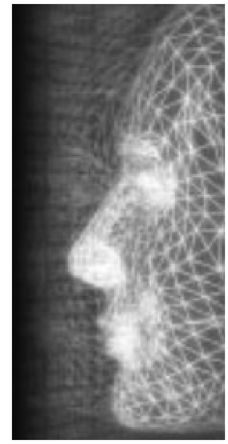
COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2007; **18**: 463–472
Published online 13 July 2007 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/cav.185



A steering model for on-line locomotion synthesis

*By Taesoo Kwon and Sung Yong Shin**

For applications such as video games and virtual walk-throughs, on-line locomotion control is an important issue. In general, the user prescribes a sequence of motions one by one while providing an input trajectory. Since the input trajectory lacks in human characteristics, one may not synthesize quality motions by blindly following it. In this paper, we present a novel data-driven scheme for transforming a user-prescribed trajectory to a human trajectory in an on-line manner. As preprocessing, we analyze example motion data to extract human steering behavior. At run-time, the input trajectory is refined to reflect the steering behavior. Together with an existing on-line motion synthesis system, our scheme forms a feedback loop, in which the user effectively specifies an intended human trajectory. Copyright © 2007 John Wiley & Sons, Ltd.

Received: 11 May 2007; Accepted: 14 May 2007

KEY WORDS: computer animation; character animation; motion control

Introduction

On-line, real-time locomotion synthesis is an important issue in applications such as video games and virtual walk-throughs. Although rich researches have been done in locomotion synthesis, the issue of on-line steering of human-like character has not been addressed well.

To get a concrete feel, consider Figure 1. Figure 1(a) and (b) show how user-specified trajectories (colored red) are different from actual human trajectories (colored green) during straight walking and running. The oscillations of the actual trajectories are due to the pelvis movements (rotations and translations) caused by supporting feet. Figure 1(c) and (d) show the variations of actual pelvis trajectories during curved walking and running. Such oscillations or curvature variations are the unique characteristics of human steering behavior. Simply placing the pelvis along a user-specified trajectory would not produce a natural motion. This immediately raises an issue: how to incorporate these characteristics into a user-specified trajectory.

For on-line applications, the user commonly prescribes a motion by interactively providing a motion type and its trajectory. In particular, the trajectory is specified either explicitly by a point stream that is sampled with an input device such as a mouse, or implicitly by a force profile that is given with a user interface equipped with slide bars (or a joystick). The former directly produces the trajectory of a human-like figure. Although it is easy to specify, the trajectory itself is neither precise nor smooth. Moreover, it is far from a natural human trajectory. On the other hand, integrating an input force profile that is sampled at each frame, the latter yields a smooth trajectory in an equally easy manner. However, the resulting trajectory is not natural, either. In either case, little effort has been made to produce a natural human trajectory.

No matter what method we employ, it would be difficult to generate high-quality locomotion with such a poor trajectory. In this paper, we present a data-driven method for refining a input trajectory for on-line, real-time locomotion synthesis, given the type of a locomotive motion. Choosing the center of the pelvis as the root of an articulated character, we describe how to yield a natural pelvis trajectory from the input trajectory. Without loss of generality, we assume that the input trajectory is given in an explicit form, that is, in the form of a point stream sampled at each frame. The refined trajectory gives the global pelvis position and orientation at each frame. The refinement is performed frame by frame in an on-line manner. Our method performs a two-step

*Correspondence to: S. Y. Shin, Korea Advanced Institute of Science and Technology, Taejon, Korea.
E-mail: syshin@jupiter.kaist.ac.kr

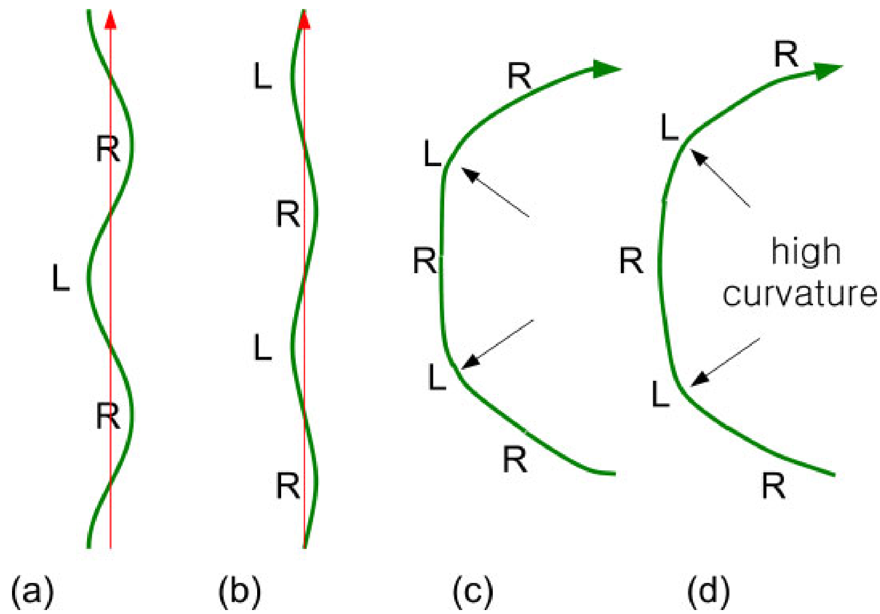


Figure 1. Pelvis trajectories. (a) Straight walking; (b) Straight running; (c) Curved walking; (d) Curved running.

refinement: first clamping speed and acceleration and then adding naturalness. For trajectory refinement, we make use of example motion data.

Related Work

Motion control has been a recurring theme in character animation, crowd simulation, and robotics. Our search focuses on the work directly related to on-line locomotion control.

Force-based Control

Reynolds[1] adopted a vehicle model to simulate typical steering behaviors for simple autonomous creatures in an on-line manner. These creatures were abstracted as simple vehicles, that is, oriented particles with their own dynamics. The steering behaviors were achieved by integrating steering forces. The vehicle model was further extended by incorporating real-time path planning.[2] We also use a vehicle model to obtain a smooth, feasible trajectory, and an exploiting example motion data.

Based on a social force model of Helbing and Molner,[3] Metoyer and Hodgins [4] simulated reactive pedestrians. Interpreting a pedestrian as a vehicle, the 2D trajectory of the pedestrian was generated by integrating the social force field. Teuille *et al.*[5] presented a more sophisticated potential field for the similar purpose. Guided by the potential field, motion capture data were used for motion synthesis. Based on force integration, the force-based approaches were able to yield smooth trajectories, although not allowing for a task-level on-line control. Moreover, it is hard to reflect natural human steering behavior with these approaches. We adopt the idea of force integration for trajectory refinement.

Position-based Control

In this category of approaches, a point stream is sampled directly for on-line locomotion control, based on motion blending. Early work by Guo and Robergé,[6] Wily and Hahn,[7] and Rose *et al.*[8] laid the ground for on-line motion control. Park *et al.*[9,10] extended the early work for on-line locomotion synthesis. Guided by a motion trajectory, they were able to synthesize a locomotion stream in an on-line, real-time manner. With a sequence of supporting foot positions sampled along a user-specified trajectory, their work demonstrated an on-line path-following capability. Mukai *et al.*[11] further extended this work based on geostatistics.

Recently, Kwon and Shin[12] presented a framework of on-line, real-time locomotion synthesis. This framework

Comp. Anim. Virtual Worlds 2007; **18**: 463–472
DOI: 10.1002/cav

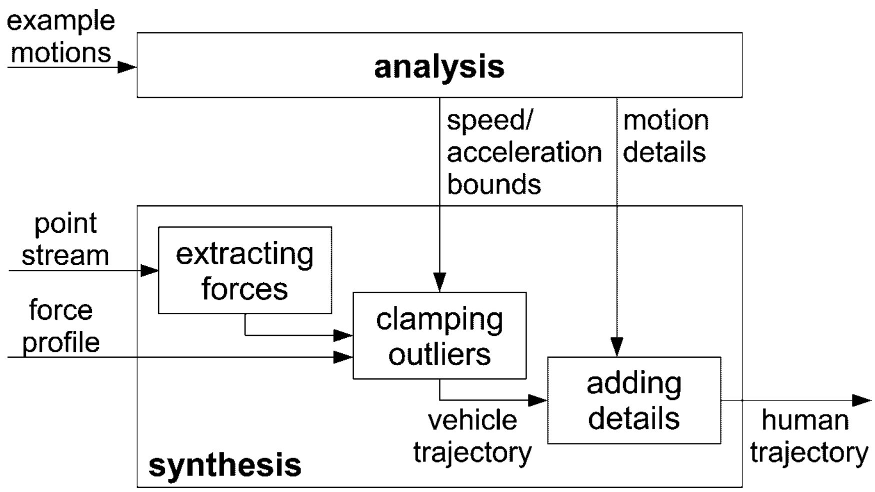


Figure 2. Block diagram.

automated the whole process of locomotion modeling and synthesis, including motion segmentation and classification. The pelvis trajectory was specified in an on-line manner by integrating a stream of 2D displacements on the ground.

In general, a position-based approach was not able to yield a smooth trajectory. In addition, the trajectory was also lacking in naturalness. We refine such a trajectory by imitating the example motion data.

Overview

As illustrated in Figure 2, our scheme for on-line character steering consists of two major components: analysis and synthesis. In analysis, characteristics of human steering behavior are extracted from example motion data. The extracted information includes bounds on speed and acceleration along pelvis trajectories and details of trajectories.

In synthesis, an input point stream goes through three steps for trajectory refinement: force extraction, clamping outliers, and adding details. The first step extracts a force profile from an input point stream. When a force profile is given as the input, this step is skipped. The rational for force extraction is that smoothness of a trajectory is easy to achieve with a force profile as long as it is continuous. In the second step, the force profile is integrated to produce a vehicle trajectory, while clamping speed and acceleration at each frame. The last step adds details to the vehicle trajectory to produce a natural pelvis trajectory. The three steps are performed frame by frame in an on-line manner, while referring to the information extracted during analysis.

Given a sequence of local joint configurations for a prescribed motion, the projected pelvis position is constrained to follow the refined trajectory. For ease of explanation, we assume the framework for on-line locomotion synthesis given in Reference [12] with some sacrifice of generality. However, this can be fixed easily, given a locomotion synthesis scheme.

Analysis

In this section, we analyze the pelvis trajectory of an example motion stream to obtain the trajectory of a simplified vehicle and motion details along the vehicle trajectory. A pelvis trajectory can be reproduced by adding the motion details to the vehicle trajectory. In this sense, the trajectory analysis stage can be viewed as the inverse of the synthesis process. We also extract bounds on speed and acceleration for clamping outliers along the vehicle trajectory.

Vehicle Model

For ease of control, we abstract a human-like character as a particle with an orientation. The particle is constrained to move on the floor. The orientation of the particle is aligned with its velocity. Such a particle is called as a vehicle.[1,2,13]

The state of a vehicle at time t is specified by an ordered tuple, $(\mathbf{x}^t, \dot{\mathbf{x}}^t, \theta^t)$, where $\mathbf{x}^t$, $\dot{\mathbf{x}}^t$, and θ^t denote the position,

DOI: 10.1002/cav

velocity, and orientation of the vehicle, respectively. The orientation of a vehicle θ^t is defined as a unit quaternion:

$$\theta^t = \left(\mathbf{h} \cdot \frac{\dot{\mathbf{x}}^t}{\|\dot{\mathbf{x}}^t\|}, \mathbf{h} \times \frac{\dot{\mathbf{x}}^t}{\|\dot{\mathbf{x}}^t\|}\right), \qquad (1)$$

where $\mathbf{h}$ is the unit halfway vector between $(0, 0, 1)^{\mathrm{T}}$ and $\dot{\mathbf{x}}^t/\|\dot{\mathbf{x}}^t\|$. Assuming that the vehicle has a unit mass, the state of the vehicle evolves as follows

$$\dot{\mathbf{x}}^t = \dot{\mathbf{x}}^{t-1} + \mathbf{f}^t; \mathbf{x}^t = \mathbf{x}^{t-1} + \dot{\mathbf{x}}^t,$$

where $\mathbf{f}^t$ denotes the force exerted to the vehicle at time t. When $\dot{\mathbf{x}}^t$ is a zero vector, θ^t is not well-defined. In this case, θ^t is set to the most recent valid orientation before time t.

Example Vehicle Trajectories

A single locomotion stream consists of one or more cyclic motions such as walking and running together with transition motions between different kinds of motions. The whole stream is delimited by a pair of standing motions. Without loss of generality, there is no intervening standing motion in the stream. For a cyclic motion, the characteristics of steering behavior are repeating at the frequency of a full cycle. We make use of the cyclic nature of locomotion to extract a vehicle trajectory from the pelvis trajectory for the motion stream.

As illustrated in Figure 3, our basic idea is to construct a pair of curves, called L-trajectory (colored red) and R-trajectory (colored blue), and then to average these curves to obtain a vehicle trajectory (colored black). In order to compute L-trajectory, our scheme samples

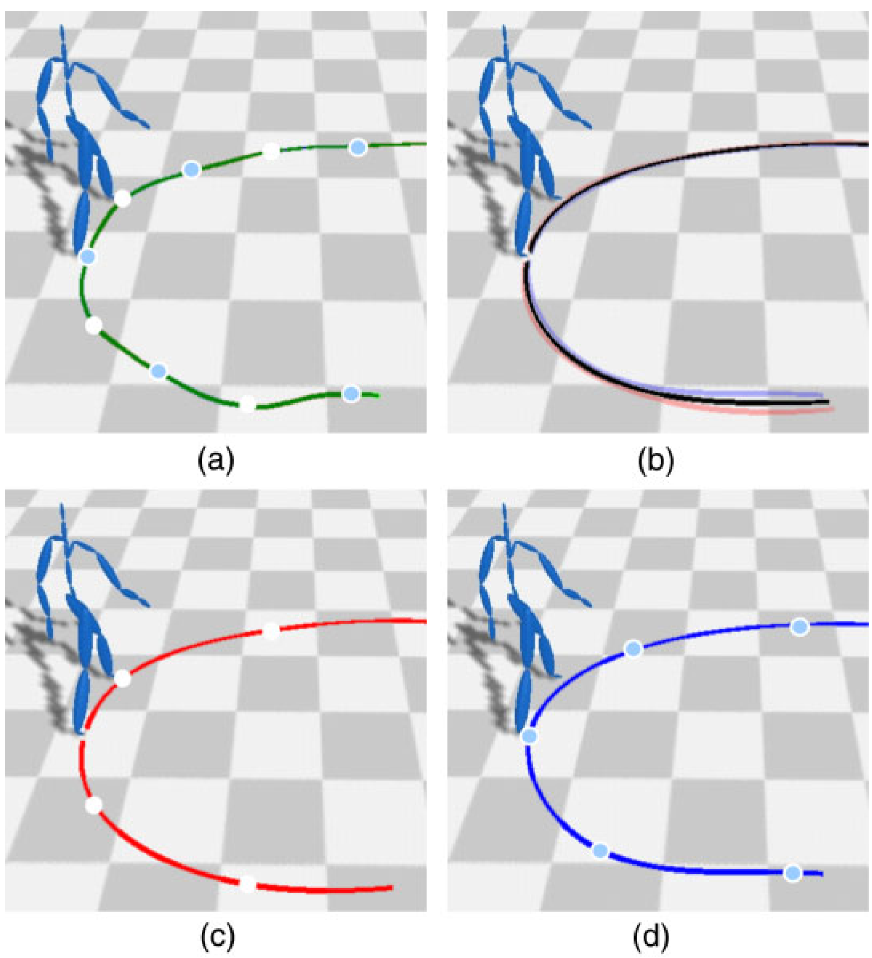


Figure 3. Vehicle trajectory estimation. (a) Pelvis trajectory; (b) Vehicle trajectory; (c) L-trajectory; (d) R-trajectory.

the position of the pelvis at the first frame of every cycle (L-cycle) that is initiated by a left supporting foot. An L-cycle starts at the middle frame of a left supporting foot phase. Let $\{y(t_i)\}_{i=0}^{n}$ be the sequence of all pelvis positions projected onto the ground. $t_i, 0 \leq i \leq n$ denotes the time (frame) at which the ith position is sampled. We find a non-uniform cubic spline curve interpolating $\{y(t_i)\}_{i=0}^{n}$ with its knot sequence $\{t_i\}_{i=0}^{n}$ to obtain L-trajectory (see Reference [14] for details). R-trajectory can be constructed in the symmetrical manner.

Let $\mathbf{x}_{\mathrm{L}}(t)$ and $\mathbf{x}_{\mathrm{R}}(t), t_0 \leq t \leq t_n$ denote L-trajectory and R-trajectory, respectively. Both curves, $\mathbf{x}_{\mathrm{L}}(t)$ and $\mathbf{x}_{\mathrm{R}}(t)$ are parameterized with the frame numbers of the same motion stream. Moreover, the curves share the same knot values at the end points. Therefore, their average is well-defined. Our vehicle trajectory, denoted by $\mathbf{x}_e(t)$, is obtained by averaging the two curves: $\mathbf{x}_e(t) = 1/2(\mathbf{x}_{\mathrm{L}}(t) + \mathbf{x}_{\mathrm{R}}(t))$. Since both $\mathbf{x}_{\mathrm{L}}(t)$ and $\mathbf{x}_{\mathrm{R}}(t)$ are C^2-continuous, $\mathbf{x}_e(t)$ is also C^2-continuous. As shown in Figure 3(b), the vehicle trajectory, $\mathbf{x}_e(t)$ (colored black) is smooth, has little oscillations and curvature variations, and lies between L-trajectory $\mathbf{x}_{\mathrm{L}}(t)$ and R-trajectory $\mathbf{x}_{\mathrm{R}}(t)$.

Motion Details

Choosing the center of the pelvis as the root, the pose of a character at every frame i is represented as a tuple $(\mathbf{p}_0^i, \mathbf{q}_0^i, \mathbf{q}_1^i, \ldots, \mathbf{q}_m^i)$, where $\mathbf{p}_0^i$ and $\mathbf{q}_0^i$ respectively denote the position and orientation of the root at frame i, and $\mathbf{q}_j^i, 1 \leq j \leq m$ is the orientation of joint j at frame i. All orientations are specified in unit quaternions. Given the joint configuration $(\mathbf{p}_0^i, \mathbf{q}_0^i, \mathbf{q}_1^i, \ldots, \mathbf{q}_m^i)$, the details of the pelvis trajectory consist of the position and orientation displacements between the pelvis (root) and the vehicle at all frames.

In order to represent the orientation displacement at each frame i, the pelvis orientation $\mathbf{q}_0^i$ at frame i is decomposed into two parts, that is, the rotational part about the vertical axis denoted by $\mathbf{q}_{\mathrm{vert}}^i$ and the remainder $\mathbf{q}_{\mathrm{offset}}^i$ such that $\mathbf{q}_0^i = \mathbf{q}_{\mathrm{vert}}^i \cdot \mathbf{q}_{\mathrm{offset}}^i$. Here, $\mathbf{q}_{\mathrm{offset}}^i$ are expressed in a coordinate invariant manner. The vertical orientation displacement between the pelvis and the vehicle is given by $d\mathbf{q}_{\mathrm{vert}}^i = \mathbf{q}_{\mathrm{vert}}^i \cdot (\theta_e^i)^{-1}$ therefore,

$$\theta_e^i = \left(d\mathbf{q}_{\mathrm{vert}}^i\right)^{-1} \cdot \mathbf{q}_0^i \cdot \left(\mathbf{q}_{\mathrm{offset}}^i\right)^{-1}$$

or

$$\mathbf{q}_0^i = d\mathbf{q}_{\mathrm{vert}}^i \cdot \theta_e^i \cdot \mathbf{q}_{\mathrm{offset}}^i$$

Thus, the orientation displacement is characterized by $(d\mathbf{q}_{\mathrm{vert}}^i, \mathbf{q}_{\mathrm{offset}}^i)$.

The position displacement between the pelvis and the vehicle at frame i is given by $\mathbf{p}_0^i - \mathbf{x}_e^i$ where $\mathbf{x}_e^i = \mathbf{x}_e(i)$. We represent it in the local coordinate of the vehicle for effectively adding details to the vehicle trajectory:

$$(0, d\mathbf{x}_e^i)^{\mathrm{T}} = \left(\theta_e^i\right)^{-1} \cdot (0, \mathbf{p}_0^i - \mathbf{x}_e^i)^{\mathrm{T}} \cdot \theta_e^i$$

where $d\mathbf{x}_e^i$ is the position displacement in the local coordinate frame. In summary, the motion details at each frame i is given by

$$(d\mathbf{x}_e^i, d\mathbf{q}_{\mathrm{vert}}^i, \mathbf{q}_{\mathrm{offset}}^i)$$

Now, a final remark is in order: When an example motion stream is segmented, the extracted vehicle trajectory and the motion details are also segmented accordingly. Thus, every motion segment has its own vehicle trajectory and motion details.

Speed and Acceleration Bounds

A raw-captured vehicle trajectory suffers from undesirable characteristics: lack of smoothness, unrealistically large speed and acceleration, and sudden changes in the force profile. In this section, we describe how to learn bounds on speed, acceleration, and inter-frame acceleration changes from example motion data to clamp outliers for later trajectory refinement.

Interframe Acceleration Variations. In our model, a vehicle is an oriented particle with a unit mass. Therefore, $\ddot{\mathbf{x}}_e^t = \mathbf{f}^t$ for all t, where $\mathbf{f}_t$ is the force that is applied to the particle at time t. The acceleration difference between a pair of consecutive frames reflects an instantaneous force change, which should be bounded within human ability, that is,

$$\left\| \ddot{\mathbf{x}}_e^t - \ddot{\mathbf{x}}_e^{t-1} \right\| \leq u_f.$$

The bound u_f for a group of example motion segments is obtained by taking the maximum magnitude of acceleration differences between consecutive frames for all motion data in the group.

Acceleration Bounds. Since a vehicle is constrained to move on the floor, an acceleration $\ddot{\mathbf{x}}_e$ has two components : a tangential component a_z and a lateral

DOI: 10.1002/cav

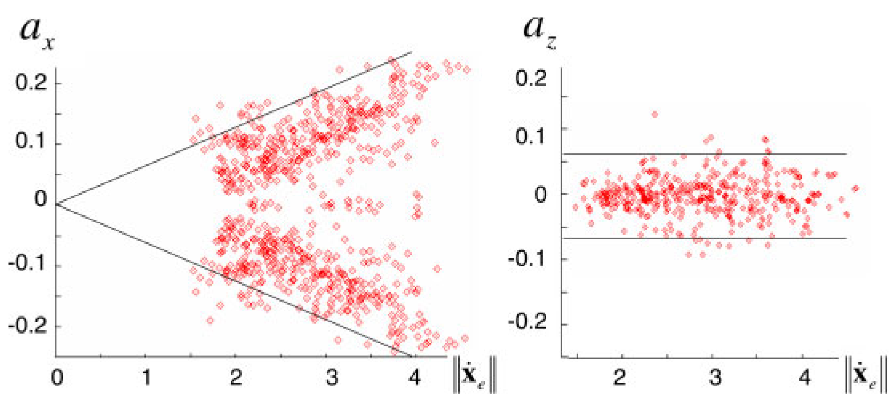


Figure 4. Acceleration bounds.

component a_x, where

$$a_z = \ddot{\mathbf{x}}_e \cdot (\dot{\mathbf{x}}_e/\|\dot{\mathbf{x}}_e\|), \text{ and}$$
$$a_x = \ddot{\mathbf{x}}_e \cdot \left[(0,1,0)^{\mathrm{T}} \times (\dot{\mathbf{x}}_e/\|\dot{\mathbf{x}}_e\|)\right] \qquad (2)$$

Since $\ddot{\mathbf{x}}_e^t = \mathbf{f}^t$, $\ddot{\mathbf{x}}_e^t$ is bounded for all t. Thus, both a_x and a_z is bounded. For cyclic motions such as walking and running, the two components show quite different behaviors with respect to the vehicle speed $\|\dot{\mathbf{x}}_e\|$ as shown in Figure 4. The tangential component is bounded within a fixed interval that contains the average regardless of $\|\dot{\mathbf{x}}_e\|$, while the variations of the lateral component are amplified as $\|\dot{\mathbf{x}}_e\|$ increases. a_x represents a centripetal acceleration when the vehicle moves along a circular path. For such a circular motion, the turning speed is given by $a_x/\|\dot{\mathbf{x}}_e\|$. Since the turning speed is not arbitrarily large, a_z is linearly bounded by $\|\dot{\mathbf{x}}_e^t\|$. This supports our intuition that pelvis rotations result in motion details along the vehicle trajectory.

Based on this observation, we estimate different bounds on these components:

$$\bar{a}_z - c_z \le a_z^t \le \bar{a}_z + c_z \text{ for all } t, \text{ and}$$
$$|a_x^t| \le c_x \cdot \|\dot{\mathbf{x}}_e^t\| \text{ for all } t$$

Here, $\bar{a}_z$ is the average of the tangential components for all frames in the example motion data. c_z and c_x are chosen so that 90% of the tangential and lateral components lie within the bounds.

For transition motions such as standing-to-walking and walking-to-running, the characteristics of a motion at the both extreme frames reflect different types of motions. Thus, we estimate the bounds of the components at the both extremes and interpolate the results to obtain the time-varying bounds.

Speed Bounds. For cyclic motions such as walking and running, the speed variations over time is insignificant, assuming that motion segments are short, for example, spanning a half cycle[12] or a full cycle.[9,11] In this case, speed variations could be linearly approximated even for transition motions. Thus, both upper and lower speed bounds for a group of motion segments are estimated at the initial and final frames, from which the time-varying speed bounds over all frames are obtained by linear interpolation. At the initial frame, the speed bounds can be acquired by finding the minimum and maximum speeds among all initial frames of motion segments in the group. Those at the final frame can also be computed symmetrically.

Synthesis

In this section, we describe how to refine a stream of raw-sampled 2D positions to obtain a human trajectory. The 2D positions are sampled at the rate of 30 Hz in an on-line manner with an interactive input device such as mouse. We assume that the example motion data are decomposed into a motion segments, which were classified into groups according to their logical similarity.[12,15] We also assume that each motion segment is parameterized by a motion type and the speed and acceleration at the initial frame. Thus, the sequence of local joint configurations of a user-prescribed motion segment is determined immediately after first trajectory position is sampled.

Force Extraction

Without loss of generality, we use a mouse to sample a point stream. We convert the raw-captured point stream

DOI: 10.1002/cav

into a force profile for smooth trajectory construction. Let $\mathbf{x}^0$ be the vehicle position at the last frame of the previously synthesized motion segment. When $\mathbf{x}^0$ is not available (the first motion segment), the user should sample an extra position$\mathbf{x}^0$. Suppose that cursor position $\mathbf{S}^t$ at time t has just been sampled. Then, $\mathbf{S}^t$ is transformed to the force that is to be exerted to the vehicle model for a character. We employ a spring-damper model to transform $\mathbf{S}^t$ to a force:

$$\mathbf{f}^t = \alpha(\mathbf{S}^t - \mathbf{x}^{t-1}) - \beta\dot{\mathbf{x}}^{t-1}$$

Coefficient α and β are chosen empirically.

Clamping Outliers

We now describe how to obtain a vehicle trajectory in an on-line manner by force integration while clamping the inter-frame acceleration change, acceleration, and speed. As defined in Subsection 'Vehicle Model', the state of the vehicle at time t is specified by $(\mathbf{x}^t, \dot{\mathbf{x}}^t, \theta^t)$, where $\mathbf{x}^t$, $\dot{\mathbf{x}}^t$, and θ^t denote the position, velocity, and orientation of the vehicle. The vehicle trajectory is fully specified by a sequence of vehicle states at all frames. Again, by the unit mass assumption, the acceleration of the vehicle at frame t is given by $\ddot{\mathbf{x}}^t = \mathbf{f}^t$. Thus, unclamped acceleration $\ddot{\mathbf{x}}^t$ at each frame is trivially obtained at every frame t.

Suppose that we are now at frame t. The inter-frame acceleration change is clamped and added to $\ddot{\mathbf{x}}^{t-1}$, using the corresponding bound that is estimated in 'Analysis' Section. That is,

$$\ddot{\mathbf{x}}^t \leftarrow \ddot{\mathbf{x}}^{t-1} + \text{clamp}(\|\ddot{\mathbf{x}}^t - \ddot{\mathbf{x}}^{t-1}\|) \cdot \frac{\ddot{\mathbf{x}}^t - \ddot{\mathbf{x}}^{t-1}}{\|\ddot{\mathbf{x}}^t - \ddot{\mathbf{x}}^{t-1}\|}$$

The tangential and lateral components of $\ddot{\mathbf{x}}^t$ are clamped using their respect bounds at frame t and the results are added to further refine $\ddot{\mathbf{x}}^t$:

$$\ddot{\mathbf{x}}^t \leftarrow \text{clamp}(a_x^t) \cdot \left((0,1,0)^{\mathrm{T}} \times \frac{\dot{\mathbf{x}}^{t-1}}{\|\dot{\mathbf{x}}^{t-1}\|}\right) + \text{clamp}(a_z^t) \cdot \frac{\dot{\mathbf{x}}^{t-1}}{\|\dot{\mathbf{x}}^{t-1}\|}$$

Here, the unknown forward-moving direction $\frac{\dot{\mathbf{x}}^t}{\|\dot{\mathbf{x}}^t\|}$ of the vehicle at time t is approximated to $\frac{\dot{\mathbf{x}}^{t-1}}{\|\dot{\mathbf{x}}^{t-1}\|}$, under the assumption that the moving direction of the vehicle is continuous and changes negligibly within a time step. From $\ddot{\mathbf{x}}^t$, we derive $\dot{\mathbf{x}}^t$ and clamp it:

$$\dot{\mathbf{x}}^t \leftarrow \dot{\mathbf{x}}^{t-1} + \ddot{\mathbf{x}}^t\Delta t,$$

$$\dot{\mathbf{x}}^t \leftarrow \text{clamp}(\|\dot{\mathbf{x}}^t\|) \cdot \dot{\mathbf{x}}^t / \|\dot{\mathbf{x}}^t\|$$

where Δt is the inter-frame time increment. The position of the vehicle at time t is given by

$$\mathbf{x}^t \leftarrow \mathbf{x}^{t-1} + \dot{\mathbf{x}}^t\Delta t,$$

and its orientation θ^t is computed using Equation (1).

Adding Details

Provided with vehicle state $(\mathbf{x}^t, \dot{\mathbf{x}}^t, \theta^t)$, we describe how to add motion details to vehicle position $\mathbf{x}^t$ and orientation θ^t in an on-line manner to convert them to a human pelvis configuration, $(\mathbf{p}_0^t, \mathbf{q}_0^t)$, where $\mathbf{p}_0^t$ and $\mathbf{q}_0^t$ denote the pelvis position and orientation, respectively. In order to do this, a motion prescription is made by combining a user-provided motion type with vehicle speed $\|\dot{\mathbf{x}}^1\|$ and vehicle acceleration $\ddot{\mathbf{x}}^1$ at the first frame (frame 1).

Given the motion prescription, a sequence of local joint configurations are generated at the first frame by blending example motion segments, using the on-line locomotion synthesis system.[12] Therefore, the number of frames for the prescribed motion is fixed at the first frame, say k. The motion details for the corresponding local poses are also blended with the system. With this information obtained, the task of adding details is performed frame by frame.

Let $(d\mathbf{x}_e^t, d\mathbf{q}_{\text{vert}}^t, \mathbf{q}_{\text{offset}}^t)$, be the motion details at frame $t, 1 \le t \le k$. Then, the pelvis orientation at frame t is derived by applying $d\mathbf{q}_{\text{vert}}^t$ and $\mathbf{q}_{\text{offset}}^t$ to vehicle orientation θ^t:

$$\mathbf{q}_0^t = d\mathbf{q}_{\text{vert}}^t \cdot \theta^t \cdot \mathbf{q}_{\text{offset}}^t$$

In order to compute pelvis position $\mathbf{p}_0^t$, let

$$(w, \mathbf{v}) = \theta^t(0, d\mathbf{x}_e^t)^{\mathrm{T}}(\theta^t)^{-1},$$

where $\mathbf{v}$ is a 3D vector. Then,

$$\mathbf{p}_0^t = \mathbf{x}^t + \mathbf{v}$$

Together with local joint configuration $(\mathbf{q}_1, \ldots, \mathbf{q}_m)$, the refined pelvis configuration $(\mathbf{p}_0^t, \mathbf{q}_0^t)$ fully specifies human pose, $(\mathbf{p}_0^t, \mathbf{q}_0^t, \mathbf{q}_1^t, \ldots, \mathbf{q}_m^t)$ at frame t.

Results

In order to visually validate our scheme, we perform trajectory refinement and synthesize locomotive

DOI: 10.1002/cav

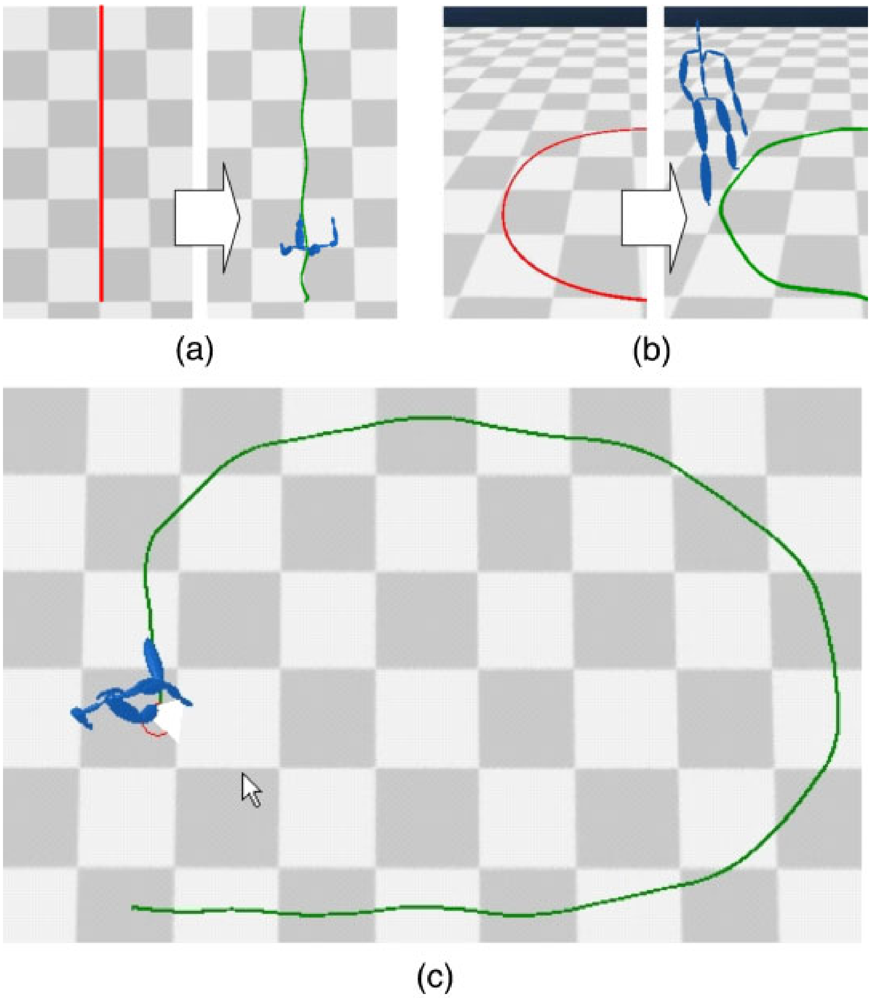


Figure 5. Trajectory refinement. (a) Straight walking; (b) Curved walking; (c) Human trajectory.

motions with these trajectories, employing the framework for on-line locomotion synthesis in Reference [12].

In the first experiments, we show how final trajectories are different from their initial versions. As shown in Figure 5(a) and (b), two parametric curves (including a straight line) are sampled to produce the corresponding point streams to be used as the input. Both the input and output curves are visualized side by side.

The next experiments are performed for data streams sampled with two types of input devices, an analog joystick and a mouse. For the mouse, time-varying cursor positions are sampled in an on-line manner to derive force profiles. For the joystick, force profiles are obtained from stick positions and directions. Motion types are specified by a keyboard or buttons.

Figure 5(c) shows a refined trajectory after clamping outliers and adding details learned from example motion data. The motions that are synthesized with final trajectories are shown in the accompanying video. Our scheme facilitates on-line locomotion control without latency. Equipped with the scheme, the locomotion synthesis system can produce more than 500 frames per second.

Discussion

Our scheme samples the intended (unknown) trajectory point by point interactively. Since the user can easily

DOI: 10.1002/cav

adapt to the clamping to form a human-in-the-loop feedback process, the intended trajectory can be achieved. As shown in the first experiment, an input trajectory could be given as a curve. In this case, the curve is converted to a point stream by properly sampling the curve. However, the curve may not preserve its global shape since its speed and acceleration are clamped.

We have parameterized the trajectory of an example motion segment with the horizontal speed of the vehicle and its horizontal acceleration vector at the initial frame. The rationale for this choice is that the motion segment is so short that the information at the initial frame characterizes the whole motion segment, which implies that the motion segment itself is parameterized in the same way. This guarantees latency-free trajectory construction. The trajectory could be further improved by incorporating a look-ahead capability while sacrificing responsiveness.

Conclusions

In this paper, we present an on-line data-driven scheme for effectively prescribing a human pelvis trajectory. Our scheme analyzes example motion data to extract the information on human steering behavior including motion details together with bounds on inter-frame acceleration variations, acceleration bounds, and speed bounds. Given a stream of point samples in an on-line manner, our scheme first transforms them one by one to a smooth vehicle trajectory by clamping outliers, exploiting the bounds that have been estimated from the example motion data. Our scheme then adds motion details to the vehicle trajectory to obtain a human pelvis trajectory to follow. For explanation purpose, we concentrate on the 'pelvis' trajectory using a 'mouse' as input device. However, we can easily generalize our scheme for other trajectories such as a center of mass (COM) trajectory and a zero moment point (ZMP) trajectory. We can also use an input devise other than a mouse, for example, a joystick or even slide bars.

ACKNOWLEDGMENTS

This work was supported by the Korea Science and Engineering Foundation (R01-2004-000-10413-0) and the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-311-D00190).

References

1. Reynolds C. Steering behaviors for autonomous characters, 1999.
2. Go J, Vu T, Kuffner JJ. Autonomous behaviors for interactive vehicle animations. In *SCA'04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 9–18, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.
3. Helbing D, Farkas IJ, Vicsek T. Simulating dynamic features of escape panic. *Nature* 2000; **407**: 487–490.
4. Metoyer RA, Hodgins JK. Reactive pedestrian path following from examples. *The Visual Computer* 2004; **20**(10): 635–649.
5. Treuille A, Cooper S, Popovic Z. Continuum crowds. *ACM Transactions on Graphics* 2006; **25**(3): 1160–1168.
6. Guo S, Robergé J. A high-level control mechanism for human locomotion based on parametric frame space interpolation. In *Proceedings of Eurographics Workshop on Computer Animation and Simulation 96*, pp. 95–107, August 1996.
7. Wiley DJ, Hahn JK. Interpolation synthesis for articulated fiture motion. *IEEE Computer Graphis and Applications* 1997; **17**(6): 39–45.
8. Rose C, Cohen MF, Bodenheimer B. Verbs and adverbs: mulidimensional motion interpolation. *IEEE Computer Graphics and Applications* 1998; **18**(5): 32–40.
9. Park SI, Shin HJ, Shin SY. On-line locomotion generation based on motion blending. In *Proceedings of ACM SIGGRAPH Symposium on Computer Animation*, pp. 105–111, July 2002.
10. Park SI, Shin HJ, Shin SY. On-line motion blending for real-time locomotion generation.*Computer Animation and Virtual Worlds* 2004; **15**: 125–138.
11. Mukai T, Kuriyama S. Geostatistical motion interpolation. *ACM Transactions on Graphics* 2005; **24**(3): 1062–1070.
12. Kwon T, Shin SY. Motion modeling for on-line locomotion synthesis. In *SCA'05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM Press: New York, NY, USA, , 2005; 29–38.
13. Correia L, Steiger-Garo A. A useful autonomous vehicle with a hierarchical behavior control. *European Conference on Artificial Life*, pp. 625–639, 1995.
14. Bartels RH, Beatty JC, Barsky BA. *An Introduction to Splines for Use in Computer Graphics & Geometric Modeling*. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1987.
15. Kim TH, Park SI, Shin SY. Rhythmic-motion synthesis based on motion-beat analysis. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)* 2003; **22**(3): 392–401.

DOI: 10.1002/cav

Authors' biographies:

Taesoo Kwon received his BS degree in Computer Science from Seoul National University, Korea, in 2000 and his MS degree in Computer Engeneering from the same university in 2002. He is currently a PhD student in the Department of Electrical Engineering and Computer Science at the Korea Advanced Institute of Science and Technology(KAIST), Korea. His primary research interests are computer graphics, data-driven character animation, and machine learning.

Sung Yong Shin received his BS degree in Industrial Engineering in 1970 from Hanyang University, Seoul, Korea, and his MS and PhD degrees in Industrial Engineering from the University of Michigan in 1983 and 1986, respectively. He is a Professor in the Department of Electrical Engineering and Computer Science at the Korea Advanced Institute of Science and Technology, Taejon, Korea. He has been working at KAIST since 1987. At KAIST, he teaches computer graphics and computational geometry. He also leads a computer graphics research group that has been nominated as a national research laboratory by the Government of Korea. His recent research interests include data-driven computer animation and geometric algorithms.

Comp. Anim. Virtual Worlds 2007; **18**: 463–472
DOI: 10.1002/cav