

Intelligent Toolkit for Procedural Animation of Human Behaviors

Paul Slinger
Carleton University
pslinger@connect.carleton.ca

Seyed Ali Etemad
Carleton University
s.a.etemad@gmail.com

Ali Arya
Carleton University
arya@carleton.ca

1 INTRODUCTION

The use of optical motion capture systems to produce high quality humanoid motion sequences has recently become increasingly popular in many areas such as high-end gaming, cinema movie and animation production, and for educational purposes. Motion capture studios are expensive to employ or are unavailable to many people and locales. For this reason, the reuse of existing motion capture data and the resulting high-quality animation is an important area of research. Strict motion capture data restricts users to pre-recorded movement that does not allow the addition of dynamic behaviors required in advanced games and interactive environment. Our procedural animation method offers animators high quality animations produced from an optical motion capture session, without incurring the cost of running their own sessions. This method utilizes a database of common animation sequences, derived from several motion capture sessions, which animators can manipulate and apply to their own existing characters through the use of our procedural animation toolkit. The basic animation types include walk, run, and jump sequences that can be applied with personality variations that correspond to a character's gender, age, and energy levels.

There are few animation systems that provide users with existing animation sequences that can be manipulated and updated to output a new and unique animation sequence. One current example is Endorphin 2.7 by Natural Motion the 3D industry's first dynamic motion synthesis software [1]. Endorphin uses interactive and adaptive behaviors with artificial intelligence, biomechanics and physics to create their animations in real-time. The software uses a standard set of supplied characters that have dimensions and joint limits based on anthropometric data sources [1]. The animations derived from the software are strictly reactive sequences resulting from the true physical interaction with the environment and physics based character behaviors. Although these behaviors are customizable, they lack a dimension of personality.

Previous work in the motion capture field has been focused on overcoming the shortcomings and complexities of these systems. Such areas of research include efforts to retargeting motion data to new characters [2], motion adaptation techniques and library construction [3], and optimizing the motion capture pipeline for real-time applications [4].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

FuturePlay @ GDC Canada, May 12-13 2009, Vancouver, Canada.
Copyright 2009 --- 1-XXXXX-XXX-0/00/0009...\$5.00.

2 THEORY OF MOTION SYNTHESIS

The output data provided by the motion capture system is in the form of 3D rotational angles in space which demonstrates the body poses for each frame. In addition to the 3D angles, a set of Cartesian measurements is also provided, showing the exact place of the center of the body with respect to the calibration origin. The overall complete output data is represented by A as shown in (1) where d_i^x represents the location of the x axis of the hip marker in the i^{th} frame and θ_i^j denotes the rotation of the j^{th} marker around the x axis in the i^{th} frame. Once both the training and test samples have been extracted, the database is constructed as explained in section 3.

$$A = \begin{bmatrix} (d_1^x, d_1^y, d_1^z) & (\theta_1^{x_1}, \theta_1^{y_1}, \theta_1^{z_1}, \dots, \theta_1^{x_n}, \theta_1^{y_n}, \theta_1^{z_n}) \\ (d_2^x, d_2^y, d_2^z) & (\theta_2^{x_1}, \theta_2^{y_1}, \theta_2^{z_1}, \dots, \theta_2^{x_n}, \theta_2^{y_n}, \theta_2^{z_n}) \\ \text{M} & \text{M} \\ (d_m^x, d_m^y, d_m^z) & (\theta_m^{x_1}, \theta_m^{y_1}, \theta_m^{z_1}, \dots, \theta_m^{x_n}, \theta_m^{y_n}, \theta_m^{z_n}) \end{bmatrix} \quad (1)$$

Different captured actions may vary in length (number of frames); therefore a series of time warping manipulations is applied to solve this problem. In simple terms, the signals for each axis of each marker are stretched or shrunk in time to create actions of same duration.

Subsequent to the time warping process, two different methods are employed for re-synthesis of motion. The first method is interpolating between existing motions. For instance, interpolation between a high energy walk and a low energy walk results in a medium ranged energy walk. The other technique used is by means of generating a linear transform function between two existing styles of action. The transform function once applied to a new action of same class but different style class, converts the style class of the action.

3 APPROACH TO DATABASE CREATION

To begin creating a database of animated movements, it was necessary to determine the types and variations of animations to be included. The animations intended for the motion database include basic animations that are commonly used for in-game animations or simple animation sequences. Examining various game design documents, default animation sequences for various game engines, animation tutorials, and articles relating to personality and behavior provided a basis for our database. The identified animation types include Walking, Running, Jumping, Idling, Strafing Left and Right, Crouching, and a Collapsing or Death sequence. For each animation type, a secondary list of variation extremes was also identified. These

variations add a level of personality and interest to the basic, unexaggerated animation types. Masculine to Feminine, Old to Young, Tired to Energetic, Happy to Sad, and Determined to Dreamy are these animation variation pairs.

After determining the required animation sequences for the database, each sequence was captured using a six-camera optical motion capture system. A range of varying actors were used in several different motion capture sessions in which each actor performed the animation types in the style of each variation extreme. This collection of raw motion data was later post-processed and used in the motion synthesis process to produce the final database of animation matrices.

4 ANIMATION TOOLKIT

Autodesk Maya is a node-based 3D modeling and animation software suite. Maya provides a scripting language called Maya Embedded Language (MEL) that allows direct access to the wide range of animation tools. This accessibility made Maya (and MEL) an ideal implementation platform for our procedural animation methods.

The toolkit employs a simple Graphical User Interface (GUI) that provides users with a range of animation controls. Organized sequentially, the controls are presented in four steps of our procedural pipeline. Step 1 allows the user to select which existing character set in the scene file, if there is more than one, to apply the new procedural animation. The dropdown menu produces a list of all the existing character sets in the scene graph and choosing an option will select the desired character hierarchy to be animated. Step 2 provides a utility to keyframe each joint for every frame, but can be optional depending on whether these keys already exist for the given character set. Set keys are required in order for new joint rotation values to be retained. Step 3 holds the primary procedures of the animation toolkit. There are two separate dropdown menus that can be used to select the type of animation (walk, run, or jump) and a corresponding animation variation (Masculine to Feminine, Old to Young, or Tired to Energetic). After selecting the two options from the dropdown menu, the user can use the slider control to apply a weight value between the two extremes of the variation. The automatic update button takes the weight percentage of the variation on the right-hand of the slider to the variation on the left-hand of the slider and applies it to the joint rotations character set, producing a new animation. Step 4, provides further control to the user by allowing them to apply any additional manual adjustment values to the newly animated character. The user can apply new rotation values and weighting for a single specific joint (or for all joints) at each individual frame (or all frames) by using the available controls.

In addition to the four primary steps, the user is also presented with visual feedback and animation playback controls. These include visual feedback to the user regarding the current frame being processed and ultimately the total elapsed time for the finished process and the standard Maya playback (Play/Stop, Skip to the End/Beginning and Step Forward/Back a frame). Figure 1 is a screenshot of the Animation Toolkit GUI in its initial startup state.

Using the Animation Toolkit, a user is able to select an animation type and apply a range of variations in order to derive a new animation sequence. As an example, if a user requires a walk sequence with a more masculine distinction, the toolkit is able to facilitate this action. By selecting the Walk type and the

“Masculine ↔ Feminine” variation from the two dropdown menus in Step 3, as well as a minimum weight value (i.e. 0.0), the resulting animation is a masculine walk cycle.

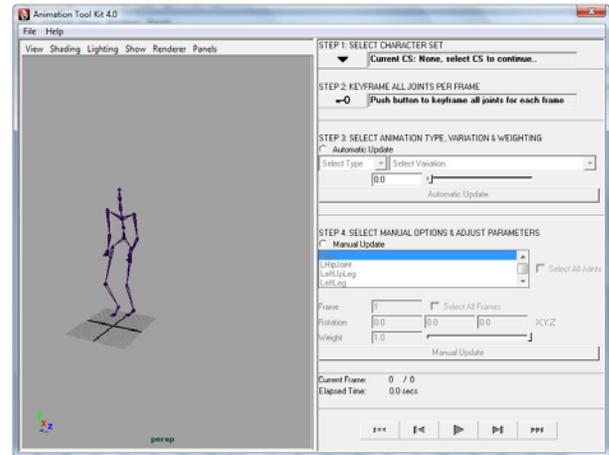


Figure1: Animation Toolkit GUI

However, adjusting the weighting to a maximum value (i.e. 1.0), results in the output of a feminine walk cycle. A user is able to select weight values between the minimum and maximum values that result in a unique mix between the two extremes. Selecting a weight value in the middle (i.e. 0.5) produces neither a masculine or feminine favored walk cycle, but an equal mix between the two variations.

The evaluation of the toolkit involves (1) allowing users (animators) to interact with the GUI and provide feedback and (2) comparing the procedurally generated animation with motion captured ones performing the same actions. Preliminary results show that the toolkit can be successful in creating a range of animated behaviors with an acceptable similarity to motion capture data methods.

Our current procedural method and preliminary tests produce expected animation results. Future extensions to our animation toolkit could include an increase in complexity and number of animation types and variations included in the database to include more uncommon animation types, an expansion of the additional manual controls (to include translation, blending, and other animation options), and translating our existing method from Maya to a designated gaming framework such as Microsoft's XNA for runtime execution.

5 REFERENCES

- [1] Endorphin 2.7 Spec Sheet. 2008. Retrieved June 30, 2008, from <http://www.naturalmotion.com/>
- [2] Zordan, V.B., Van Der Horst, N.C. 2003. Mapping Optical Motion Capture Data to Skeletal Motion Using a Physical Model. In Eurographics/SIGGRAPH Symposium on Computer Animation 2003.
- [3] Geng, W., Yu, G. 2003. Reuse of Motion Capture Data in Animation: A Review. In ICCSA 2003, LNCS 2669, pp. 620-629, 2003.
- [4] Cooper, S., Hertzmann, A., Popović, Z. 2007. Active Learning for Real-Time Motion Controllers. In ACM Transactions on Graphics, Vol. 26, No. 3, Article 5, Publication date: July 2007.