Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2012)
M. Lau (Posters Editor)

Dynamic Motion Graphs

Björn Krüger[1], Arno Zinke[2], Andreas Weber[1]

[1] Bonn University, Institute of Computer Science II, Computer Graphics
[2] GfaR mbH, Bonn

Abstract

In this work a novel method for synthesizing human motion from a sequence of sparsely distributed key frames is presented. The proposed framework allows even untrained users to quickly create believable-looking results using a mocap database. We show that our framework can handle challenging cases that include different motion classes and large spatial gaps between key frames.

1. Introduction

The synthesis of human motion sequences is a time-consuming task, even for trained artists. For that reason, being able to create a motion sequence from a rough, vague description is highly desirable. The goal of this work is to provide a uniform framework for synthesis of human motion from very sparse input data, i.e., sparsely distributed key frames.

Our metheod is comprising the following steps, cf.Figure 1:

- We first search a mocap database for motion segments that are suitable for connecting subsequent key frames.
- Retrieved motion segments are then used to locally build a motion graph-like structure connecting pairs of key frames.
- Eventually, a complete motion, comprising all key frames, is found by a shortest path algorithm on the previously computed motion graph. In the following we will refer to this result as *intermediate motion*.
- This motion, serves as a starting point for further improvements and may still contain blending or foot skating artifacts.

Therefore, the intermediate motion gets enhanced by a motion refinement step to obtain the final result.

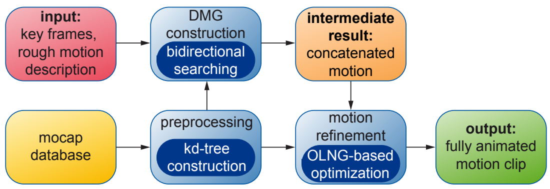

Figure 1: *Workflow of the proposed method for synthesizing motion sequences.*

2. Motion Synthesis

In this section we describe how the dynamic motion graph, that connects two subsequent key frames, is constructed on the fly.

Bidirectional search: The main idea of our bidirectional searching is to construct two trees: a tree $\mathbf{T}_{\text{out}}$ of outgoing motion segments of a first key frame $\mathbf{k}_i$ and a tree $\mathbf{T}_{\text{in}}$ of incoming motion segments to a second key frame $\mathbf{k}_{i+1}$. The two trees are then connected by blending suitable outgoing and incoming motion segments of the respective trees. The result is a directed, acyclic graph $\mathbf{G}$. We refer to this graph as *dynamic motion graph* (DMG). The DMG can be searched for an optimal motion connecting two key frames. The trees $\mathbf{T}_{\text{out}}$ and $\mathbf{T}_{\text{in}}$ are extended incrementally by employing a k nearest neighbor search around leaves (leaf poses) of each of the trees. From this search we obtain a ranked list of pose indices from the database that represent poses similar to a given frame according to Krüger et al. [KTWZ10]. More precisely, we use the feature set $\mathcal{F}_{\text{E}}^{15}$: the positions of the hands, feet and the head.

Based on the retrieved poses a leaf gets extended by a motion segment, given by the subsequent poses of the nearest neighbors stored in the database. Thus, we then get a set of outgoing motion segments $\mathcal{O}_i = \{O_{i,1}, \ldots, O_{i,k}\}$ containing all continuations. Similarly, a set of incoming motion segments can be defined: $\mathcal{I}_i = \{I_{i,1}, \ldots, I_{i,k}\}$ where the corresponding frames to a motion segment are denoted by

$O_i = [\mathbf{o}^1_{i,j}, \ldots, \mathbf{o}^{M_{i,j}}_{i,j}]$. $M_{i,j} \in \mathbb{N}$ is defining the length in number of frames of a motion segment.
Transitions between motion segments are enforced each time the state of ground contacts is changing. For that reason ground contacts are stored explicitly in the database on top of the motion data.
For building the graph we think of a node as a single dedicated frame and of an edge as a motion sequence connecting such frames. At every node a transition to another motion segment is possible. We start the construction process of the motion graph by setting a key frame $\mathbf{k}_i$ as root node of the tree $\mathbf{T}_{\text{out}}$ of outgoing motion segments and setting the key frame $\mathbf{k}_{i+1}$ as root node of the tree $\mathbf{T}_{\text{in}}$ of incoming motion segments. We then iterate the following steps until either the two trees can be connected, or a user defined number of iterations is reached.

1. *Tree continuation:* Find k outgoing motion segments $\mathcal{O}$ and k incoming motion segments $\mathcal{I}$ for the b best nodes according to a distance D_{node} for each of the two trees. Here, $D_{\text{node}}(\mathbf{a})$ is the Euclidean distance between the hands, the feet and the head of $\mathbf{a}$ to the closest node $\mathbf{b}$ in the other tree. In the first iteration there are only two such nodes, corresponding to the two key frames.
2. *Adding new nodes:* Store the end frames $[\mathbf{o}^{M_{i,1}}_{i,1}, \ldots, \mathbf{o}^{M_{i,k}}_{i,k}]$ of the outgoing segments and the start frames $[\mathbf{i}^1_{i,1}, \ldots, \mathbf{i}^1_{i,k}]$ of the incoming segments in nodes.
3. *Check for connections:* Check if an edge $O_{i,j}$ of the tree $\mathbf{T}_{\text{out}}$ of outgoing segments and an edge $I_{i,j}$ of the tree $\mathbf{T}_{\text{in}}$ of incoming segments are close enough to build a connection between the two trees.
4. *Update distances:* Recompute the distances D_{node} for all nodes.

If one or more connections between the trees are found we search for a path connecting two subsequent frames.

Connecting Edges: By connecting edges, the two trees $\mathbf{T}_{\text{out}}$ and $\mathbf{T}_{\text{in}}$ are merged to a full motion graph containing two subsequent key frames $\mathbf{k}_i$ and $\mathbf{k}_{i+1}$. Edges $O_{i,j}$ and $I_{i,j}$ are connected if some frames of the according motion segments are close enough. If a window of adjacent frames in two motion segments is found, we perform a linear blending for both, skeleton's root trajectory and rotational data.

Path Search and Intermediate Result: Once a dynamic motion graph $\mathbf{G}$ is constructed, the next step is to find an optimal path connecting the given key frames $\mathbf{k}$. Since the graph is directed and acyclic, a topological ordering of its nodes is directly given by construction and the optimal path can be computed in linear time [CLRS01] (chapter 24.2). This path is an intermediate result that gets refined in a subsequent step.

Refining the Intermediate Result: The simple concatenation of motion segments can lead to undesired artifacts, i.e. discontinuous movement, especially around the key frames. A second problem that arises is that blending of motion segments can introduce foot skating artifacts. Therefore, a further processing of the motion is necessary. For this purpose we employ a data-driven refinement along the lines of Krüger et al. [KZBW11].

3. Results

In this section we discuss selected examples of motions synthesized with our method.

Large Synthetic Examples: We computed several examples by extracting a set of key frames from motion capture recordings taken from the HDM05 database [MRC*07]. Every two seconds a key frame was extracted. The motion sequence from which the key frames were taken was deleted from the database. For results we refer to the video in the supplemental material.

Random Scenes: We generated example scenes by distributing four key frames (standing pose, cartwheel pose, jumping jack and standing pose) randomly in space. Solutions for a couple of these scenes based on the entire HDM05 database and CMU [Car12] database are presented in the supplemental video. Naturally, these cases are much more challenging, since they are not directly covered by the database examples.

4. Conclusion

In this work a technique for generating motion sequences from a few key frames was introduced. We can handle challenging cases including different styles of motions and large spatial distances between key frames. Moreover, our method is effective even for large mocap databases containing many different motion classes.

References

[Car12] CARNEGIE MELLON UNIVERSITY GRAPHICS LAB: CMU Motion Capture Database, 2012. mocap.cs.cmu.edu. 2

[CLRS01] CORMEN T. H., LEISERSON C. E., RIVEST R. L., STEIN C.: *Introduction to Algorithms*. MIT Press, 2001. 2

[KTWZ10] KRÜGER B., TAUTGES J., WEBER A., ZINKE A.: Fast local and global similarity searches in large motion capture databases. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (July 2010), SCA '10, Eurographics Association, pp. 1–10. 1

[KZBW11] KRÜGER B., ZINKE A., BAUMANN J., WEBER A.: Data-driven texturing of human motions. In *ACM SIGGRAPH ASIA 2011: Posters* (Dec. 2011). 2

[MRC*07] MÜLLER M., RÖDER T., CLAUSEN M., EBERHARDT B., KRÜGER B., WEBER A.: *Documentation: Mocap Database HDM05*. Computer Graphics Technical Report CG-2007-2, Universität Bonn, May 2007. www.mpi-inf.mpg.de/resources/HDM05. 2