

Real-Time Anatomically Based Character Animation

Juan Ramos*

Universidad Rey Juan Carlos, Madrid, Spain

Caroline Larboulette†

1 Introduction

This poster presents a method for deforming the skin of 3D characters in real-time using an underlying set of muscles. We use a geometric model based on a revolution of two cubic splines joined with C^1 continuity to generate fusiform muscles. They are attached to the bones through tendons at origin and insertion points. To simulate the force exerted by the muscles under strain and the consequent skin bulge, our model includes tension under isometric and isotonic contractions while preserving volume. We also provide an accurate sliding movement of the skin over the muscles. We have tested our algorithm with a GPU implementation obtaining real-time for a 100 character instances composed of 1400 muscles total. Our algorithm is suitable for applications where accurate skin deformation is desired as well as video games or virtual environments where fast computation is necessary.

2 Muscle Model

Similar to previous models [Wilhelms and Gelder 1997; Lee and Ashraf 2007], we make the assumption that the force exerted by the muscle follows a fictive *action line* that lies on the central axis of the muscle shape. This action line is anchored to the bones at the *origin* and *insertion* points. Our muscle is defined entirely by 8 parameters which gives a wide variety of non-symmetrical shapes. An elliptical cross section can be obtained by changing the scale factor along a chosen direction. An additional tension parameter is used for isometric contraction which makes the muscle bulge in one direction while it narrows in the perpendicular direction. Volume preservation is approximated by summing up the volumes of two truncated elliptic cones. We have implemented a Maya plugins that allows the user to modify the 8 parameters described above to design the muscles.

3 Skin Deformation

The skin deformation algorithm is carried out in two steps: first, a standard method to deform the skin as a result of the bones movement is computed; skeleton subspace deformation, dual quaternion skinning or any other method can be used. Next, a vertex displacement is added due to the deformation of the underlying muscles. We use two methods to displace the skin mesh vertices by the influence of a muscle: a radial one, which uses the muscle's action line as the center from which the vertices are displaced; and a vector oriented one, where the user chooses a direction to displace the affected vertices. That displacement is weighted for each vertex to let the user fine-tune the resulting deformation. An additional fat layer keeps a distance among the skin and the underlying muscle. The final result is a visually accurate sliding skin over the muscles. In addition, as the character rigging process is a difficult task by hand, we have reduced the setup time by proving an automatic muscle weighting system based on templates. Our Maya plugins lets the user choose an influence template for each muscle which calculates the weights for all of the vertices in its neighborhood. Normal recomputation is necessary when the muscle deforms the skin as the original normal becomes invalid.

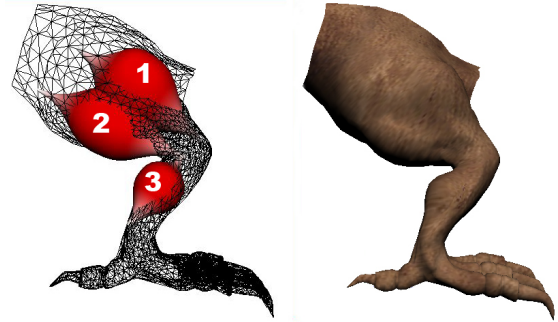


Figure 1: Setup of the three muscles we used to animate a dinosaur leg. On the left, the modeling of the muscles. On the right, the resulting mesh deformation with normals recomputed.

4 GPU Implementation

When programming on the GPU it is important to keep data transfer between the CPU and GPU low. We use a parametric representation to define a muscle. Only a very small data structure (72 bytes) is sufficient to store its current shape. Furthermore, the basis of the parametric primitives that serve for the muscle shape construction are cached in the GPU's memory for enhanced performance. As a GPU does not have enough vertex adjacency information to compute smooth normals we have used a pre-calculated vertex adjacency index buffer in a two step process: the first step calculates all vertices displacements due to bones and muscles. The second step, recomputes the normals using the adjacency buffer.

To sum-up, our contributions are:

- A new parametric fusiform muscle model capable of both isometric and isotonic contraction that preserves its overall volume.
- A new geometric algorithm for skin deformation that achieves skin sliding behavior without the high cost of a physics simulation.
- An automatic muscle weighting system based on templates that facilitates the rigging process.
- A GPU implementation of our model including normal correction that is suitable for use in video games.

References

- LEE, K. S., AND ASHRAF, G. 2007. Simplified muscle dynamics for appealing real-time skin deformation. In *International Conference on Computer Graphics and Virtual Reality 2007*.
- WILHELMS, J., AND GELDER, A. V. 1997. Anatomically based modeling. In *Proceedings of SIGGRAPH'97, ACM Computer graphics*, ACM Press/Addison-Wesley Publishing Co., 173–180.

*e-mail: tomillo1235@gmail.com

†e-mail: caroline.larboulette@urjc.es