Human Skeletal and Muscle Deformation Animation Using Motion Capture Data

Ali Orkan Bayer
Department of Computer Engineering, Middle East Technical University
06531 Ankara, Turkey
orkan@ceng.metu.edu.tr

Ayça Müge Sevinç
Tubitak Space Techonologies Research Institute,
06531 Ankara, Turkey
muge.sevinc@ceng.metu.edu.tr

Tolga Can
Department of Computer Engineering, Middle East Technical University
06531 Ankara, Turkey
tcan@ceng.metu.edu.tr

ABSTRACT

Human character animation has problems of unrealistic motion and unrealistic skin deformation. In this study, we present an application of motion capture data for generating realistic human character animation using anatomically correct skeleton and muscle models. For this purpose, we attach a skeleton model to the motion data and then construct and attach a set of muscle models to the underlying skeleton. Deformation under isotonic contractions are modeled using a number of geometric primitives. As a result, we have implemented a toolkit where animations can be viewed and recorded as a video file. In order to demonstrate the use of the toolkit, we have produced a number of short movies for entertainment purposes. The generated muscle deformations are realistic. As future work, we plan to apply skin to our model and generate realistic looking skin deformations.

Keywords: Character animation, motion capture, anatomical based modeling of the human musculature, muscle deformation.

1 INTRODUCTION

Ever since the beginning of animation, people are interested in human character animation. However, we do not see many human characters in computer generated animation movies, since animation of human characters is still a challenging task.

One of the major problems in animating human characters is the difficulty of creating realistic motion. Human body has many degrees of freedom and human motion includes many parameters. Even when the constructed motion is very close to the real one; since we observe humans all the time, satisfying the audience is a very challenging task. One of the most widely applied solution to this problem is the use of motion capture technology in which the movements of real humans are recorded and then applied to the virtual characters. In this study, we present yet another application of motion capture data by developing an application that maps generic motion capture data to a skeleton model with attached muscles. Our goal is to present the audience a realistic motion.

The second problem in human character animation is to create body deformations. Human skin is affected by and deforms with motion. A character that does not have any skin deformation does not look realistic.

In this study, we present a skeletal character animation using motion capture data. Also, a subset of the muscles of the upper body is produced to model the deformation of the body in motion.

We present our study in two main parts. The first part is for entertainment purposes and includes the generation short movies of a moving skeleton (See Figure 1). The second part presents muscle deformations during body movements. The main purpose of this paper is to present the know-how of the application of generic motion capture data to a specific human character.

2 BACKGROUND AND PREVIOUS WORK

2.1 Motion Capture

Motion capture is the procedure of locating some sensors on human subjects; and then with the help of tracker devices recording the motion of the sensors during the movement of the human body.

In this study, we have used motion capture data to obtain realistic motion. We have used the Carnegie Melon University's motion capture database [2]. In the database, there are many skeleton models for different human subjects and their recorded motions.

Although there are many different formats for representing motion capture data, all of these formats represent very similar information. In the CMU database .asf file format is used for the skeletal information, and .amc file format is used for the description of the move-

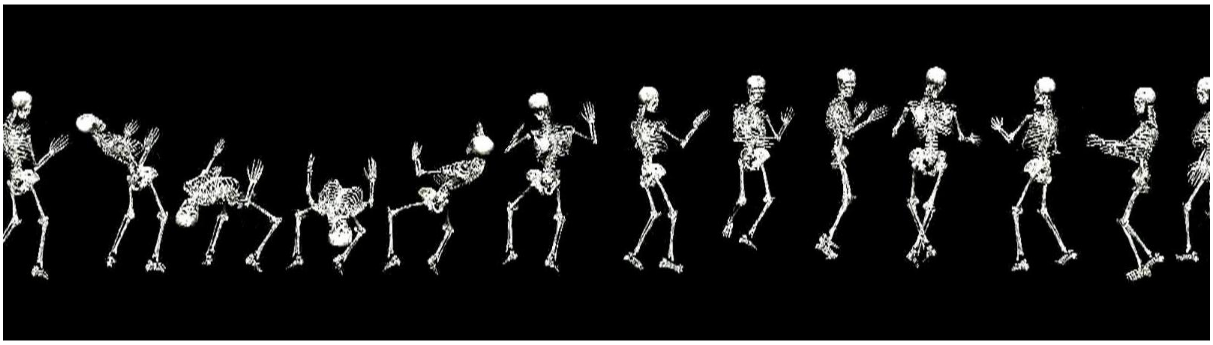
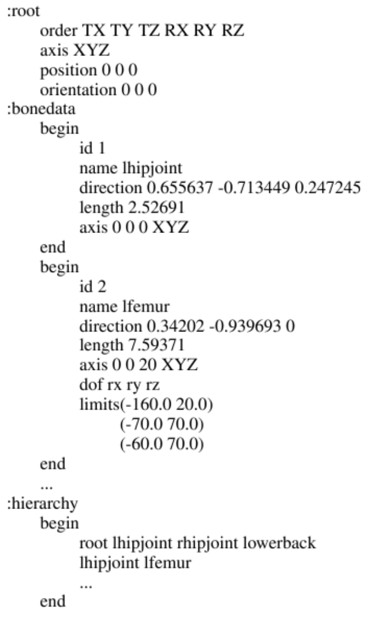


Figure 1: Animation of a skeleton with salsa dance motion capture data.

ment. The skeleton file (.asf file) supplies information about the skeleton of the human subject.

Figure 2 shows an example .asf file which consists of three parts containing information about the root of the skeleton, each joint, and the hierarchy of the skeleton. The part about the root of the skeleton contains the information such as the order of the channels (translation and rotation axes) that the motion would be applied to the root, the rotation order (eg., XYZ or YZX), the starting position, and the orientation of the root. For each joint, the unique id and the name of the joint, the orientation vector in the world coordinate system, the length of joint, the rotation of the local coordinate system for the joint with respect to the world coordinate system, and how many degrees of freedom this joint has are given in blocks between the keywords 'begin' and 'end'. In the last part of this file, the parent-children relation of these joints are given. Figure 3 shows how the joints of the skeleton are represented hierarchically.



```
:root
    order TX TY TZ RX RY RZ
    axis XYZ
    position 0 0 0
    orientation 0 0 0
:bonedata
    begin
        id 1
        name lhipjoint
        direction 0.655637 -0.713449 0.247245
        length 2.52691
        axis 0 0 0 XYZ
    end
    begin
        id 2
        name lfemur
        direction 0.34202 -0.939693 0
        length 7.59371
        axis 0 0 20 XYZ
        dof rx ry rz
        limits(-160.0 20.0)
            (-70.0 70.0)
            (-60.0 70.0)
    end
    ...
:hierarchy
    begin
        root lhipjoint rhipjoint lowerback
        lhipjoint lfemur
        ...
    end
```

Figure 2: An example .ASF file (Taken from [2])

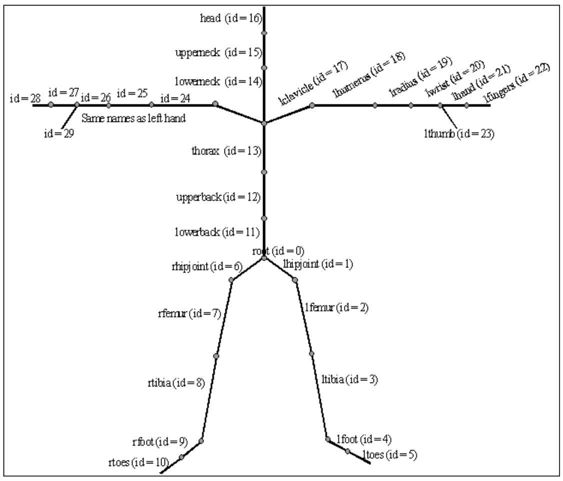


Figure 3: Skeleton hierarchy defined in .asf file (Taken from [7])

The motion file (.amc file) presents information about the transformations of the joints in their local coordinate systems for each frame.

2.2 Skin Deformation

The realistic looking motion should be supported with realistic skin deformation to obtain a successful human character animation. There are basically two approaches in the literature.

The first approach is known as smooth skinning or skeleton-subspace deformation [5]. In this approach, the shape of the skin is estimated by the transformations of the underlying joints. This approach is very efficient; however, it yields unrealistic results in some situations. For more information on the approach and the problems, the reader is referred to [5].

The second approach is a layered approach [1]. The layers include the skeleton, underlying muscles, and finally the fatty tissue and the skin. The skin actually sits on the top of the volume that is consumed by the skeleton and the muscle layer. Thus, skin deformation can be obtained by the underlying layers. The layered approach can be used successfully if the muscle and muscle deformations can be modeled correctly. Hence, studies on anatomical modeling emerged to target that problem. Scheepers [8] uses artistic anatomy and models the skeleton and the muscles as close to as reality. To simplify computations, the layers are modeled using geometric primitives. Realistic results are observed, however, it is very hard to model muscles and insert them to the skeleton manually. To overcome the difficulties of manual modeling, in [3], computer aided tomography data are used to model muscles and their behavior correctly.

One recent study, [6] uses large amount of optical markers that are placed on the skin of the subjects with high resolution. In this method, they try to capture skin deformations with the motion of these markers. The results are very satisfactory; however, there are limitations due to large number of markers.

In this study, we present anatomical modeling that is inspired from [8]. However, skin deformation is not in the scope of the study. We have only performed modeling and deformations of a small portion of the musculature. We have used a publicly available skeleton model and attached the model to motion capture data. This process was very time consuming since the rest position of the .asf file and our skeleton model were different. The rest position of the skeleton model and the mapping of that position to the motion capture data are shown in Figure 4. We have followed a similar approach to [8] and try to model muscles by using geometric primitives. The details of our study are given in the next section.

3 IMPLEMENTATION DETAILS

The application is implemented in C language using the openGL library. Also, ffmpeg library is used for producing the accompanying videos. We have tested our system on the Linux operating system.

The developed application consists of many modules. The first module parses the skeleton and motion data files into an appropriate hierarchical data structure. The second module determines the position of the joints by traversing the data structure in a hierarchical manner. The third module is the skeleton module which draws the bones. The final module, deals with the construction of the muscles. In addition to these modules, the last module, video module, saves the produced animations as video files. The following sections give the details of these modules.

3.1 Data Module

As mentioned above, this module is responsible for constructing the required data structures from skeleton and motion files. It reads the skeleton file and stores the skeletal information considering the parent and child relationship of the joints. The motion file is also read and each frame information is stored in the memory.

3.2 Motion Capture Engine Module

The motion capture engine module is the underlying module that generates motion. Each frame is generated by traversing the skeleton in a breadth first manner and applying the given transformations.

The transformations for each joint are given in the joint's local coordinate system. In that respect, the transformation matrices that give the global coordinate system given the local coordinate system are constructed. The .asf specification presents the rotations that maps the local coordinate system to the global coordinate system as a series of rotations around the global x, y, and z axes one by one. Briefly, the order of operations applied to each of the joints for each frame is given as follows:

- Translation is performed using the direction vector and the length of the joint.
- The rotation that maps the global coordinate system to local coordinate system is applied.
- The rotations of the joint are applied.
- The rotation that maps the local coordinate system to global coordinate system is applied.
- The transformation of the parent joint is applied.

3.3 Skeletal Module

The main purpose of the skeletal module is to draw the skeleton. For this purpose, the module applies the necessary transformations to each bone to attach them to the joints. As we have mentioned in the previous sections, the rest poses of the .asf data and the skeleton model do not fit. In that respect, the necessary transformations for attachment is determined manually by trial and error. Thus, this is a very time consuming process. We have actually supplied the correct transformations for only one .asf data (79.asf in the CMU database), the rest can be found by using the ratio of the length of the corresponding joints between the supported one and the targeted one, however, the application we have developed does not support this feature.[1]

This module also supports smooth transformations for some of the bones that consist of a bone structure rather than just a single bone, e.g., the spine. The .asf file presents 6 segments between the root and the head, where the spine fits, however, in reality the spine contains around 30 segments. If we segment the spine into 6 segments and rotate them using the motion data,

[1] Since in the .amc file, the motion is defined in the local coordinate system of the joints, the motion data for other skeletons can be applied to the supported one. It seems to work fine but there may be some problems in the synthesized motion due to the small differences in the calibration stage of the motion capture equipment for each subject. Thus, we plan to add support for other skeletons as future work.

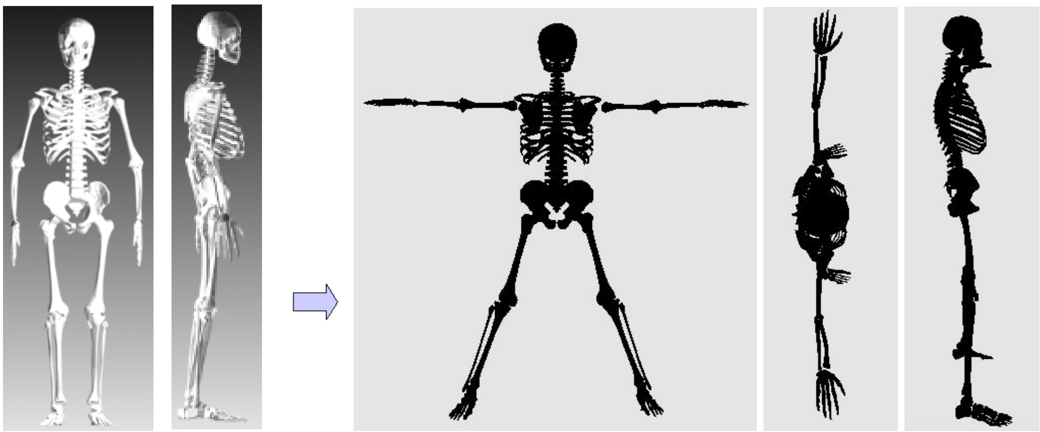


Figure 4: Attaching the skeleton model to the skeleton hierarchy in asf files requires mapping between the rest poses

the spine looks broken at the connection points. To avoid this problem we have applied the rotations to each vertex of each segment proportional to its local y-coordinate. Thus, the bottom vertex is not rotated, the upper vertex is fully rotated and the rotations of intermediate vertices are interpolated linearly.

Another purpose of this module is to determine the global coordinates of the attachment points of the muscles. This is needed because each end of the muscle is attached to a different bone. The attachment point of each muscle to each bone is determined manually by using the coordinates of the vertex of the bone that the muscle attaches (Weighted average of multiple vertices is also used.). Hence, for each frame the global coordinates of the attachment point must be calculated. This is done by applying the transformations of the corresponding joint and the bone to the coordinates of the attachment point. Later the muscle module draws the muscles that originates from these points.

3.4 Muscle Module

The muscle module is responsible for drawing the muscles between attachment points determined by the skeleton module. In addition, the module performs deformations that are caused by isotonic contraction.[2] The deformations caused by isometric contractions[3] are not implemented because the motion capture data do not provide us with the forces applied by each joint.

We have modeled only a small portion of the human musculature. Instead of using real anatomical data, the shapes and the attachment points of the muscles are modeled manually by trial and error using an anatomical atlas. This is also a time consuming process and with the help of digital anatomical data one can produce more realistic results. We have modeled 5 groups of muscles for both sides of the upper body. Brachialis muscle, biceps brachii muscle, triceps brachii muscle, deltoid muscle, and pectoralis muscle are modeled. The muscles we have modeled can be seen in Figure 5. We have generally used three modeling strategies for each of the groups, which are presented in detail below.

The muscle module generally uses some of the approaches given in [8]. It also allows tendons to be inserted. The details of the approaches we have used are given in the next sections.

Fusiform Muscle Brachialis muscle, biceps brachii muscle, and triceps brachii muscle are modeled using the "fusiform muscle approach"[8]. This approach constructs muscles using ellipsoids. This type of muscle takes the volume and ratio of two radii as parameters, also calculates the muscle length using the distance between the attachment points. Then the three radii of the ellipsoid can be found.

The fusiform muscle model that has certain length, width, and height uses the following parameters:

$$a = \frac{height}{2}$$
$$b = \frac{width}{2}$$
$$c = \frac{length}{2}$$

Thus, the volume, v and the ratio of two radii, r are given as:

$$v = \frac{4\pi abc}{3}$$
$$r = \frac{a}{b}$$

[2] In isotonic contraction, the volume of the muscle is preserved and the shape of the muscle deforms only as a result of the change in the length of the muscle.

[3] In isometric contraction the volume of the muscle is increased when it is tensed and is decreased when it is relaxed.

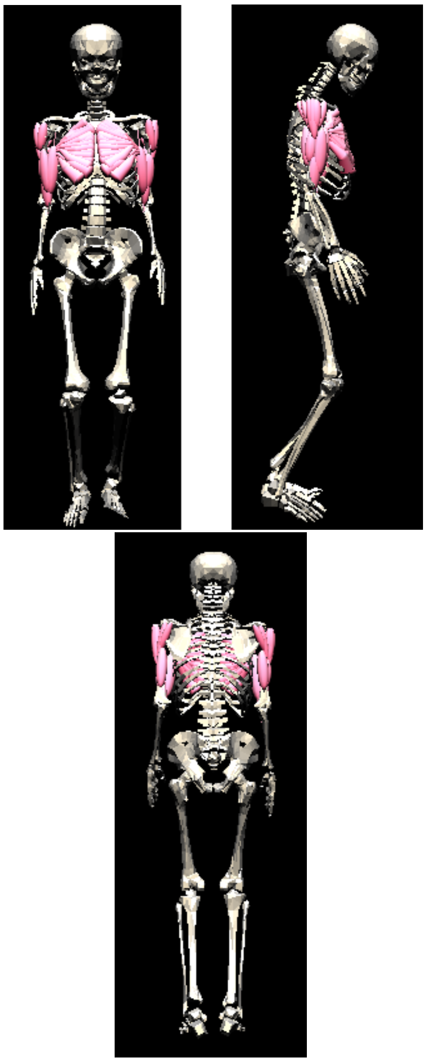


Figure 5: Anterior, lateral, and posterior view of the model.

When the length of a muscle changes, the new values of *a*, *b*, and *c* can be derived as follows considering the volume is preserved:

$$
\begin{aligned}
c' &= \frac{l'}{2} \\
b' &= \sqrt{\frac{3v}{4\pi r c'}} \\
a' &= b' r
\end{aligned}
$$

For the correct shape to be constructed, the local coordinate system of the muscle should be found. The z-axis of the muscle is determined to be the vector from the first attachment point to the other. The x-axis is found by taking the cross product of the z-axis with the vector from the first attachment point to the joint coordinate of the bone that the point is on. The y-axis is given as the cross product of x-axis and z-axis. The transformation matrix that maps the local coordinate system to the global coordinate system is constructed using these axes.

The fusiform model may also contain tendons. A tendon is a structure that attaches a muscle to a bone and its length does not change. Thus, tendons that have certain lengths can be given as parameters to the muscle model for having muscles with tendons. If tendons are used when calculating the length of the muscle, length of tendons should be subtracted from the distance between the two attachment points.

When modeling triceps brachii muscle, the distance between attachment points cannot be used as the muscle length, since the tendon of that muscle group passes from the back of the elbow. Thus, to estimate the correct length for triceps brachii muscle, an additional control point is added at the elbow.

Multi-Belly Muscle The multi-belly muscle model, inspired from [8], is used to model the deltoid muscle. This muscle consists of several segments and attachment points for each segment are given separately. Each segment of multi-belly muscle is actually a fusiform muscle. One way to insert a fusiform muscle for each segment is to use the technique described in the previous section, but using such an approach would yield sharp orientations for each segment. Instead of finding the orientations of each segment separately, this model uses the attachment points of the neighboring segments to find the orientation of the each segment. The local z-axis of each segment is the vector from the first attachment point to the second. The y-axis can be found by the cross-product of two vectors from the first points of the neighboring segments to the second point of the current segment. If the current segment is a boundary segment, one of the vectors is the z-axis of the segment. The x-axis can be found by the cross product of the z-axis and the y-axis.

The segments presented in this section behave similar to the fusiform muscle described above. The tendons for each of the segments are also added as described in the previous section.

Multi-Attachment Muscle The multi-attachment muscle is used to model pectoralis muscle. The pectoralis muscle sits on the ribs and one end is connected to sternum (the bone that connects the ribs at the front) and the other end to humerus (the bone that is between the shoulder and the elbow). To place the muscle on the ribs, we determine two additional control points that the muscle connects other than the attachment points. The cross-section area of the muscle gets smaller as moved from sternum to humerus. Thus, we have used cones as an approximation of this shape.

The deformation of the muscle is approximated by the change of the base radius of the cone as the length of the muscle changes. The new radius ,*r*, of the cone with volume *v* and length *l* is given as:

$$r = \sqrt{\frac{3v}{l\pi}}$$

The length of the muscle is determined as the sum of the Euclidean distances between the consecutive control points and the attachment points. The muscle is drawn as segments between these points starting from the first attachment point.

The orientation of this muscle can be found similarly as in fusiform muscles. However, the orientation of each segment is found separately. The z-axis of each segment is found using two attachment points, the x-axis is found by the cross product of the z-axis with the vector from the point of the joint that the muscle connects and its parent joint. The y-axis is similarly determined as the cross product of x and z axes. Actually, the orientations of x and y axes have no effect on the shape of the muscle, since the cross section of the muscle on z-axis is a circle.

3.5 Video Module

The video module is used to record the animation to a video file. We have used ffmpeg [4] library for recording each frame to a video file. We have only support for .mpg extension and 30 frames per second or below. The framebuffer is read from the memory for each frame and recorded as a video frame using the library.

4 THE APPLICATION OUTCOMES

The application we have developed has three main outcomes.

The first outcome is the animation toolkit that can be used either to view animation of motion capture data or to record the animation as a video file. The user can select to view or record only the markers, the skeleton, or the skeleton with muscles. The user can also select viewing directions by arrow keys (this property is very limited only to four degrees of freedom and should be developed further).

The second outcome is related to entertainment. We have observed that the motion of a human skeleton is very entertaining and decide to produce short movies using motion capture data. The production of movies started from the recording of the desired animation using the toolkit. Later, we have used a commercial movie editor to produce credits and titles, and to insert music or sound into the animation files.

The final outcome, is the muscle deformations. We have produced realistic looking muscle deformations. The pictures of the muscles are given in Figures 6 and 7. As the lengths of the biceps brachii and triceps brachii muscles change, their deformations can be easily seen. In this study, we have provided only small portion of the muscles. However, these muscles can be used to generate realistic deformation with a skin attached on top of them.

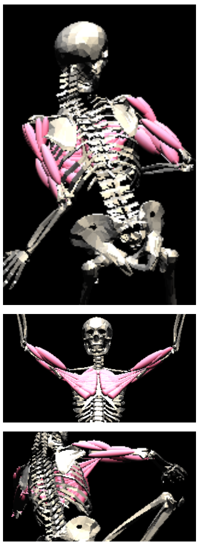


Figure 6: Muscle deformation with some postures of the model.

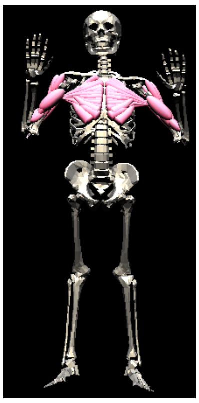


Figure 7: Muscle behavior in a posture.

5 CONCLUSIONS

Realistic modeling of the human body and motion is an interesting subject in computer animation. It is also very challenging because of many parameters that the human motion involves. In this study, we have produced human motion animation using motion capture data, modeling muscles, and simulating muscle deformations. We have constructed some of the human upper body muscles using some geometric primitives and a skeleton model which supports the muscles.

The produced deformations of the muscles can be used to find the skin surface deformations by attaching a skin over the volume that is formed by the skeleton and the muscles.

Although motion capture data yield very realistic results, it has some limitations. The markers for capturing motion is placed on the surface of the body. Therefore, when these marker points are attached to a skeleton, a number of problems will arise. One of the problems is the segmented rotation of the spine. We have solved that problem by applying rotation proportional to the y-coordinate of the bone and made the rotations smoother. The motion data for hands were also problematic. The hand definitions in general .asf files do not generally fit on a skeleton model; therefore we had to model the hands as rigid objects and could not present the motion of the fingers. Actually, the motion of hands requires high resolution markers and they are usually not regarded as an important component for full body capture.

Muscle modeling and deformation are done for only a small subset of the muscles. This is a tedious work because we did not employ any digital anatomical data. We have looked up the positions and the shapes of the muscles from an anatomy atlas and tried to obtain a similar shape at similar locations on the model. Results are realistic and can be used for the surface skin deformation. Whole body muscles can be modeled in a similar way.

In conclusion, we have tried to animate a skeleton character using motion capture data. Although we have produced movies at the skeletal level, we have obtained successful results for entertainment purposes. In addition to that, we have tried to model muscle deformations using geometric primitives. A small portion of the muscles is modeled and realistic results are obtained. This approach can be used for generating surface skin deformation, however, all the human musculature should be modeled.

ACKNOWLEDGMENTS

The data used in this study were obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217. We are greatly thankful to the people at Graphics Laboratory at the Carnegie Mellon University for publishing their motion capture database on the Internet.

REFERENCES

[1] J. E. Chadwick, D. R. Haumann, and R. E. Parent. Layered construction for deformable animated characters. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 243–252, New York, NY, USA, 1989. ACM Press.

[2] CMU. CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu, 2001.

[3] F. Dong, G. J. Clapworthy, M. A. Krokos, and J. Yao. An anatomy-based approach to human muscle modeling and deformation. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):154–170, 2002.

[4] FFMPEG. Ffmpeg. http://ffmpeg.mplayerhq.hu/, 2007.

[5] J. P. Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[6] Sang Il Park and Jessica K. Hodgins. Capturing and animating skin deformation in human motion. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 881–889, New York, NY, USA, 2006. ACM Press.

[7] Nancy Pollard. Description of motion capture file format. http://graphics.cs.cmu.edu/nsp/course/15-464/Fall05/assignments/StartupCodeDescription.html, 2005.

[8] Coenraad Frederik Scheepers. *Anatomy-base Surface Generation For Artivulated Models of Human Figures*. PhD thesis, The Ohio State University, 1996.