allow artists to create volumetric muscle shapes (complete with tendons) from sketches of muscle silhouettes (patches) defined on the external surface of a character (skin) and attachment points corresponding to insertion and origin of anatomical muscles on the skeleton. By using an artist-

DOI: 10.2312/egsh.20171008

led physically-based simulation framework that includes shape inflation in the direction of the skeleton and muscle-to-muscle and muscle-to-bone collision handling, we allow user to design plausible volumetric reconstructions. As a result the shape of the muscle and the corresponding fiber field are obtained, both ready for further simulations. The key idea of our system is to exploit the information of the sculpted or scanned skin of a character, define areas on it corresponding to superficial muscles and use them to generate plausible volumetric reconstructions which are interpenetration-free. The results obtained show how using patches can reduce the complexity of the sculpting process, while guaranteeing that output meshes are suitable for simulation.

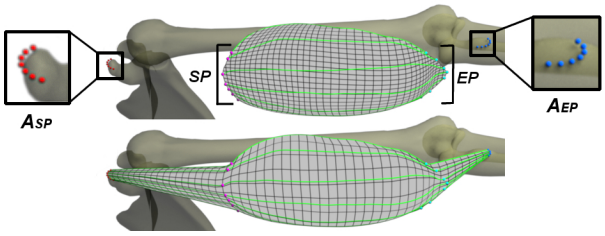

Figure 2: *Top: initial bicep patch segmented from the character's skin; initial fiber curves in green; point sets SP and EP in magenta and cyan; target point sets A_{SP} and A_{EP} in red and blue, respectively; bottom: tendon reconstruction via flock simulation.*

2. Related Work

The human body consists of over 650 muscles, but the ones having a visible effect on the surface appearance are about 110. Previous research work containing methods to produce muscles primitives relies mainly on existing anatomical data. [TSB*05] create tetrahedral meshes from the Visible Human dataset via level sets and are able to reconstruct missing tendon information using Constructive Solid Geometry. More recently, the work by [JBE*16] reconstructs very accurate musculature of a full body female subject via segmented cross sectional MRI data for simulation purposes. [SZK15] instead utilize a commercially available anatomical model and focus on the physics-based modeling of different body shapes by growing or shrinking muscles and fat. In production, because of the complexity of modeling muscles, a popular way to create them is by using either parametric surfaces or procedural primitives controllable with numerous parameters.

The fibers in [TSB*05] are obtained with B-spline solids and used to simulate muscle contraction with the Finite Elements Method and a transversely isotropic constitutive model, while the more recent work of [CB13] focuses on obtaining the fiber field by solving a Laplace equation subject to flux boundary conditions in the tendon regions.

Our work is inspired by the field of computational design, where [STBG12] compute the rest shape a balloon must have in order to achieve a desired target when inflated, using a physics-based optimization.

3. Method

The inputs of our system are a character model, its skeleton geometry, patches to be inflated and tendon attachment points. A patch $\mathcal{P}_i$ is a segmented subset of the input character mesh and follows the silhouette of the visible muscles under the skin. Users can manually segment the outer skin by placing on it nodes of a closed spline curve using their anatomical knowledge and surface details; the enclosed mesh is then cut and extracted (see Fig. 1b). For each muscle that a patch represents, attachment points are positions on the input skeleton where its tendons attach (see section 3.2).

3.1. Fiber Curves

The initial polygonal patch is dependent on the input skin topology by construction; this means that in general the edges are not following the ideal fiber flow of the patch shape; therefore the simulation can suffer from unaligned and non uniform elements. For this reason the patch is first converted to an intermediate parametric representation for which the topology is controllable. This is achieved by first sketching a series of fiber curves along the desired flow. In particular, after defining a set of starting points SP and ending points EP on the borders corresponding to where the tendons would attach, a fiber curve fib_j connects a point $sp_j \in SP$ to a corresponding end point $ep_j \in EP$ on the opposite border (see Fig. 2 top). This operation can be performed manually (as in this paper), or alternatively one can use the method in [CB13]. The fiber curves, after a resampling step, produce a parametric NURBS surface through a process of lofting; the result can then be converted back to polygons keeping full control on the resolution. These operations provide a mesh with clean topology and edge flow that follows the fiber directions.

3.2. Tendon Reconstruction

The retopologized patches are still missing the tendons that connect the muscle to the bone. We reconstruct this missing geometry by running a sequential flock simulation, using as agents (boids) particles that are positioned at the points of SP/EP and with velocity equal to the tangent of fib_j at those points.

Boids are subject to the standard known rules of cohesion, separation, steering towards target, alignment and collision avoidance [Rey87]. Given target attachment point sets A_{SP} and A_{EP} specified on the skeletal bone geometries as input, the steering rule is responsible for driving the boids starting from sp_j and ep_j towards their corresponding goals (see Fig. 2). These boids define curves extending the initial input fiber curves that can be lofted and turned into a polygonal surface as explained in section 3.1 to produce the final patch geometry. By varying the weight of the output vectors of the rules, the user has control over the smoothness and curvature of the geometry connected to the patch. We also reduce the velocity of the boids after entering a minimum distance from the target to perform a better trajectory. Additionally, we add a collision rule that prevents tendons from intersecting each other by steering away from other tendons. A boid i checks if its distance from an object is less than a safety offset d and steers away from it using the vector:

$$\mathbf{steer}^{\mathbf{i}}{}_{coll} = projTan(\mathbf{n}^{\mathbf{i}}{}_{closest}, (\mathbf{p}_{bary} - \mathbf{p}^{\mathbf{i}})) - \mathbf{v}^{\mathbf{i}}$$

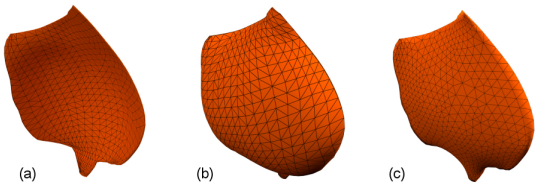

Figure 3: *(a) Deltoid extruded patch; (b) inflated muscle; (c) new rest shape after smoothing and relaxation.*

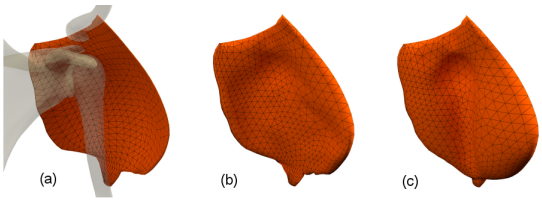

Figure 4: *(a) Deltoid extruded patch with collision objects; (b) partial inflation; (c) full inflation.*

where $\mathbf{p}_{bary}$ is the barycenter of the rest of the flock, $\mathbf{p^i}$ and $\mathbf{v^i}$ are the current position and velocity of boid i and $projTan(.,.)$ is a function that returns the tangent component of an input vector given the normal at the closest point on the mesh $\mathbf{n^i_{closest}}$. The magnitude of $\mathbf{steer^i}_{coll}$ is clamped to a maximum allowed value.

3.3. Muscle Generation

The initial patches are three-dimensional open manifolds. The next step of our method is to find a way to turn these surfaces into closed meshes obtaining plausible muscle shapes. The patches are, by construction, very close to each other but not intersecting. Therefore the desired rest closed configuration of the patch is forced to be very thin. We proceed by extruding the initial patch by a very small controllable offset. Then in a first phase we apply pressure to the extruded vertices and simulate without collisions until full inflation. With the result, we generate a new rest shape by applying alternated Laplacian smoothing iterations and tangential mesh relaxations only on the inflated vertices (see Fig. 3). The iterations continue until an initial estimate of the volume is reached. This step is needed so that the resulting mesh is much less likely to create unwanted folds when inflated. In a second phase, we run the actual simulation with collisions in which final muscles are created. At the very start of the simulation, the vertex positions of the initial extruded vertices constitute the initial conditions applied to the rest shape created in the first phase (so there is a non-negative elastic potential).

Inspired by the physics of rubber balloons [STBG12], increasing pressure forces are applied on the nodes of the closed elastic patch, towards the skeleton. During inflation, the vertices of the initial patch with tendons are set fixed as Dirichlet boundary conditions and are not allowed to move in order to obey the user input. In this process deformable meshes collide with other inflating patches and with the rigid skeletal bones (see Fig. 4). The inflation continues until a desired volume amount is achieved or until a specified percentage of the total area is in contact. When inflating, artistic control is important to direct how the shape evolves in time. For this reason artists have full control over the individual muscles' pressure parameters by painting a spatial pressure weight map (per vertex).

Elastic Model and Forces. The forces involved in the simulation are the stretching and pressure forces. Inspired by the work of [WOR11], we represent the rest pose of the inflatable patches in a two dimensional material space computing for each triangle material coordinates of undeformed and deformed configurations. We calculate in-plane stretching forces using a simple model for the elastic energy. At initialization stage we calculate 2D material space coordinates for each triangle and the discretized deformation gradient $\mathbf{F}$ of dimension 3x2. From $\mathbf{F}$ the right Cauchy-Green tensor is computed, defined as $\mathbf{C} = \mathbf{F}^T\mathbf{F}$. With it the familiar metric Green's nonlinear tensor is derived: $\mathbf{G} = \frac{1}{2}(\mathbf{C} - \mathbf{I})$.
From Hooke's law, the in-plane stress is calculated using a linear stress-strain relationship: $\sigma = \mathbf{B} \cdot \varepsilon$, where $\mathbf{B}$ is a precomputed 3x3 symmetric stiffness tensor that depends on Young's modulus E and Poisson's ratio ν while σ and ε are expressed in Voigt form.
Considering that the surface should not present transverse shearing resistance, $\mathbf{B}$ can be simplified from the original 9 parameters to 6 as in [WOR11]. The elastic potential energy density per triangle can then be defined as: $E_e = \int_{V_e} \frac{1}{2}(G : B : G) = \frac{1}{2}(G : \sigma) \cdot \overline{A}_e h$,where $\overline{A}_e$ is the area of the undeformed element, h its thickness and $V_e = h\overline{A}_e$ its volume. The total stretching energy of the deformable patch is obtained by summing up the contributions of all the elements: $E_{stretch} = \sum_i^{faces} E_e^i$.
Finally, stretching forces and Hessian (needed for implicit integration), are obtained by analytical derivation solving: $-\nabla_x(E_{stretch})$ and $-\nabla_x^2(E_{stretch})$ respectively.

As described in [STBG12], muscles inflate thanks to a discrete nodal pressure force defined as: $\mathbf{f}_i = \sum_{j \in \mathbb{N}_i} \frac{1}{3} \cdot p_i \cdot A_j \mathbf{n}_j$,where $\mathbb{N}_i$ are the neighbouring triangles of vertex i, A_j and $\mathbf{n}_j$ the area and normal of triangle j, and p_i is the base pressure per vertex. In practice, small triangles present in the tendon region result in small pressure forces, which prevent the tendon from fully inflating. Therefore we use a unit area A_j for all triangles. A topology-independent method as the one presented in [CBE*15] would be beneficial. Pressure is increased linearly progressively over time and we set it to zero when a node collides, in order improve the solver's stability. Note that bending forces can be disregarded in the simulation because the progressive pressure increments that define the shape over time cause stretching forces to be very large and the most influential. When inflating, artistic control is important to direct how the shape evolves in time. For this reason artists have always full control on the single muscles pressure and material parameters (E and ν) parameters by painting a spatial weight map to create variations in the inflating behavior.

Fiber Field. Having a volumetric fiber field segmented by muscles is very important for a subsequent physics based simulation of a tetrahedral mesh with an anisotropic constitutive model capable of contraction. Alternatively the muscle geometries themselves can be tetrahedralized and simulated with fiber activa-

tions and intercollisions. One of the advantages of generating the closed patch by extrusion is that the creation of the fiber field of the inflated muscle is greatly simplified. In our method, first the fiber curves are duplicated and translated by the same extrusion offset. Triangle barycentric coordinates are calculated for each control node of the curve so that the inflation deformation can be transferred to the curves. The result are deformed fiber curves that follow the muscle (see Fig. 1d). From these curves we generate a three dimensional vector field of fiber directions which we store on a grid that can be interpolated trilinearly inside the volume.

4. Implementation and Results

The physics system is implemented as an Autodesk Maya plugin in C++, while the flocking system is a Python custom tool. The smoothing and relaxation steps of section 3.3 utilize standard Maya nodes. All the patches in this paper were generated manually. We noticed that for best results it is useful to maintain a minimum distance between one patch and another so that the subsequent steps do not create intersections in rest pose. For collision detection and response during inflation our system implements iterative constraint anticipation [OTSG09] with spatial hashing.

As use case we tested the presented method on a full arm of a muscular human character (see Fig. 1). A real arm consists of about 25 muscles; we used 12 of them (including one pectoral muscle) for our simulation as the rest are not superficial or do not contribute substantially to the deformation. Thanks to the constraints represented by the input patches, inflation produces reconstructions that the artist can expect and control. To test performance, for simplicity we split the arm in upper arm (8 muscles) and forearm (7 muscles) and simulate them separately. In both cases inflating objects collide with each other and with 6 rigid bones. We run a 400 frames long simulation computing 1 substep per frame with a small timestep of 0.0005 and additive pressure increase factor of 0.01 per frame. On a Intel Xeon X3470, our non optimized code runs on a single thread and takes a cumulative time of 904ms and 1222ms per substep for the 12K and 15K elements of forearm and arm respectively. This result can be greatly improved with multi-threading and, if desired, by trading the accuracy provided by a constrained based collision system for a faster approach using penalty forces. The muscles and associated fibers produced with our method, despite not being anatomically correct, present organic shapes that are usable directly for simulation of the character (for muscle contraction and secondary effects) (see Fig. 5 and supplementary video).

5. Conclusion and Future Work

In this work we presented a method to generate volumetric muscles based on input patches sketched on the surface geometry of a character. As a result we were able to obtain simulation-ready muscle shapes along with fiber fields to be used for further simulations. Despite the promising results, our system presents some limitations and space for improvement. The areas at the borders of the patch are in fact still quite sharp and could be made smoother by adding custom energy terms, more resolution or turning the constrained vertices at the border into soft constraints. Some produced muscles, because of the flatness of the source patch, do not always fill all the available interior space and leave gaps between them and the bones. One could increase the pressure, use a softer material or run the inflation process for a longer time to improve on these issues. In our examples we discarded the presence of a fat layer between muscles and skin: an offset based on a specified 3D texture could be applied per patch to take it into account ([SZK15]).
An alternative way to manually sketch patches, provided that the mesh contains UVs, a texture containing a representation of the muscle anatomy can be prepared and the initial patches automatically generated based on rgb colour values segmentation; this would also be effective for transferring them between characters. As future work, in addition to extending the system to the full body and try out methods of transferring patches across characters, we intend to test the application of a plasto-elastic material model and adaptive meshes.

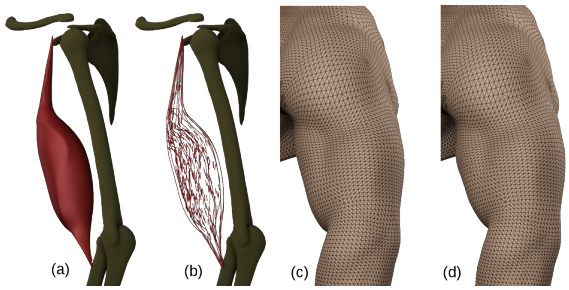

Figure 5: *(a) Generated bicep muscle; (b) Fiber curves and interpolated internal fibers; (c) Rest pose mesh; (d) Embedded mesh of deformed pose obtained using a tetrahedral FEM simulation with anisotropic material (isometric contraction).*

References

[CB13] CHOI H., BLEMKER S.: Skeletal muscle fascicle arrangements can be reconstructed using a laplacian vector field simulation. *PLoS ONE 8(10)*, e77576 (2013). 2

[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 SCA* (New York, USA, 2015), ACM, pp. 175–183. 3

[JBE*16] JACOBS J., BARBIC J., EDWARDS E., DORAN C., VAN STRATEN A.: How to build a human: Practical physics-based character animation. In *Proceedings of DigiPro* (2016), ACM, pp. 7–9. 1, 2

[OTSG09] OTADUY M. A., TAMSTORF R., STEINEMANN D., GROSS M.: Implicit Contact Handling for Deformable Objects. *Computer Graphics Forum* (2009). 4

[Rey87] REYNOLDS C. W.: Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph. 21*, 4 (Aug. 1987), 25–34. 2

[STBG12] SKOURAS M., THOMASZEWSKI B., BICKEL B., GROSS M.: Computational design of rubber balloons. *Comput. Graph. Forum 31*, 2 (May 2012), 835–844. 2, 3

[SZK15] SAITO S., ZHOU Z.-Y., KAVAN L.: Computational bodybuilding: Anatomically-based modeling of human bodies. *ACM Trans. Graph. 34*, 4 (July 2015), 41:1–41:12. 2, 4

[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 TVCG 11*, 3 (May 2005), 317–328. 2

[WOR11] WANG H., O'BRIEN J. F., RAMAMOORTHI R.: Data-driven elastic models for cloth: Modeling and measurement. *ACM Trans. Graph. 30*, 4 (July 2011), 71:1–71:12. 3