Volumetric Muscle Controller

Seunghwan Lee
Seoul National University
seunghwan.lee@mrl.snu.ac.kr

Nathan Mitchell
University of Wisconsin-Madison
nmitchel@cs.wisc.edu

Mridul Aanjaneya
University of Wisconsin-Madison
mridul.aanjaneya@cs.wisc.edu

Eftychios Sifakis
University of Wisconsin-Madison
sifakis@cs.wisc.edu

Jehee Lee
Seoul National University
jehee@mrl.snu.ac.kr

ABSTRACT

We describe a controller with a fully integrated musculoskeletal model actuated by more than a hundred muscles. Inspired by a QP-based control algorithm, we improved the algorithm to control the volumetric muscles directly. Furthermore, the highly-detailed musculoskeleton was modeled using a non-manifold method and a computation time was effectively reduced through implicit jacobian computation. Our entire system can simulate the musculoskeleton model with detailed volumetric muscles in a feasible computation time.

CCS CONCEPTS

• **Computing methodologies → Physical simulation**;

KEYWORDS

Biomechanics, Musculoskeletal system

ACM Reference format:
Seunghwan Lee, Nathan Mitchell, Mridul Aanjaneya, Eftychios Sifakis, and Jehee Lee. 2017. Volumetric Muscle Controller. In *Proceedings of SCA '17, Los Angeles, CA, USA, July 28-30, 2017,* 2 pages.
DOI: 10.1145/3099564.3106646

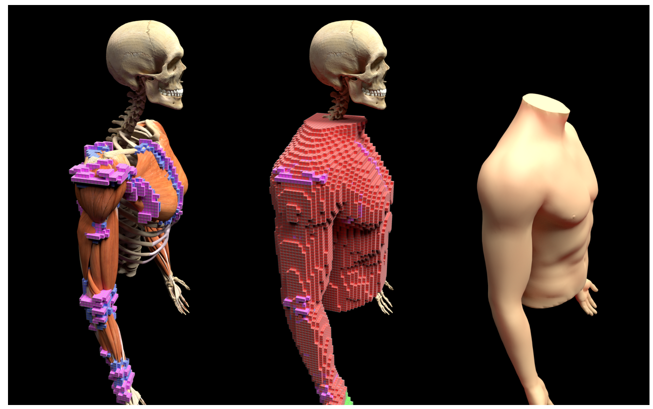

Figure 1: The model of volumetric muscle: The voxeliziation of the tendons, the voxeliziation of the total system and the flesh.(from left to right)

1 INTRODUCTION

Reproducing natural human movements is an important goal in Computer Graphics. Physics-based simulation has been widely used as a good tool to create the natural human motions. Considering the human body as an ideal robot, a controller applies torques directly at the joints. Recently, there have been efforts to incorporate biological characteristics by utilizing the musculoskeletal model. However, several critical challenges have emerged when we try to simulate a complex and detailed musculoskeletal model. For examples, Hard(bones) and soft(muscle, tendon and flesh) bodies are simulated and coupled. In addition, the contacts and the collisions between the muscles and the bones and between the muscles have to be handled properly.

[Lee and Terzopoulos 2006], [Wang et al. 2012] and [Lee et al. 2014] proposed a line-segment muscle model to simplify the physical properties of the muscles. They described algorithms to control the simplified musculoskeletal model, and showed the humanoid controlled by the muscles. Meanwhile, [Sifakis et al. 2005] and [Lee et al. 2009] describe a system to incorporate volumetric muscles in an one-way coupled way. They embeded the line-segment muscles in the volumetric muscles to control the humanoid while merely actuating the volumetric muscles. They showed the movements of the volumetric muscles such as bulging effects which are impossible to observe in the line-segment muscle model. However, they still used static postures or kinematics, the features of the volumetric muscles was ignored when they generate the motions.

In this paper, we propose a system to control the comprehensive musculoskeleton using the volumetric muscles. By controlling the volumetric muscles, not only we can reproduce the natural human motions, but also we can simulate the changes of movements due to the muscle growth or the topology changes of the muscles. Inspired by the QP-based control algorithms [Lee et al. 2014], [Lee et al. 2009], we improve the algorithms to use volumetric muscle directly. In addition, we use lattice-based simulation and implicit jacobian computation to make the computation time of the simulation feasible. We also apply the non-manifold method to the musculotendon to describe the details of the musculotendon. As a result, we propose an algorithm that efficiently controls the detailed musculoskeletal model.

2 OUR APPROACH

We employ the multi-physics framework to construct the complex musculoskeleton. We also enhance our algorithms to utilize the

SCA '17, Los Angeles, CA, USA

DOI: 10.1145/3099564.3106646

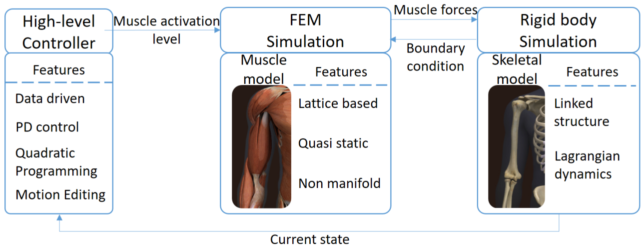

Figure 2: The simulation loop of our system.

volumetric muscles. Furthermore, we enrich our system by creating detailed musculotendon, and optimizing our computation time. Section 2.1 describes how to simulate and couple rigid and deformable body simulators. Section 2.2 describes our QP-based control algorithm. Section 2.3 presents the non-manifold method for detailed modeling of the musculoskeletal model, and finally section 2.4 describes the implicit jacobian computation method to optimize the computation time of our simulation loop.

2.1 Simulation and Coupling

We focus on the upper body to ignore the balance problem. Each bones are articulated by the joints, and the muscles, the tendons, and the flesh are simulated as deformable bodies using FEM(Finite Element Method). For the skeleton, we fixed the complex movements in the neck and the torso to make our system simple, and modeled the wrist, elbow, and shoulder joint as a ball-and-socket joint, respectively. For the deformable body, we embed the muscle and the tendon in the flesh, and we added a zero-length spring constraint to the skeleton and the corresponding flesh node so that the flesh can follows the skeleton's transformation. We also attached dirichlet boundary conditions to the end of the tendons to compute the musculotendinous forces.

2.2 QP-based Control

Our goal with the controller is to compute the optimal muscle activation level given the desired transformation of the skeleton. Our system is essentially an under-specified system where the number of muscles are greater than the DOF of the skeleton. Therefore, we try to find the optimal muscle activation level by minimizing its norm.[Lee et al. 2014]

2.3 Non-manifold Method

When both the muscle and tendon are embedded in the flesh, the force dissipates by the adjacent muscle. This phenomenon is fatal, especially because the muscles in the forearm are clustered in a small region. We apply a non-manifold method for each tendons to detach the tendons away from the flesh to prevent the forces from vanishing. In the second figure in Figure 1, the purple elements illustrate the elements separated from the flesh.

2.4 Implicit Jacobian Computation

Since our controller controls the forces using muscle activation levels, it is essential to compute a jacobian of the forces with respect to the muscle activation levels. Unfortunately, the human body have over 100 muscle actuator in upper body, and we simulate the body using over 100,000 nodes, so computing the jacobian numerically takes a long time. Instead, we compute the jacobian implicitly using a stiffness matrix we already have computed. This is faster than the Naive method because it only requires a single linear system.

3 ACKNOWLEDGEMENT

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the SW STARLab support program (IITP-2017-0536-20170040) supervised by the IITP(Institute for Information & communications Technology Promotion.

REFERENCES

Sung-Hee Lee, Eftychios Sifakis, and Demetri Terzopoulos. 2009. Comprehensive Biomechanical Modeling and Simulation of the Upper Body. *ACM Trans. Graph.* (2009).

Sung-Hee Lee and Demetri Terzopoulos. 2006. Heads Up!: Biomechanical Modeling and Neuromuscular Control of the Neck. *ACM Trans. Graph.* (2006).

Yoonsang Lee, Moon Seok Park, Taesoo Kwon, and Jehee Lee. 2014. Locomotion Control for Many-muscle Humanoids. *ACM Trans. Graph.* (2014).

Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. 2005. Automatic Determination of Facial Muscle Activations from Sparse Motion Capture Marker Data. *ACM Trans. Graph.* (2005).

Jack M. Wang, Samuel R. Hamner, Scott L. Delp, and Vladlen Koltun. 2012. Optimizing Locomotion Controllers Using Biologically-based Actuators and Objectives. *ACM Trns. Graph.* (2012).