mira: musically interactive rendered animation

Ning Hu (ninghu@cs.cmu.edu)
Philo Chua (pchua@andrew.cmu.edu)
Kevin AuYoung (auyoung@cmu.edu)

Goal of the project

The goal is to build an animated figure that dances on stage with the rhythm/music using an existing set of motion capture data.

We want to simulate a stage dancer in a CG world. The whole system gets the music waveform [audio signal] as input, detects the beats, and generates the animation from the motion capture data. No two generated animation sequences will be the same because of different sets of music/rhythm, control, and environment generation.

Why the project is interesting and what work have others done in the area.

We are really interested in exploring the interactivity between different media. Computer animation and music should be able to interact in a smart way because both are structures in TIME. The idea for this project might be a starting point for developing new interactive techniques that merge different forms of media.

The project is focused on research in the interaction/perception/expression between computer animation and music. The ultimate goal of the project is to be played as an interesting game interacting with the music or even shown on stage as an art piece.

Related research

- Beat Detection
  Eric Schierer (MIT)
  He finished his Ph.D. thesis on music understanding in 2000. And he is famous for the beat tracking algorithm.

- Interactive Music Systems
  Masataka Goto "Cindy"
  He has done several years of research on "Cindy", which is a music-graphics interactive system. However, the character uses synthetic motion instead of captured motion from real persons.

  Simon Dixon
  Simon Dixon also did some research in interactive music system.

- Inspiration
  Lionhead Studios – Dancing Bear demo
  Space Channel 5 – video game by Sega
  Random sampling of hip-hop music videos
  Raw footage from dance clubs

All of these video games or eye-candies emphasize on music/rhythm and are well rendered.

What have you done? What previous work/software are you building on?

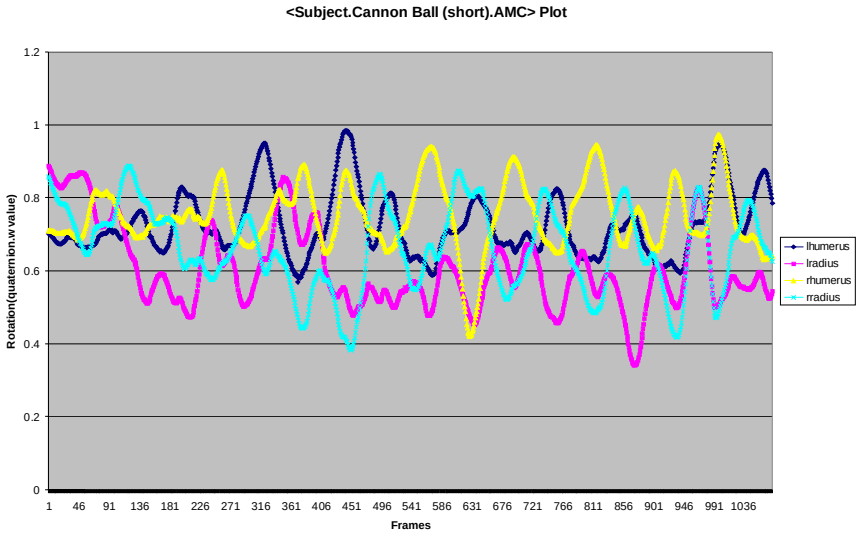
- Motion Capture

We captured a lot of dancing motions while the dancer was dancing with the real music. Those motions vary in length, ranging from 4-5 seconds to 20-25 seconds, using the 120 frames per second frame rate. We cleaned up 23 of these motions and ended up using 18 of them.

- Motion analysis and transition

We are using the interface and some source code from assignment one and two as the base of our system, but ported it to the windows platform.

After reading in all the dance motions, our system determines the rhythm and the starting points of the motions by counting the local minima/maxima of the motion. Below is the plot from a typical dance motion. The plot represents the scalar component of the orientation quaternion for individual body parts. Strong rhythms are obviously shown in the plot.



The system then creates a transition table between different dance moves. Those transition points are restricted to be on the beat of the motion rhythm.

We also used a simple method to avoid the dead ends so the transition tables are complete graphs. If the ending transition point does not have any good transition (as we use some constraints to quickly cut off unlikely transitions), then we force it to accept the branch that has the minimum comparison value as the transition. We also do not allow transitions to go to the ending transition points except their immediate ancestors.

- Beat tracking

Currently only music with strong rhythms are designed to be used for our system.

We've got some source code for beat tracking. But some are matlab code, some are written in java. None of them is easy to be integrated into our system. So we use the Winamp avs (advanced visualization plug-in) instead. The system gets the on-beat information of the music from the avs plug-in we wrote for Winamp and calculates the BPM (beats per minute).

$$\text{Current BPM} = (\text{average time of last 10 beats})^{-1}$$

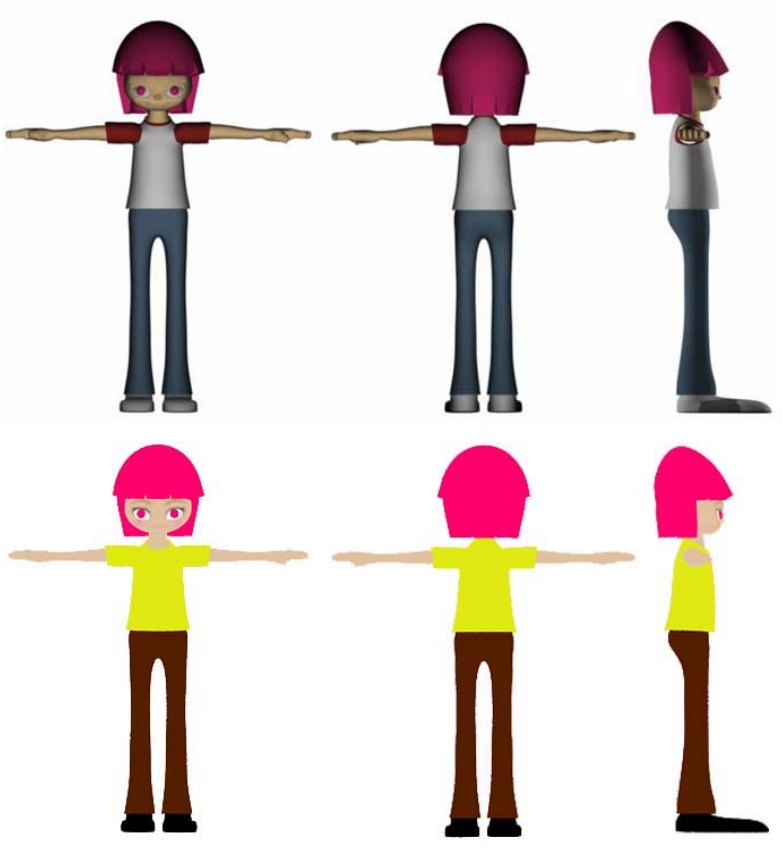
- Motion control and blending

We control and change the speed of the motion in real-time according to the BPM of the music and the pre-set display frame rate by using interpolation techniques. Transitions happen randomly. In order to smooth the transitions, we gradually blend between previous frames of the transition points before the motion actually jump.

For rendering purposes, the program can also save the generated motion to an AMC file.

- Rendering

For the offline rendering, we have modeled and textured a female character. She has been fitted with a skeleton from the ASF file. The final animation was rendered with Renderman using flat shading for a stylized look. Then we combined the video and audio clips into a single movie file with Adobe Premiere.



Demonstration of how well it works. [See accompanying CD]

In our video, we combined three songs with different BPM into a single minute clip. The dancer is clearly dancing on beat with the music, and the rhythm of the motions change when the music changes. However, in the last third of the clip where the music has a very high tempo (224 BPM, more than twice the original BPM of the captured motion), the dance motions are scaled to an unrealistic speed, causing the motions to look ridiculous. This shows that there is a speed limit for the captured motion. If the scaled speed were too different from the original motion speed, the result would not look natural.

Overall, you can observe that the system works fairly well. The transitions and the rhythms of the motions blend smoothly. For the music used in the demo clip, we observed an average of 9-15 jumps in motion.

What each person contributed.

Ning Hu

- Motion Capture
- Beat detection
- Motion Analysis
- Motion control and blending

Philo Chua

- Modeling and Texturing
- Motion capture and data cleanup
- Rendering
- NPR Research

Kevin AuYoung

- Musical Synthesis
- Movie generation
- Camera and Environmental Research

What you would do next if you had more time.

The project is very interesting and has a large scope, but was scaled down due to time constraints.

With more time, we would like to develop our own beat tracking module, because the winamp avs doesn't do a good job and it doesn't provide enough control for custom use.

Also, another area of interest is more and better motion analysis. By extracting characteristics from the motion like whether it is subtle or exaggerated, gentle or rigid, the system could better pick appropriate motions for different dance styles. The music analysis for the music would also need to be more intelligent than just checking for the beat. Combining these two, we can produce better motions appropriately interpreting the style/mood of the music.

The system currently renders offline via Maya and Renderman, but the motion can be generated and played back through a simple skeleton structure in real-time. By developing a real-time renderer for the system, it would be possible to employ MIRA in venues like a night club where all the DJ needs to do is make sure his music enters the system.

It could be argued that the flat shading used in the video isn't technically non-photorealistic rendering. It was a decision to use the flat shading available through Renderman because there were problems encountered while applying toon shading on the model. Given more time, a number of different rendering styles would be available to the system to use based upon the current piece of music. Pop music could be rendered with a more toon style while classical music could use a watercolor technique.

We would love to generate artistic environment but not enough time was spent on it. There are a number of things that can be done with environment. The timing in camera control, like cutting and panning would vary between musical styles, and that type of information would need to be available to the system. A higher level environmental control system would be responsible for moving lights, changing colors, and cutting between shots.

05/10/2002