On the Interface Between Steering and Animation for Autonomous Characters

Shawn Singh Mubbasir Kapadia Glenn Reinman
Petros Faloutsos

Abstract

Intelligence in autonomous agents is usually modeled with a hierarchy of layers, where each layer communicates with its neighboring layers through well-defined and often simplified interfaces. This is particularly true about the interface between crowd steering and motion synthesis – common interfaces include force vectors, velocity vectors, trajectory segments, or discrete behavior actions. By assuming one of these interfaces, the steering task and motion synthesis task are well isolated, but this also presents many problems. In particular, problems arise at the critical point where animations *are* the navigation decisions, for example, when a character chooses to side-step instead of turn, or when a character wants to fit through a tight space. In this project we propose that footprint data – position, orientation, and time duration – is a more appropriate interface. We illustrate this by comparing a behavior interface, vector interface, and a footprints interface. For the purposes of evaluating the interface, we assume "ideal" steering intelligence and "ideal" animation quality in all cases, by hand-constructing illustrative examples. Furthermore, we have developed a prototype space-time planner based on footprints, that allows precise steering in both space and time.

Keywords: footprints, agent navigation

1 Introduction

Almost all crowd steering algorithms use one of the following methods of output: force vectors, velocity vectors, center-of-mass trajectories, or discrete behavior actions. The output is then used as input to motion synthesis to produce an animation. However, these interfaces have the following disadvantages:

- *Limited locomotion constraints:* Very few crowd steering algorithms account for locomotion constraints. Trajectories may have discontinuous velocities, oscillations, awkward orientations, or may try to move a character during the wrong animation state, and these side-effects make it harder to animate the character intelligently. For example, a character moving forward cannot easily shift momentum to the right when stepping with the right foot, and a character would rarely side-step for more than two steps at a steady walking speed.

- *Limited navigation control:* Existing steering algorithms usually assume that motion synthesis will automatically know how to obey a vector interface. This is not the case – motion synthesis does not have enough information to choose appropriate subtle maneuvers, such as side-stepping versus reorienting the torso, stepping backwards versus turning around, stopping and starting, planting a foot to change momentum quickly, or carefully placing footprints in exact locations. These details are critical to depicting a character's local steering intelligence, and thus it is appropriate for steering to have better control.

We propose a steering algorithm that outputs sequences of footprints as the interface between

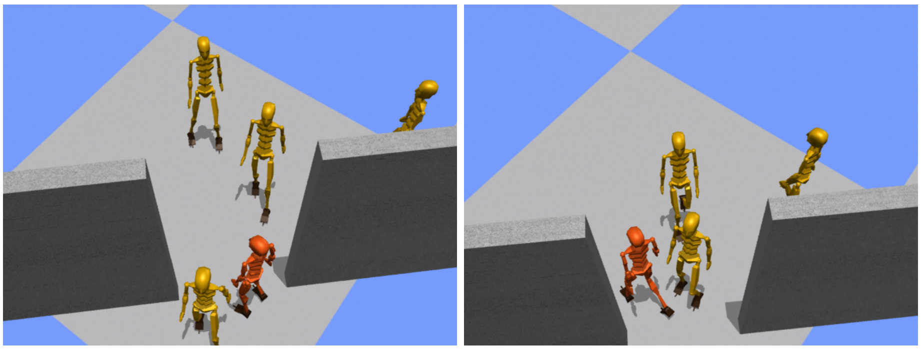

Figure 1: Several characters cooperating at a doorway.

steering and motion synthesis. Footprints, including location, orientation, and timing, are an intuitive and representative abstraction for most locomotion tasks. Footprints offer finer control which allows navigation to choose more natural and nuanced actions. At the same time, the series of footprints produced by navigation communicates precise, unambiguous spatial and timing information to motion synthesis. There already exist several motion synthesis algorithms that can animate a character to follow timed footprints exactly and the missing element is a crowd navigation algorithm that *generates* footprints that can be used by animation.

2 Related Work

There already exist several motion synthesis algorithms that can animate a character to follow timed footprints exactly, for example [1, 2, 3, 4, 5, 6, 7].

Fewer works have considered generating footprints for navigation. Footprint generation has been considered in robotics, e.g., [8, 9, 10]. These techniques focus on single robots being able to produce plausible footprints. This is a different challenge than trying to produce human-like footprints for crowds of agents. A few animation techniques have generated footprints for virtual agents. Van de Panne and Torkos [4] generated footprints to randomly wander, changing direction if nearby objects were too close; this was primarily used as a way to demonstrate their motion synthesis system. Chung and Hahn [5] generated footprints by aligning the next step to the orientation of the trajectory, using smaller footprints around curves. Coros et al. [6] demonstrate a physically-based character with the ability to carefully place footprints in exact locations. Zhang et al. [11] propose a hierarchical planning approach that compute stable footprints and upper body motion to solve manipulation tasks in highly constrained environments. In all these works, the focus was on generating footprints for robotics or animation of individual agents, usually in static environments.

A good survey of existing crowd steering techniques can be found in [12]. To our knowledge no prior steering algorithm has generated footprints for dynamic crowds.

3 Generating Footprints

We use a planning approach to dynamically generate a short sequence of footprints. The state of the character is represented by the location and orientation of the previous step, as well as the position and momentum of the character's center of mass. For each state, we define a set of footprints that the character can take next. Then, the planner searches for the sequence of footprints that minimizes energy to reach a short-term goal. Invalid steps, such as steps that cause collisions in space-time, are pruned from the search space.

Figure 1 illustrates results of our approach. The steering algorithm controls footprints, so the animations better reflect the human-like steering intelligence. Because the steering algorithm is based on short-term planning that considers other characters, agents also exhibit predictive cooperation. Agents can steer fluidly around each other, for example at doorways.

Since the planner is optimizing energy to reach the goal, it automatically finds whether it is appropriate to side-step or turn, depending on the situation.

We also compare the footprint interface with a vector interface as well as an interface of discrete behavior animations (for example, motion clips). The vector interface has problems when the vector changes directions suddenly – in general it is difficult for steering to communicate an exact desired movement using only a vector interface. For example, when a character tries to animate a turn using motion graph synthesis, there is often an unacceptable latency between the turning command and the actual character turning. The vector interface also has trouble distinguishing between movement and orientation. For example, it is often unclear if a brief turn should be interpreted as a side-step or as an actual turn. The behavior interface has difficulty with precise steering tasks, such as fitting through a narrow doorway. It would require too many discrete actions to cover all possible variations of a single task. This could be addressed by using parameterized blending of behaviors, but then these parameters become essentially like a vector interface. With the footprint interface, these problems are avoided.

References

[1] Michael Girard. Interactive design of 3d computer-animated legged animal motion. *IEEE Comput. Graph. Appl.*, 7(6):39–51, 1987.

[2] Heyongseok Ko and Norman I. Badler. Straight line walking animation based on kinematic generalization that preserves the original characteristics, technical report. 1992.

[3] Michiel Van de Panne. From footprints to animation. *Computer Graphics Forum*, 16(4):211–223, 1997.

[4] Nick Torkos and Michiel van de Panne. Footprint-based quadruped motion synthesis. *Proceedings of Graphics Interface '98*, pages 151–160, 1998.

[5] Shih-Kai Chung and J.K. Hahn. Animation of human walking in virtual environments. In *Computer Animation, 1999. Proceedings*, pages 4–15, 1999.

[6] Stelian Coros, Philippe Beaudoin, Kang Kang Yin, and Michiel van de Panne. Synthesis of constrained walking skills. *ACM Trans. Graph.*, 27(5):1–9, 2008.

[7] Golaem S.A. http://www.golaem.com, 2009.

[8] Joel Chestnutt, Manfred Lau, Kong Man Cheung, James Kuffner, Jessica K Hodgins, and Takeo Kanade. Footstep planning for the honda asimo humanoid. In *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2005.

[9] K.h. Nishiwaki, T. Sugihara, S. Kagami, M.y. Inaba, and H. Inoue. Online mixture and connection of basic motions for humanoid walking control by footprint specification. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 4110–4115, 2001.

[10] James Kuffner, K. Nishiwaki, Satoshi Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue. Online footstep planning for humanoid robots. In *IEEE Int'l Conf. on Robotics and Automation (ICRA'2003)*. IEEE, September 2003.

[11] Liangjun Zhang, Jia Pan, and Dinesh Manocha. Motion planning and synthesis of human-like characters in constrained environments. *Lecture Notes in Computer Science, Proceedings of MIG'09*, 5884/2009:138–145, 2009.

[12] Norman Badler. *Virtual Crowds: Methods, Simulation, and Control (Synthesis Lectures on Computer Graphics and Animation)*. Morgan and Claypool Publishers, 2008.