

Physics-Aided Editing of Simulation-Ready Muscles for Visual Effects

Fabio Turchet
MPC London, Bournemouth University

Marco Romeo
MPC London

Oleg Fryazinov
Bournemouth University

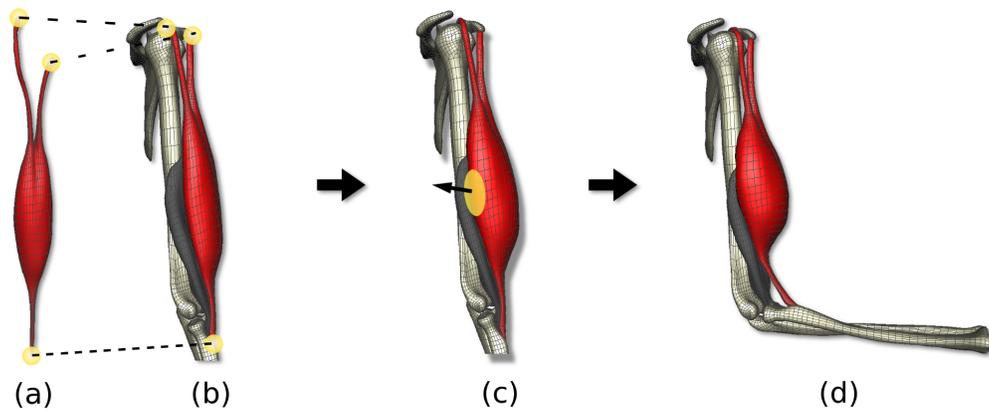


Figure 1: Main steps of our system in a scene with bicep brachii and brachialis muscles. From left to right: rest shape of a muscle (a); dynamic placement (b); context-aware sculpting of intersection-free geometry with tetrahedral cage deformation (c); simulation with contraction of the newly created muscle (d).

Keywords: physics based animation, muscle simulation

Concepts: •Computing methodologies → Physical simulation;

1 Introduction

Recent developments in character rigging and animation shape the computer graphics industry in general and visual effects in particular. Advances in deformation techniques, which include linear blend skinning, dual quaternion skinning and shape interpolation, meet with sophisticated muscle and skin simulations to produce more realistic results. Effects such as skin sliding, wrinkling and contact of subcutaneous fat and muscles become possible when simulating the anatomy of human-like characters as well as creatures in feature films. One of the main techniques adopted nowadays in the industry is the Finite Element Method (FEM) for deformable objects. Despite the life-like results, the setup cost to generate and tweak volumetric anatomical models for a FEM solver is not only very high, but it cannot easily guarantee the quality of the models either, in terms of simulation requirements. In a production environment in fact (see Fig. 1), models often require additional processing in order to be ready for FEM simulations. For example, self-intersections or interpenetrations in rest pose may result in unwanted forces from the collision detection and response algorithms that affect negatively the simulation at its start.

In this work we present a prototype framework and toolset, implemented as a Maya plugin, based on the 3D deformable object FEM library VEGA [Sin et al. 2013] that assist artists and technical direc-

tors (TDs) in the creation of simulation-ready muscle models in a production environment. Current 3D authoring applications do not provide efficient solutions for this task. Therefore we propose a way to minimise the number of iterations between geometry modelling and simulation-ready meshes production, which allows the artist to see the simulated result of contracting muscles interactively. As a result we increase the efficiency of modelling targeted for simulation. A typical workflow in our system consists of the following macro steps:

- Import of an existing muscle template from a library of meshes or existing muscle rigs
- Interactive physics-based placement of the muscle geometry through manipulation handles.
- Sculpting of the muscle automatically avoiding interpenetration with surrounding objects.
- Simulation and contraction preview of the newly generated shape.

2 Method

A typical muscle simulation requires a tetrahedral mesh, a high resolution triangular mesh for rendering, embedded by using barycentric coordinates or similar methods, collision objects (rigid or deformable) and constraints (boundary conditions). In addition, the tetrahedral mesh contains volumetric vector attributes such as a fibre field, continuum mechanics materials and activation levels used during contraction.

Template library Our system assumes the existence of a library of pre-sculpted anatomical models of bones, muscles and skin. The source models for the library can be taken from commercially available collections, as well as reconstructed from freely available medical MRI/CAT scans and muscles from existing characters.

Muscle placement Once the model is selected from the library,

the muscle placement stage begins. The goal here is to place muscle attachments (typically constrained tendons) of the template model near the bones via an interactive user-controlled simulation. One of the main features of our system is that it solves for collisions with nearby muscles and bones in the placement process, elastically adapting the muscle to preserve its volume. In this way our framework differs from standard 3D modeling tools which do not provide these features and which result in plastic deformations that can hardly produce a simulation-ready mesh. To control the placement, the user can add hard constraints on specified nodes of the mesh which serve as 6-DOFs manipulators. In the integrator, which implements Implicit Backward Euler, the constrained nodes inherit position and velocity of these manipulators, rather than being directly solved. This produces a change of the deformation gradient that triggers elastic deformation. Rayleigh stiffness damping and zero gravity are used to limit the effects of secondary motion; a quasistatic version of the solver could be alternatively used. In placement mode all surrounding objects remain rigid. For fast intersection and response tests with static objects such as bones we use Signed Distance Fields (SDF). For collision detection and response queries with dynamic objects, such as other muscles, we do not use SDFs to avoid field re-building, but a fast distance approximation with a BVH spatial data structure instead. Collision detection is based on the method described in [Teran et al. 2003]. Collision response is performed using a simple penalty-based method in which objects have a small collision offset hull around them. For each colliding vertex of the embedded mesh, a virtual spring is instantiated in the penetration direction and its dampened force is implicitly applied at each substep to the closest corresponding node on the cage. In our tests we used an isotropic nonlinear St Venant-Kirchhoff constitutive model, however a faster yet less accurate linear corotational material can also be used at this stage.

Sculpting Once the placement process is complete, the artist might still require further adjustments of the muscles’ shape to fulfill creative and functional requirements. For this purpose the proposed toolset builds on top of standard sculpting Maya tools by providing a specialized intersection resolution node, acting on the high-resolution embedded shape. After two consecutive sculpting operation s_1 and s_2 , the vertices of the new deformed mesh V_{s_2} are compared against the vertices at s_1 to detect which ones have changed. After a broad-phase collision detection based on bounding boxes that finds surrounding objects, the vector between the old V_{s_1} and new vertex position V_{s_2} defines the direction for a ray-triangle intersection test (accelerated with a static BVH). As intersections are solved at each operation, this ensures that in the next operation the first hit point is where the vertex is stopped. In addition, the sculpting tool can optionally apply to the point the same collision offset used during simulation.

The plastic deformation performed by the sculpting process is not used to generate a blend shape for a pose, but results in a mesh that becomes a new rest pose used by the solver to compute elasticity and perform further simulations. Note that the artist sculpts the embedded mesh, but the solver needs the volumetric cage for simulation. In order to avoid additional expensive tetrahedralizations, our framework uses a custom wrap deformer based on closest point distances and adjustable falloff. This is used to deform the internal nodes of the cage mesh and the associated fibre field. Compared to the embedded mesh, the tetrahedral volume has fewer degrees of freedom, so the wrap process cannot guarantee that the cage perfectly contains it as before the sculpting stage. This has not been a problem for the embedding process.

Simulation Once the new sculpted muscle is completed, the barycentric coordinates have to be recomputed and the solver is re-initialized with the new tetrahedral mesh rest shape to preview the

contraction of the muscle in an animation. In the simulation step we use the same solver as in the muscle placement stage which allows the integration of the same data structures seamlessly as a part of the same framework. The muscle material can be changed from a pure isotropic to a transversely isotropic that makes use of the fibre directions field. In particular, we implemented the constitutive model presented in [Teran et al. 2003] with active and passive stress response, with different Young modulus for tendons and muscle body materials. In the simulation stage the system supports not only hard constraints, but also soft and sliding constraints. This allows to create the interconnected network of connective tissue known as fascia which keeps the muscles packed together during motion.

Implementation The solver and the sculpting tool are implemented as C++ multi-threaded Maya plugins. The initial tetrahedralization is performed using TetGen and the SDFs are stored using OpenVDB.

In terms of performance, one muscle made of 400 tetrahedra subject to 15 hard constraints and 100 collision constraints interacting with one bone and one other muscle can be placed interactively at ~ 15 fps using a timestep of 0.01s. Our code can be further optimized.

3 Conclusions and Future Work

In this work we presented a workflow aimed at helping the modelling of simulation-ready muscles in a production environment for the visual effects industry. The system was developed as a proof-of-concept prototype and is yet to be fully tested before going into the production pipeline. Nevertheless it shows promising results by proposing an integrated solution to a practical problem that rigging artists are facing while creating complex muscle systems.

There are many areas of improvement on which we want to focus on. First of all, we would like to adapt a constraint-based dynamics approach to avoid the problems of stability of penalty forces when high stiffnesses make the system matrix poorly conditioned. To increase the user experience and make the simulation to more closely resemble the aimed results, it would be also beneficial to add an external skin mesh representing the superficial fascia as a boundary condition surface on which both the sculpting tool and the simulation can collide against. For the placement stage, we are also planning of adding a plastic deformation functionality directly integrated in a fast solver as in [Saito et al. 2015] where a decomposition of the deformation gradient is used to grow the muscles.

Acknowledgements

The authors want to thank the rigging team at MPC London and in particular Tom Reed and Benjamin Jones. Special thanks to Jamie Portsmouth for the support and Andy Van Straten via CGCircuit for the anatomical models.

References

- SAITO, S., ZHOU, Z.-Y., AND KAVAN, L. 2015. Computational bodybuilding: Anatomically-based modeling of human bodies. *ACM Trans. Graph.* 34, 4 (July), 41:1–41:12.
- SIN, F. S., SCHROEDER, D., AND BARBIC, J. 2013. Vega: Non-linear fem deformable object simulator. *Computer Graphics Forum* 32, 1, 36–48.
- TERAN, J., BLEMKER, S., HING, V. N. T., AND FEDKIW, R. 2003. Finite volume methods for the simulation of skeletal muscle. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA ’03, 68–74.