CSC 2504F Computer Graphics
Graduate Project -Motion From Primitives

Alex & Philipp Hertel

December 2, 2002

Introduction

Most partner dances such as Salsa, Swing, Cha Cha, Merengue, and Lindy Hop are danced to music with eight beats per bar. Typically, dancers perform a series of 'moves' during each song, where each 'move' takes eight beats to complete. We will consider 'moves' to be the atomic building blocks out of which these dances are built. The idea is that certain moves lead very well into others and with only a few moves it is possible to put together many different combinations, which keeps the dance from becoming repetitive.

Each of these dances therefore has an underlying structure which completely captures it, namely a directed graph whose nodes represent moves, and whose directed edges represent valid transitions between those moves. When motion or animation data is included in such a graph, we call it a 'motion graph' (MG for short).

Motion graphs can be used in computer graphics to create quite sophisticated and realistic animations. For example, performing a random walk of a Salsa MG and outputting the corresponding moves to the screen as nodes are visited will give rise to characters that seemingly 'know' how to dance. Unlike simple repetitive animations which become boring after a few cycles, animations derived from motion graphs contain greater variety and do not become repetitive. If the motion data is obtained from a motion capture device and if the MG is designed correctly, the animations become increasingly realistic and sophisticated.

Of course, pretty animations are not the only benefit gained from motion graphs. They also provide a means of motion control, and have many more extensions and potential applications in industry, which will be discussed under the section below entitled 'Extensions'.

Project Overview

The purpose of this project is to explore the use of motion graphs for capturing and animating a partner dance called the Salsa, a Latin dance that has become very popular during the past few years. As already mentioned, Salsa is a couple dance that is danced on eight beats. Prior to starting this project, we had already developed a MG for Salsa. The subgraph of 'basic' Salsa moves is shown below (Figure 1).

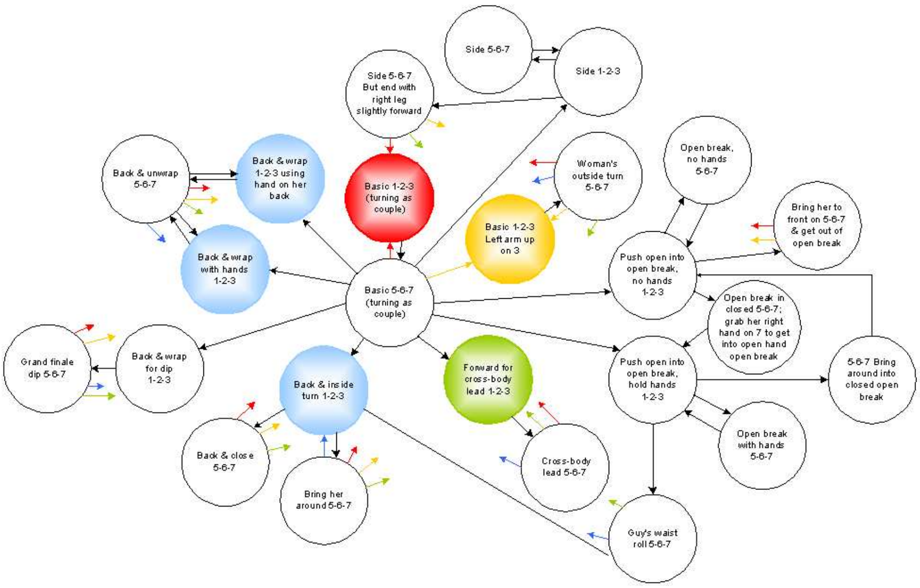

Figure 1: The Beginner's Salsa Motion Graph

In this MG, moves have each been split into two nodes, one representing beats 1,2,3,4, and the other representing beats 5,6,7,8. Although not strictly necessary, this subdividing of nodes makes the MG more expressive, and therefore more powerful. It also allows for some automated debugging to be done; any MG with split nodes must be bipartite, a condition that can be automatically checked.

The coloured nodes represent moves which are fundamentals of Salsa, in that they lead into many other moves. As this MG was hopelessly non-planar, colour-coding the high degree nodes allowed for 'shorthand' coloured edges to be created, thereby simplifying the embedding and making it more readable. Techniques such as this become more important as the complexity of the MG increases. Though a colour-coding scheme such as this has not been implemented in the current project it is clearly a useful UI feature.

At the heart of the Salsa dance is a move called the 'basic'. To use the Salsa MG, once the music begins, one starts at the node labelled 'basic 1-2-3 (turning as couple), and then follows edges until the music stops. The Salsa MG is robust, because it is connected. In other words, from any single node, there is a path to any other. This is important because it ensures that all motions are accessible at all times. This MG represents only the moves that are typically taught in a beginner's Salsa class, and therefore contains only a tiny subset of all possible Salsa moves.

For this project, our main objective was to implement an animation engine that could create and read serialized MG (.mg) files, and display animated stick figures dancing Salsa. We implemented this animation engine in C++ using OpenGL for the graphics, and Microsoft Foundation Classes (MFC) for the windowing. The implementation is not meant to be comprehensive, and contains only a small subset of all beginner's moves. The moves that were included were hand-modelled, and are only to demonstrate how MGs work. It is our intention to expand this project in the near future by adding motion captured data, thereby making the animation more complete and much more realistic.

Using The Application

The application is dialog-based, and consists of three dialog boxes:

1. The Main (animation) Dialog Box (Figure 2)

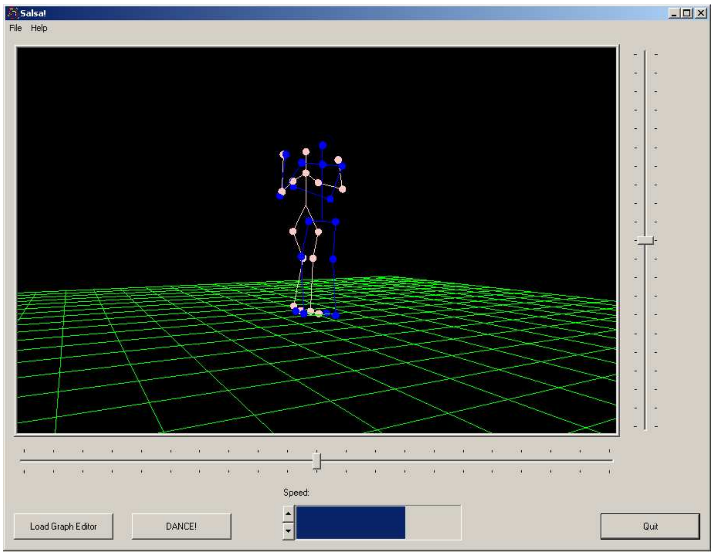

Figure 2: The Main Dialog Box

- This dialog box is used for viewing the animation resulting from randomly traversing the Beginner's Salsa MG.
- The main dialog box's menu and its quit button are operated in the familiar, intuitive way.
- The vertical slider allows you to zoom in on the dancers.
- The horizontal slider allows you to rotate the camera around the dancers.
- The button labelled 'DANCE!' will start the figures dancing, and change the label on the button to 'STOP!'. Repeatedly pressing on the button will toggle dancing on and off.
- The spin buttons below the label 'Speed:' will speed up and slow down the animation, and adjust the speed-indicator (progress bar) accordingly. The default speed is 1000 milliseconds per node, which is equivalent to 2 seconds per 8 Salsa beats. Each click on a spin button will increase or decrease the time by 50 milliseconds per node.
- The button labelled 'Load Graph Editor' will load the next dialog box:

2. The Graph Editor Dialog Box (Figure 3).

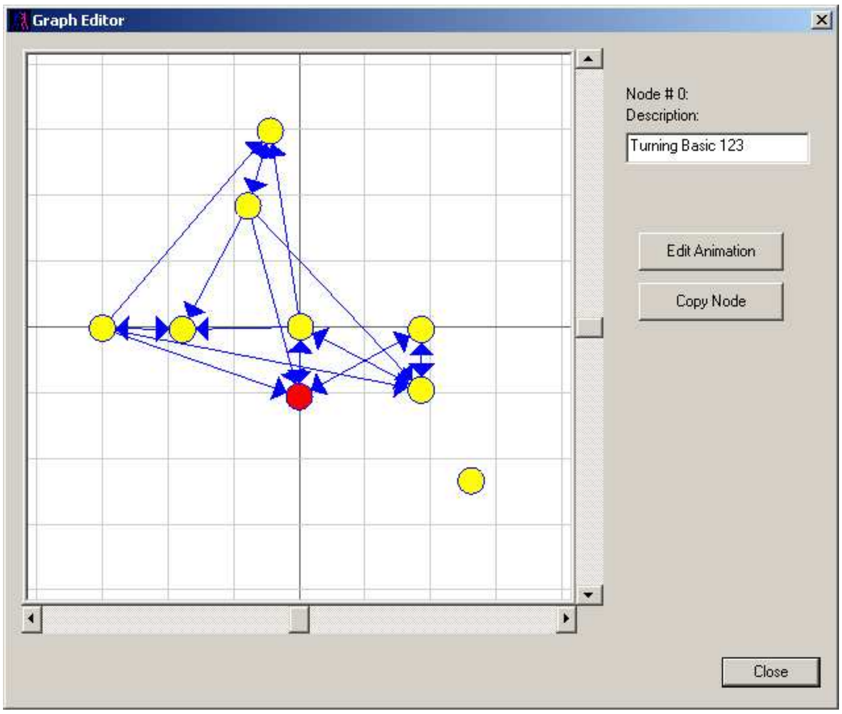

Figure 3: The Graph Editor Dialog Box

- This dialog box is used for editing the Salsa MG.
- To add a node, simply click once anywhere in the gridded canvas. Note that nodes cannot be superimposed on top of one another. Clicking and dragging has two different effects: If you left click on a node and drag to open space, it will draw an edge from the first node to a newly created node. If you left click in open space and drag to an already-existing node, it will create a new node with an edge leading to the second node.

- To select a single node, either left or right click on it. There are two ways to select multiple nodes: Use the right button to drag a bounding rectangle around the nodes you wish to select, or hold down the shift or control keys and click on different nodes to toggle them. Edges can be selected only by right clicking on them, not by the bounding box.
- To drag a node, simply press on it with the right mouse button and move it to its new location. Again note that nodes cannot be superimposed. To drag multiple nodes, select them, and hold the control or shift key down while dragging one of them. Their orientation will be preserved unless one is dragged onto an already-existing node.
- To delete a single node or a group of nodes, make your selection and hit the delete key. Likewise, to delete edges, first select them and then hit the delete key.
- To re-center the gridded canvas, double-right-click on an empty point. Note that since the Salsa MG for this project is small, the horizontal and vertical sliders are not yet functional.
- Information pertaining to the currently selected node is displayed in the upper right corner, which includes an edit box that can be used to change a node's description.
- The 'Copy Node' button will create a duplicate copy of the currently selected node and place it at the mouse pointer. If the mouse is not in the canvas, then a copy is made at the last point where the mouse pointer was in the canvas. The keyboard accelerator for this command is *ctrl* + *c*.
- The button labelled 'Edit Animation' will load the next dialog box. It can only be accessed when exactly one node is selected:

3. The Animation Editor Dialog Box (Figure 4).

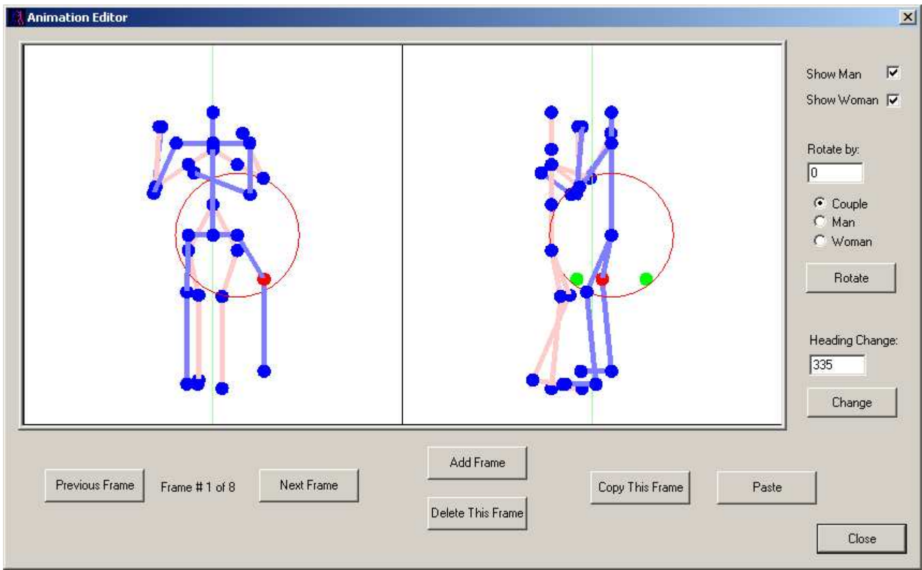

Figure 4: The Animation Editor Dialog Box

- This window is used for editing the animation frames contained by the nodes of the MG, and is the most complicated of the three dialog boxes. The idea is that with enough frames, it is possible to make each successive animation frame slightly different than the previous one in order to create smooth animation.

- Each node by default is created containing only one frame. A node can contain an arbitrarily large number of animation frames, but because Salsa has a steady rhythm, the amount of time taken to animate each node is constant. The 'Previous Frame' and 'Next Frame' buttons are used to scroll through the animation, frame by frame. The 'Add', 'Delete', 'Copy', and 'Paste' buttons work in the intuitive way.

- The stick figures are implemented as trees, rooted at the pelvic node. Other than the root, each node has a parent node.

- The canvas is broken up into a front and side view. The dancers are manipulated by directly moving their joints. Editing the stick figures' poses is a two step process. Firstly, select a joint node by clicking on it. If ambiguity exists, the ambiguous nodes will be coloured green in the other view. Simply click on one of them to resolve the ambiguity. Once a node is selected, a red circle will be drawn around the node's parent in both the front and side views. In reality, this is a sphere centered on the parent node. To move the selected joint node, drag it to any position on the surface of the sphere. This will create an ambiguity in the other view which must be resolved by clicking on a green node. The sphere guarantees that distances between parent and child joints do not change, which models the human skeleton realistically since bones are rigid. Moving a joint node will also move all of its children, which maintain their original relative positions. To translate a figure, simply select its root node and drag it to the desired position. Note that the figures can not be translated vertically because beginner's Salsa moves do not include any jumps.

- The rotation tools on the right hand side are used to rotate the figures clockwise in degrees.

- The 'Heading Change' tools on the right serve an important purpose. When dancing Salsa, certain moves cause the couple's orientation to change with respect to the dance floor. In order to seamlessly put together a series of moves, the orientation must be noted so that the next move can be rotated appropriately before being played. The 'Heading Change' tools allow the user to enter what the orientation of the couple is at the end of the move.

Design Decisions

The following were the most important design decisions made for this project:

- Instead of representing the stick figures using a combination of angles and positions in a similar way to Maya, joint nodes are represented using only x,y,z positions, and not angles at all. Although this necessitates the use of spheres for guaranteeing that distances between joints do not change, there are certain advantages. One of the major future extensions to this project is to import motion-captured data, which is typically represented using x,y,z coordinates, and not angles. This design decision was made to facilitate the future inclusion of motion capture data.

- The heading change controls in the animation editor dialog box may seem unnecessary, since it seems that updated headings can be calculated and updated automatically. This may be the case for beginner's Salsa moves, but with more advanced moves heading calculation become non-trivial. In order to allow for such moves, the heading control was left manual.

- Although many 3D modelling applications include three separate views, the animation editor dialog contains only front and side views. A top view was not included because two views are sufficient for unambiguously positioning the stick figures' joints in 3D space.

Extensions

The following are some possible extensions to this project:

Superficial / Simple Extensions:

- Make the Skeleton Class more general so that the project is not restricted to humans. For example, make it possible to create a stick figure for a horse, lion, or whale.
- Use the project to animate other activities, for example, Swing dancing, Karate, Kung-Fu, Sword Fighting, or any other motion that can be broken down into primitive building blocks.
- Put meat on the bones; expand the project so that it is not restricted to stick figures, but rather can handle any animated figures, perhaps using a standard format.
- Allow the MG to be more general by allowing the animations at different nodes to take different amounts of time, instead of being restricted to constant time per node the way dances are.
- Allow the user to colour the important nodes, and allow for 'shorthand edges' similar to those in Figure 1, so that the user interface is more friendly towards large motion graphs.
- Instead of random traversals of the MG, allow the user to control which transitions are used. The obvious application for this is the use of MGs in computer games and movies. Another option is to turn the edges into weighted edges, so that popular moves can be statistically favored.
- Allow for 'hard coded move sequences'; some move combinations flow very well, and the ability to represent this would make a MG more powerful.
- In this project, the MG nodes contain animations and the edges represent legal transitions. It is unclear whether this representation of motion is the most intuitive for every use. An alternative is to take the current graph's dual. That is the edges contain animations, and the nodes contain states. Adjust the graph editor dialog box so that the user can switch between these two representations with a simple click. This would allow for interesting extensions of the UI. For example, each edge in the graph could double as a slider that displays the edge's animation frame by frame as it is manipulated.

Fundamental / Non-trivial Extensions:

- This project has been designed from the very beginning to be extended using motion capture data to make the animations more realistic. Inclusion of motion capture data will give the additional opportunity of developing a clever user interface for integrating the data with the MG nodes.
- Although beat detection seems like a simple concept, the rhythm of Salsa music is not easy to catch. In fact, it takes most people many months to learn to interpret the beat properly. Including beat detection and other analysis would allow the figures to dance to any arbitrary song, and perhaps even start improvising. For example, a popular move is for the man to dip his partner as the song is ending; a simple analysis of the song being played would allow the figures to perform such dips.
- Integrate behavioral characteristics with the MGs. For example, if the male and female dancers each were to have their own MG, it would be possible to have the woman react behaviorally to the man's leads instead of the dance being totally scripted. Moving beyond dance, the inclusion of behavioral models greatly expands the scope of MGs. For example, assume two characters are fighting, and each has a sword and shield. A simple MG would give them a library of moves such as blocking with their shields, thrusting their swords, moving around, etc. However, the inclusion of a behavioral model would allow the characters to react to one another. For example, they could circle each other, attempt to block each other's sword thrusts, search for weaknesses, etc.

- One could integrate a notion of goals into a behavioral model. Setting goals which are in opposition to the MG would create more realistic behavior and motion. For example, on a crowded dance floor it is not always possible to perform a flawless move due to a lack of space. A character could be given the goals of avoiding collisions while trying to stay as close as possible to the motion described in the MG in order to capture a person's need to slightly improvise in these situations. The MG would therefore become a motion schema rather than a script. Another possible set of goals would be the addition of waypoints which the dancers try to travel through while dancing.

Conclusion

This project was a successful first attempt, and gives a basic demonstration of what MGs are and how they can be used, but it has really only scratched the surface. The fact that the 'Extensions' section above is so large is a testament to how much this project can still be improved. Many of the ideas related to the goal/behavior-oriented applications of MGs mark interesting areas for potential research and original work. We are currently pursuing these avenues of research.

Bibliography

L. Kovar, M. Gleicher, F. Pighin: Motion Graphs. Computer Graphics 2002.