Graphical Simulation of skeletal muscles for virtual humans characters and humanoid robots

Octavio Navarro Hinojosa[1] and Moises Alencastre Miranda[1]

Tecnolgico de Monterrey, Campus Santa Fe,01389 Mexico City, Federal District
A00967953@itesm.mx, malencastre@itesm.mx

Abstract. Current muscle animations and simulations are based on the Hill muscle model or on techniques such as motion capture. However, those approaches do not generate accurate and real enough simulations and animations of the human body. We have analyzed the way skeletal muscles work and decided to generate a mathematical model to best simulate how these work. To simulate said model, we selected Bezier surfaces because of the likeness they have to the muscles, and because of the control they provide. We implemented a Bezier surface algorithm to run concurrently in GPU using CUDA, and are working with the mathematical model abstraction.

Keywords: Bezier surfaces, Skeletal Muscles, Parallelism, CUDA

1 Motivation and previous work

Human character's animations and simulations are used in many fields: from video games and movies to medical simulations. However, achieving a realistic looking and anatomically correct human character is a challenge in computer graphics [6]. Researchers have made some significant progress in the area. In [4], a complete biomechanical model of the human head-neck system is introduced. In [5], the previous work is extended, and a comprehensive biomechanical model of the human upper body is introduced. However, the achieved characters, models and simulations still do not resemble humans as accurately or realistically. Most of the proposed models also use the Hill muscle model [1], that uses a mechanical approach to muscle movement. By creating a mathematical model and simulations that emulates realistic human muscles, there are many applications that may benefit many fields: medical operations, crash dummy testing, fashion design, teaching, creating movies and animations, robot movement and control, among others. There lies the importance and motivation of this project.

2 Hypothesis

By developing and implementing a mathematical model of how the skeletal muscles of the human body work, it will allow us to generate real time graphical simulations of human characters that resemble and move like real human beings.

3 Methodology

To develop this project, we will first do research in order to understand how the muscles operate, and what previous work already exists. We will then select and implement a graphic structure that enables us to emulate the muscles behavior. Since performance is a big concern, we will develop all our algorithms with parallel techniques. After the graphic structure is implemented, we will abstract a mathematical model from real muscle data in order to apply it to our muscle's simulations. Once this model is implemented, we will conduct several experiments with different movements and poses from a selected group of muscles, so that we can compare them to the muscle data as well as to similar motion capture data. Once the mathematical model is validated, we can use it to model other muscles and generate a more complex simulation.

4 State of the research

In order to define a new mathematical model that is able to graphically simulate the real behavior of the skeletal muscles of the body, we first need to understand how the muscles operate. After reviewing sources such as [1] and [2], we understood that muscles are comprised of many fibers that contract and expand to generate movement. Because of this, we decided to use Bezier surfaces to simulate the real structure of the muscles.

Regarding the development of the Bezier surfaces, we first decided to use the surfaces that are already implemented within OpenGL. However, when we conducted some experiments, we realized that there is a limit with the number of control points that can be processed at once. On average, the maximum number of control points available is thirty. We also found no practical way to improve the performance using parallel techniques. Because of these, we decided to implement an algorithm to generate the surfaces, without limits regarding control points, line segments or parallelism.

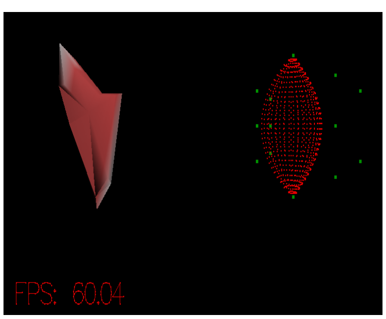

Fig. 1. Secuential Bezier surface simulation.

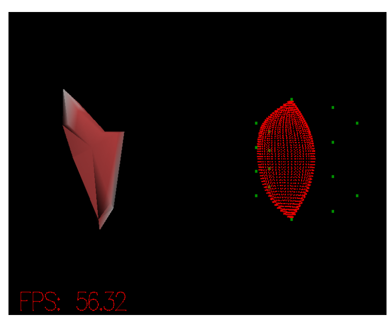

Fig. 2. Parallel Bezier surface simulation.

The Bezier surface algorithm is well established in literature, so we used [3] as a reference to implement it. We also developed a program that allows us to move the control points of a surface in real time in order to try to simulate a muscle's form. We used this program to simulate surfaces both sequentially and concurrently using CPU. The sequential surface, seen in figure 1, had 25 control points and 30 points per line segment. The concurrent one, seen in figure 2, had 50 control points and 50 points per line segment. We used OpenMP for the parallel part of the simulation.

The next step was to implement the algorithm and program in the GPU using CUDA for the surface point processing in order to increase its performance and to be prepared for the implementation of the muscle's mathematical model. We developed the Bezier surface algorithm using both blocks and threads of the GPU so that we could use the all its cores. A surface simulation using CUDA and the GPU can be seen in figure 3.

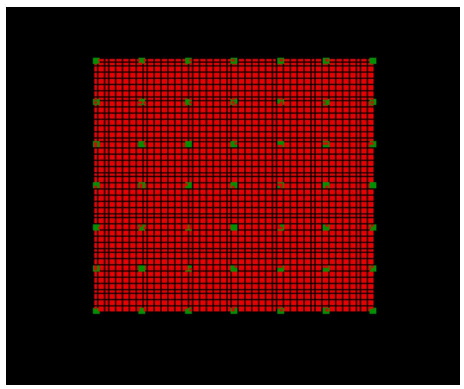

Fig. 3. Surface simulation using CUDA.

5 Preliminary Results

We were able to obtain an average of 60 frames per second on both the sequential and the parallel version of the surface simulation. The important difference was that we used a lot more control and surface points for the parallel implementation. Considering the amount of control and surface points for both versions, we found that if we used more than 30 control points and 40 points per line segment for the sequential version and 60 control points and eighty points per line segment for the parallel version, the frames per second dropped to around 15, which is way below the frames rate needed to perceive a smooth animation.

With the CUDA version of the algorithm, we found that we could correctly calculate surfaces with up to 70 control points and 200 points per line segment at 30 frames per second (with 150 points per segment or less, we managed to render at 60 frames per second). Figure 4 shows the speedups we obtained with surfaces of 70 control points and a varying amount of points per segment. It is important to note that the processing speedup of the surface points is really large (around 16000); however, a lot of time is used to copy memory from CPU to GPU, so we decided to consider that speedup for analysis as well.

	Dots per line segment					
	150	200	230	250	280	300
Sequential	-	-	-	-	-	-
Parallel	3.601637	1.32110	3.70494	3.718160	3.758594	3.750128
Cuda	16094.53	10489.74	54287.78	43946.92	52495.02	118593.9
CudaMem CPY	13.09415	4.718134	13.30434	13.10069	13.24649	13.20276

Fig. 4. Several speedups for the CUDA version of the algoritm.

The mathematical model is a critical part of this project, because its main objective is to adjust the surface points to better simulate a muscle's movements and positions. To be able to abstract such a model, we require a lot of muscle data, which mainly comes from an electromyography. We have just contacted some medical researchers of our institution who will provide us with electromyography data for us to analyze.

References

1. Hill, A.V.: The heat of shortening and dynamics constants of muscles. Proc. R. Soc. Lond. B (London: Royal Society) 126 (843): 136195. (1938)
2. Timothy Taylor.: Muscular System. [Online] Accessed August 24 2013 at http://www.innerbody.com/image/musfov.html#full-description
3. David F. Rogers.: An Introduction to NURBS: With Historical Perspective. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. (2001)
4. Lee, S.H.: Terzopoulos, D.: Heads up!: biomechanical modeling and neuromuscular control of the neck. In ACM SIGGRAPH 2006 Papers (SIGGRAPH '06). ACM, New York, NY, USA, 1188-1198. (2006)
5. Sung-Hee Lee, Eftychios Sifakis, and Demetri Terzopoulos. Comprehensive biomechanical modeling and simulation of the upper body. ACM Trans. Graph. 28, 4, Article 99 (September 2009), 17 pages. (2009)
6. Dongwoon Lee, Michael Glueck, Azam Khan, Eugene Fiume, and Ken Jackson. Modeling and Simulation of Skeletal Muscle for Computer Graphics: A Survey. Found. Trends. Comput. Graph. Vis. 7, 4. DOI=10.1561/0600000036. (2012)