Extended Abstract

International Journal of Innovative Research in Computer and Communication Engineering

2020
Vol.5 No.2.6

Computer Graphics 2015: Character Animation using Genetic Algorithms

Benjamin Kenwright

Terra State Community College, USA

The emergence of evolving search techniques (e.g., genetic algorithms) has paved the way for innovative character animation solutions. For instance, generating human movements `without' key-frame data. Instead character animations are often created using biologically inspired algorithms in conjunction with physics-based systems. While the event of highly parallel processors, like the graphical processing unit (GPU), has opened the door to performance accelerated techniques allowing us to unravel complex physical simulations in reasonable time frames. The combined acceleration techniques in conjunction with sophisticated planning and control methodologies enable us to synthesize ever more realistic characters that transcend pre-recorded ragdolls towards more self-driven problem solving avatars. While traditional data-driven applications of physics within interactive environments have largely been confined to producing puppets and rocks, we explore a constrained autonomous procedural approach. The core difficulty is that simulating an animated character is straightforward, while controlling one is difficult. Since the control problem isn't confined to human type models, e.g., creatures with multiple legs, like dogs and spiders, ideally there would be how of manufacturing motions for arbitrary physically simulated agents. This presentation focuses on evolutionary algorithms (i.e., genetic algorithms), compared to the normal data-driven approach. We explain how generic evolutionary techniques are ready to emulate physically-plausible and life-like animations for a good range of articulated creatures in dynamic environments. We help explain the computational bottlenecks of evolutionary algorithms and possible solutions, such as, exploiting massively parallel computational environments (i.e., graphical processing unit (GPU)).

The emergence of evolving search techniques (e.g., genetic algorithms) has paved the way for innovative character animation solutions. For instance, generating human movements without key-frame data. Instead character animations are often created using biologically inspired algorithms in conjunction with physics-based systems. While the event of highly parallel processors, like the graphical processing unit (GPU), has opened the door to performance accelerated techniques allowing us to unravel complex physical simulations in reasonable time frames. The combined acceleration techniques in conjunction with sophisticated planning and control methodologies enable us to synthesize ever more realistic characters that transcend pre-recorded ragdolls towards more self-driven problem solving avatars. While traditional data-driven applications of physics within interactive environments have largely been confined to producing puppets and rocks, we explore a constrained autonomous procedural approach. The core difficulty is that simulating an animated character is straightforward, while controlling one is more complex. Since the control problem isn't confined to human type models, e.g., creatures with multiple legs, like dogs and spiders, ideally there would be how of manufacturing motions for arbitrary physically simulated agents. This paper focuses on evolutionary genetic algorithms, compared to the normal data-driven approach. We demonstrate generic evolutionary techniques that emulate physically-plausible and life-like animations for a good range of articulated creatures in dynamic environments. We help address the computational bottleneck of the genetic algorithms by applying the tactic to a massively parallel computational environment, such as, the graphical processing unit (GPU). Realistic and interactive character animation is a crucial problem in computer games and virtual environments. Within the past decade, motion graph based methods are a standard approach to the present problem. The efficiency of graph traversing is that the primary bottleneck of this approach. More recently, reinforcement learning techniques are employed to construct character controllers. However, reinforcement learning cannot handle environments that haven't been experienced. Also, reinforcement learning is bedeviled by the curse of dimensionality. Hence, realtime online computing techniques are urgent for a good range of applications.

Data-driven character animation has been extensively explored within the past decade because it is efficient thanks to generate new animations supported motion capture examples. Early works proposed a replacement arrangement called motion graph to arrange motion data. Plausible transition points are identified as nodes and short motion clips are edges in motion graph. Top quality animations are often synthesized by carrying on various search methods on motion graph like branch-bound and randomized search methods. However, these methods are too slow for interactive avatar control as graph traversing is time consuming. Within the following years, motion graph was strengthened and augmented in several aspects. First, the connectivity was strengthened by building fully connected subgraphs between similar motion clips, by adding intermediate poses interpolated from similar motion clips or by caching good blending samples. Second, parameterization was applied to similar motion clips. Third, interpolation was added to realize precise control. Two main hurdles in interactive character animation are the massive computation effort and therefore the realtime restriction. Character animation requires an outsized amount of computation to seek out a string of motion clips which will be pieced together naturally. Meanwhile, applications like video games are sensitive to computation time, so all computation tasks must be finished in realtime. During this paper, we employ parallel computing techniques to deal with these hurdles. Each action sequence is evaluated by the sum of discounted costs over a finite horizon. The genetic algorithm is chosen to deal with our problem because we glance for fast parallel evaluation methods and genetic algorithms are ideal for parallelization. We implement our algorithm on CUDA which may be a parallel computing architecture on GPU. Taking advantage of the facility of recent GPU, top quality animations are often generated in real-time. As no precomputing is required, our algorithm is applicable to a good range of applications. Our algorithm extracts similar motion clips automatically from the input motion data. These similar motion clips are parameterized in order that they are often represented by low dimensional variables like turning angle and walking speed. Unlike the optimal controllers that evaluate each action in an infinite horizon, our controller selects each action supported its performance over a finite horizon. Theoretical analysis and experiments show top quality animations are often produced by short horizon planning.

This work is partly presented at 2nd International Conference and Expo on Computer Graphics & Animation on September 21-22, 2015 held at San Antonio, USA