Computers & Graphics 36 (2012) 904–915

Contents lists available at SciVerse ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Graphics Interaction

Inverse kinodynamics: Editing and constraining kinematic approximations of dynamic motion

Paul G. Kry [a,*], Cyrus Rahgoshay [a], Amir Rabbani [a], Karan Singh [b]

[a] School of Computer Science, Centre for Intelligent Machines, McGill University, Canada
[b] Department of Computer Science, University of Toronto, Canada

ARTICLE INFO

Article history:
Received 2 July 2012
Received in revised form
26 August 2012
Accepted 29 August 2012
Available online 14 September 2012

Keywords:
Inverse kinematics
Secondary dynamics
Key framing

ABSTRACT

We present inverse kinodynamics (IKD), an animator friendly kinematic work flow that both encapsulates short-lived dynamics and allows precise space–time constraints. Kinodynamics (KD), defines the system state at any given time as the result of a kinematic state in the recent past, physically simulated over a short time window to the present. KD is a well suited kinematic approximation to animated characters and other dynamic systems with dominant kinematic motion and short-lived dynamics. Given a dynamic system, we first choose an appropriate kinodynamic window size based on accelerations in the kinematic trajectory and the physical properties of the system. We then present an inverse kinodynamics (IKD) algorithm, where a kinodynamic system can precisely attain a set of animator constraints at specified times. Our approach solves the IKD problem iteratively, and is able to handle full pose or end effector constraints at both position and velocity level, as well as multiple constraints in close temporal proximity. Our approach can also be used to solve position and velocity constraints on passive systems attached to kinematically driven bodies. We describe both manual and automatic procedures for selecting the kinodynamic window size necessary to approximate the dynamic trajectory to a given accuracy. We demonstrate the convergence properties of our IKD approach, and give details of a typical work flow example that an animator would use to create an animation with our system. We show IKD to be a compelling approach to the direct kinematic control of character, with secondary dynamics via examples of skeletal dynamics and facial animation.

1. Introduction

Physical simulation is now a robust and common approach to recreating reality in virtual worlds and is almost universally used in the animation of natural phenomena, ballistic objects, and character accessories like clothing and hair. Despite these strides, the animation of primary characters continues to be dominated by the kinematic techniques of motion capture and above all traditional keyframing. Two aspects of a primary character in particular, skeletal and facial motion, are often laboriously animated using kinematics.

We note from conversations with about half a dozen master animators that there are perhaps three chief reasons for this. First, kinematics, unencumbered by physics, provides the finest level of control necessary for animators to breathe life and personality into their characters. Second, this control is direct and history-free, in that the authored state of the character, set at any point in time, is precisely observed upon playback and its impact on the animation is localized to a neighborhood around that time. Third, animator interaction with the time-line is WYSIWYG (what you see is what you get), allowing them to scrub to various points in time and instantly observe the character state without having to playback the entire animation.

The same animators expressed the utility and importance of secondary dynamics overlaid on primarily kinematic character motion to enhance the visceral feel of their characters. Various approaches to such secondary dynamics have been proposed in research literature [1–3], some of which are available in commercial animation software. Overlaid dynamics, unfortunately compromise the second and third reasons animators rely on pure kinematic control.

A kinematic solution incorporating secondary dynamics called *kinodynamic* skinning [4] was suggested in the context of volume preserving skin deformations. With this approach, a kinodynamic state at any time is defined as a kinematic state in the recent past, physically simulated forward to the given time. In this paper we develop this idea of kinodynamics (KD) as a history-free kinematic technique that can incorporate short-lived dynamic

* Corresponding author. Fax: +1 514 398 3883.
E-mail address: kry@cs.mcgill.ca (P.G. Kry).

0097-8493/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cag.2012.08.010

behavior. Note that the above usage of the term ''kinodynamic'', while similar in spirit, is distinct from its use in the context of robot motion planning where it addresses planning problems where velocity and acceleration bounds must be satisfied [5].

The KD *window size* determines how far into the recent past we start a physical simulation in order to compute a KD state. We must formulate an appropriate KD window size for a given kinematic motion and physical parameters: both long enough to ensure a temporally coherent KD trajectory that captures the nuances of system dynamics, and short enough for interactive WYSIWYG computation and temporal localization of the influence of animation edits on system state. Many goal directed actions such as grasping, reaching, stepping, gesticulating, and even speaking, however, involve spatial relationships between the character and its environment, that are best specified directly, as targets state that the character (or parts of the character) must observe at given times. Techniques such as inverse kinematics (IK) and space time optimization algorithmically infer the remaining system states and animation parameters from these animator specified spatio-temporal targets. However, IK does not give the secondary dynamics, and space time optimization is typically computationally expensive. Analogous to these techniques, we develop an inverse kinodynamics (IKD) algorithm allowing animators to prescribe position and velocity constraints at specific points in time within a KD setting.

Kinodynamics is an interesting approach for interactive character animation, where animators can continue to leverage a direct history-free kinematic work flow, coupled with the benefits of arbitrary physically simulated secondary dynamics. The problem that we are solving is the inverse kinodynamics problem, and our solution allows an animator to easily edit kinodynamic trajectories such that desired constraints can be met. Specifically, the contributions of this paper are

- an iterative inverse kinodynamics (IKD) solver with fast convergence properties;
- details on how to implement our IKD solver for a wide range of scenarios: constraints on full poses or hands and feet in character animation, position and velocity constraints, multiple overlapping constraints, and constraints on passive deformable objects and in facial animation;
- an automatic method for selecting kinodynamic time windows from the physical parameters of the system and acceleration bounds on the kinematic trajectory.

We also discuss limitations, timings, convergence rates, and we describe a typical work flow example that an animator would use to create an animation with our system.

2. Related work

Secondary dynamics provides a significant amount of visual realism in kinematically driven animations and is an important technique for animators. In the case of tissue deformations produced by the motion of an underlying skeleton, various methods can be used to produce this motion through simulation or using precomputation [1–3]. With respect to secondary dynamics of skeletal motion, it has similarly been demonstrated that tension and relaxation of the skeletal animation can be altered through physically based simulation [6]. These techniques provide an important richness to an animation; while the style of the results are controllable by adjusting the elastic parameters or gains of controllers used for tracking, precise control of the motion itself to satisfy given constraints or keyframes is typically left as a separate problem.

In contrast to the simulations that provide secondary dynamics, it is the direct local control and WYSIWYG interface provided by forward and inverse kinematics techniques that animators primarily use in the creation of character animation. As a result, there has been a vast amount of research on inverse kinematics over the years, for instance, combining direct and inverse control during editing [7], using nonlinear programming [8], using priority levels to manage conflicting constraints [9], or alternatively, singularity-robust inverse computations [10] or damped least squares [11]. In this work our focus is on a problem similar to inverse kinematics, but in the new setting of kinodynamics. We leave the issue of solving inverse kinodynamics in the presence of conflicting constraints for future work.

The rest of the related work can be categorized into two groups. First, there are approaches which try to control a physically based simulation to have it meet some desired constraints. Second, there are approaches which use kinematic editing techniques to produce animations that meet desired constraints and exhibit physically plausibility.

Controlling physically based simulations is a difficult problem. There has been a significant amount of work in this area on controlling rigid bodies [12,13], fluids [14,15], and cloth [16]. Other recent successes on controlling physically based animation use gentle forces to guide an animation along a desired trajectory, accurately achieving desired states, but also allowing physical responses to perturbations [17]. Physically based articulated character control has received a vast amount of interest. Building on the seminal work of locomotion control [18], it is now possible to have, for instance, animation of physically based motions that respond naturally to perturbations [19–21], maintain balance during locomotion [22,23], and editable animations of dynamic manipulations that respect the dynamic interactions between characters and objects [24]. Our work is very different from these approaches, and is instead more closely related to work by Allen et al. [25], which changes PD control parameters to produce skeletal animations that interpolate keyframes at specific times. In our work, however, we keep the control parameters fixed and alter the kinematic trajectory.

Jain and Liu [26] show a method for interactively editing interaction between physically based objects and a human. In this work, it is the motion of the dynamic environment which is edited through kinematic changes of a captured human motion. In comparison, we focus on altering and editing a kinodynamic motion with different styles (tension and relaxation) and different constraints. Directly related to the problem of authoring motion, physically correct motion can be achieved by solving optimizations with space–time constraints [27]. Also relevant is work that uses analytic PD control trajectories for compliant interpolation [28], and work on generating physically plausible motion from infeasible reference motion using a dynamics filter [29].

In contrast to the work on controlling fully dynamic simulations, we are addressing a simplified problem due to the finite time window involved in simulating the state at a given time in a kinodynamic trajectory. This leads to benefits in the context of animation authoring, and allows for a straightforward solution to the inverse kinodynamic problem that we present in this paper. We use correction curves with a limited temporal width to solve the IKD problem. In a scenario with many constraints at different times, the correction resembles a smooth displacement function, such as the B-splines that are commonly used in solving space–time constraint problems (for instance, in the motion editing work of [30]). An important difference is that our displacements are applied on the reference kinematic trajectory, thus the final motion is the product of a physical simulation as opposed to a displaced motion that satisfies physical constraints. In a different approach, with similar objectives to our own work, Kass and