Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors) *Posters and Demos*

Parametric Motion Graphs

Rachel Heck and Michael Gleicher

University of Wisconsin-Madison

Abstract

We present a motion synthesis technique that allows automated authoring of high-fidelity, controllable motion streams for interactive applications, such as video games and simulations. Our technique uses a new data structure called a parametric motion graph *(PMGs) that describes valid ways of producing streams of motion by concatenating motion clips dynamically generated through parametric synthesis. PMGs can provide accurate control of individual motion clips while generating seamless linear blend transitions between them in realtime. The key to our technique is using sampling methods to identify and represent good transitions between parameterized* spaces *of motion. Because PMGs contain a single, controllable node for each type of motion, they are highly structured, facilitating fast decision-making for interactive character control. We have created characters that seamlessly perform requested actions in realtime using different kinds of motion, such as cartwheeling and boxing.*

1. Introduction

In many interactive applications, such as video games and simulations, the movements of humanoid characters play an important role. Ideally, any motion synthesis method used in these applications should efficiently produce continuous streams of high-fidelity motions; be responsive to changing inputs; generate motions that accurately meet supplied constraints, such as the location where a character should punch or the direction a character should move; and allow easy authoring of new movements. Existing approaches for generating character motions make limiting tradeoffs between motion quality, accuracy, responsiveness, and ease of authoring. Methods used in practice for creating the motions in video games require extensive work to author the structures used for motion control, and the results are often limited in their movement quality and/or control accuracy. Alternatively, methods developed by animation researchers provide automated authoring of high-fidelity motions but fail to provide the accurate control, flexibility in movement types, or responsiveness demanded by interactive applications. The goal of this paper is to provide a motion simulation technique that allows automated authoring of accurate, controllable, high-fidelity motion streams for interactive applications.

We introduce *parametric motion graphs* (PMGs), an example-based motion synthesis data structure that provides easy authoring of high-quality motions while also supplying the responsiveness, precise control, and flexibility demanded by interactive applications. A PMG describes possible ways to generate seamless streams of motion by concatenating short motion clips generated through parametric synthesis. Parametric synthesis allows accurate generation of any motion from an entire space of motions, parameterized by a continuously valued parameter. For example, parametric synthesis can generate motions of a person picking up an item from any location on a shelf. While neither seamless motion concatenation nor parametric synthesis is a new idea, by combining both synthesis techniques, PMGs can provide accurate control through parametric synthesis and can generate long sequences of high-fidelity motions without visible seams using linear blend transitions. To date, these two motion synthesis methods have only been combined for very specific cases, such as walking, or through large amounts of manual labor in the gaming industry. This combination of motion synthesis methods make PMGs highly structured, facilitating efficient interactive character control. We have been able to easily author interactively controllable characters that can walk, cartwheel, punch, change facing direction, and/or duck in response to user issued requests.

The nodes of a PMG represent parametric motion spaces that produce short motions for given values of their continuously valued parameters. The directed edges of the graph encode valid transitions between source and destination motion spaces. While prior work on synthesis by concatenation has focused on representing seamless transitions between individual clips of motion, we face the problem of defining valid transitions between parameterized *spaces* of motions, where it is not often possible to transition from any motion in one motion space to any motion in another. For example, consider a parameterized motion space representing a person taking two steps, parameterized on curvature. One can imagine that this parameterized motion space can follow itself; a person can take two steps, and then take two more, and so on. However, a transition should not be generated between a motion where the character curves sharply to the right and another where the character curves sharply to the left; the resulting transition would not look realistic. Thus, the edges in a PMG must encode the *range* of parameters of the target space that a motion from the source space can transition to, as well as the correct way to make the transition between valid pairs of source and destination motions. The key challenge to PMGs is finding a good way to compute and represent these transitions. By approaching the problem from a sampling perspective, we provide an efficient way to compute and encode the edges of a PMG, allowing automated authoring and fast transition generation at runtime.

2. Method and Results Overview

To provide PMGs as a method for interactive character control, we have developed techniques for:

Building PMGs: By using a method based on sampling, we can efficiently locate and represent transitions between parameterized motion spaces.

Extracting Data from PMGs: Our representation of transitions allows fast lookup of possible transitions at runtime by using interpolation.

Using PMGs for Interactive Control: Because PMGs are highly structured, they facilitate the fast decision-making necessary for interactive character control. Furthermore, because motions are generated using parametric synthesis, they accurately meet requested constraints.

To illustrate the utility of PMGs, we give a concrete example. We have created a character that can be directed through an environment while walking with smooth turns. Using existing parametric synthesis methods, we first built a parametric motion space of a person walking at different curvatures. Next, we quickly built a PMG from this motion space using our sampling algorithm. The resulting graph contains a single node, representing the parameterized walking motion space, and a single edge that starts and ends at this node. This edge describes how to transition from the end of a generated walking clip to any generated walking clip in a subspace of the parameterized motion space. This simple structure provides an organized way of controlling the character. By translating a user's desired travel direction into desired curvature requests, we can synthesize a continuous stream of walking motion that reacts to a user's commands. This

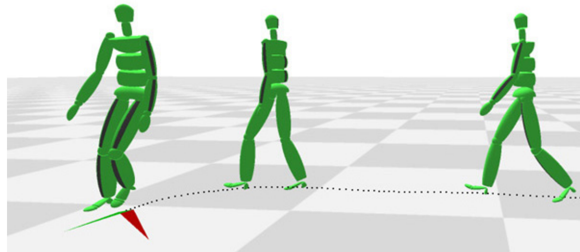

Figure 1: *An interactively controllable walking character using a PMG. The character is turning around in order to walk in the user requested travel direction depicted by the red arrow on the ground.*

stream of motion will run at interactive rates and only contain high-fidelity transitions between clips of motion. See Figure 1 to see a screen capture of our interactive walking character. Unlike other techniques, our technique requires little authoring effort, is capable of accurate motion generation through parametric synthesis, and works with a wide range of different motions. For instance, once we have an interactive walking character, it is easy to create a character that locomotes by cartwheeling, simply by building a parameterized motion space of cartwheeling motions, also parameterized on curvature.

3. Discussion

PMGs are able to produce seamless, controllale motion streams in realtime with little authoring effort. Yet, there are some limitations. Because our technique assumes that sampling and interpolation methods can describe how to transition between spaces of motions, it is important that these spaces of motions are smooth. This requirement is not a large limitation as many recent parametric synthesis methods produce motions by smoothly blending between example motions over parameter space. Furthermore, our method is limited to transitioning between motions only at one point near the end of a clip. Similarly, we do not adjust the parameter vector while generating a motion. These limitations mean that for parameterized motion spaces that represent long motions, it may take time for the character to react to user requests. This problem can be lessened by choosing parameterized motion spaces that represent short motion clips. Improving the responsiveness of our methods is future work.

Our technique shows that motions for interactive characters can be designed in an automated way, allowing fast, accurate, high-fidelity motion generation in realtime. Our method gains the benefits of accurate motion generation using parametric synthesis as well as the ability to make good transitions between clips using a continuous representation of transitions between parameterized spaces of motion. This technique can decrease the amount of time it takes to author interactive characters, increase the accuracy of these characters, and provide high-fidelity motion in a reliable way.