

Automating Expressive Locomotion Generation

Yejin Kim and Michael Neff

University of California, Davis,
Department of Computer Science and Program for Technocultural Studies,
1 Shields Avenue, Davis, CA 95616, U.S.A.
rokkim@ucdavis.edu, neff@cs.ucdavis.edu
<http://idav.ucdavis.edu>

Abstract. This paper introduces a system for expressive locomotion generation that takes as input a set of sample locomotion clips and a motion path. Significantly, the system only requires a single sample of straight-path locomotion for each style modeled and can produce output locomotion for an arbitrary path with arbitrary motion transition points. For efficient locomotion generation, we represent each sample with a loop sequence which encapsulates its key style and utilize these sequences throughout the synthesis process. Several techniques are applied to automate the synthesis: foot-plant detection from unlabeled samples, estimation of an adaptive blending length for a natural style change, and a post-processing step for enhancing the physical realism of the output animation. Compared to previous approaches, the system requires significantly less data and manual labor, while supporting a large range of styles.

Keywords: character animation, locomotion style, motion transition, motion path, motion capture data.

1 Introduction

The popularity of motion capture technology makes it possible to convert the physical realism of live movements into manageable data. Taking as input a set of recorded motion clips, previous approaches such as *motion blending* and *motion concatenation* have been developed for locomotion synthesis. They all have inherent trade-offs: A large number of sample motions are required to synthesize an output locomotion following an arbitrary motion path. Furthermore, most of them uses a basic locomotion type such as walking or running as their sample set, where a single blending length has been used throughout the synthesis process. With stylistic samples, this can introduce undesirable lag or suddenness during a motion transition [20].

In this paper, we introduce an animation system that generates expressive locomotion from a set of sample clips and a motion path specified by an animator. By using short, straight-path locomotion clips as our sample set, we require only a small amount of data. We focus on automating locomotion synthesis so that

a virtual character can adapt to the input path without an animator’s further intervention. Since the quantitative aspects of various styles such as body speed, number of foot steps, and range of end-effector positions differ, we represent each sample by constructing a *loop sequence* which consists of a minimum number of foot-plants required to capture the key style and loops smoothly with itself. We show how to construct such a loop sequence from an unlabeled sample with minimum manual effort. During the locomotion synthesis, an adaptive blending length is estimated for transition between styles. After the synthesis, we enhance the physical validity of the output animation by enforcing foot constraints and postural adjustments.

Our system contributes an automated method of generating expressive locomotion from unlabeled sample clips. We provide a simple and efficient method of constructing a loop sequence from detected foot-plants that is then used in the synthesis process. An adaptive blend duration is automatically determined to provide natural style changes. Notably, our system only requires a small number of sample clips for a wide range of output. Further, the physical realism is enhanced, especially for turning motions on a curved path. With our system, an animator can quickly generate an output animation through control of a motion path alone, all at interactive speed.

2 Related Work

Motion concatenation is widely used to generate variants of sample motion without destroying the physical details of the original motion. For concatenation of motion segments, a *motion graph* [1,7,9] is typically constructed to represent an output motion sequence as a valid graph path, satisfying edge and node constraints specified by an animator. Coupled with motion blending techniques, this type of approach can provide online locomotion generation [8,12,14], where control parameters such as speed, turning angles, or emotional values can be continuously fed into the system. Kovar and Gleicher [6] investigated the range of motions that can be created automatically using blending. A low-dimensional control model was suggested by Treuille *et al.* [18] to generate continuous locomotion sequences from a reduced number of samples. Oshita [11] proposed an automated system for generating a continuous motion sequences from a set of short motion clips based on the different support phases of the foot between two motions. Recently, Zhao *et al.* [22] introduced an algorithm to reduce the size of motion graph by limiting the transition time between sample motions. Nevertheless, the output quality of graph traversal or multidimensional blending methods is inherently related to the density of input sample motions. Thus, all these approaches require additional sample data, such as various turning motions, for each style modeled, for an arbitrary motion path. Furthermore, they rely on a fixed and user-defined blending length for motion transitions, which is not suitable for generating a natural transition between stylistically different

motions. Our system shares a similar goal in terms of generating high-quality locomotion sequences from samples; however, our approach only requires one sample for each style and uses an adaptive blending length for a convincing motion transition.

Some approaches [2,3,10,16,19,21] put more emphasis on editing the locomotion’s style by controlling kinematic constraints. Based on the Fourier series approximation of the joint angles, Unuma *et al.* [19] interpolated or extrapolated walking and running with different emotions. Given a set of edited key poses, the displacement mapping technique [2,21] modifies the overall shape of a motion while maintaining the local details; however, its output quality is highly reliant on manual specification of key poses. Gleicher [4] applied a similar mapping technique to edit a sample motion following a different arc-length parameterized path. Although they preserved foot constraints relative to a path, transformed turning motions from a straight-path can be unrealistic due to the physical invalidity. Provided with an arbitrary motion path, Sun and Metaxas [16] presented a gait generation system, where low-level joint rotations are adjusted from the high-level constraint parameters such as step length and height. Glardon *et al.* [3] provided adaptive foot-plant detection and enforcement on noisy data. Recent work has explored style adjustment through interactive control of a set of pose parameters, correlated between different body parts [10]. None of these approaches target expressive locomotion generation from stylistically different samples and an arbitrary motion path directly specified by an animator in a fully automated way.

Maintaining physical realism during the synthesizing process is another key issue for generating a high-quality output animation. Given a kinematically edited motion, several motion filtering techniques [15,17] have employed the ZMP-based balance adjustment which corrects physically implausible postures into balanced ones. Inspired by this, we apply a similar adjustment to create realistic turning motions to match path curvature.

3 Overview

Figure 1 shows an overview of our locomotion generation. For input data, we use a set of unlabeled locomotion clips, capturing from a wide variety of locomotion styles, all on a straight-path, as shown in the accompanying video.

Our locomotion generation starts with constructing a motion path from a series of path points (Section 4). To represent the key style of each sample, a corresponding loop sequence is constructed from detected foot-plants (Section 5) and concatenated repeatedly or blended with another loop sequence until the target path is filled with the sample motions (Section 6). Finally, the physical realism of the initial output animation is enhanced by foot constraint enforcement to remove any foot-skate as well as postural adjustments to generate convincing turning motions on a curved path (Section 7).

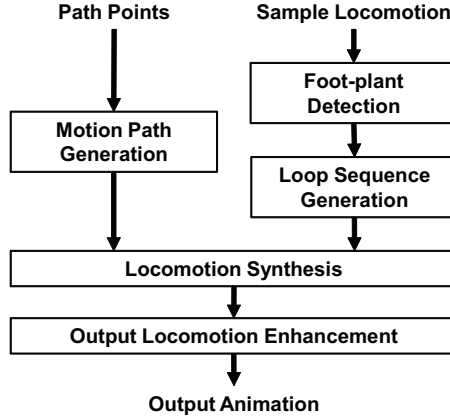


Fig. 1. Overview of the expressive locomotion generation

4 Locomotion Path Generation

In our system, a motion path is defined with a time-varying curve, which specifies the root position and orientation of the character at a given time, t . For intuitive control, it is approximated directly from a series of input path points, sampled from an animator’s input device such as a mouse pointer. Thus, given a set of N continuous sample points, each point, p_i , is appended continuously to form a rough path, $\mathbf{p}(t)$. This path undergoes a smoothing operation by applying a 1D Gaussian filter kernel of width one to the last N_p neighborhood points in order to generate \hat{p}_i , a weighted average of neighboring points. For the next input point, p_{i+1} , we replace p_i with \hat{p}_i and repeat the operation until all of the remaining points are filtered to form $\hat{\mathbf{p}}(t)$. Here, we set $N_p = 7$, which was sufficient to prevent sudden orientation changes of a character’s root following the given path. During locomotion synthesis, $\hat{\mathbf{p}}(t)$ is parameterized by arc length in order to vary the body speed based on the path condition.

5 Loop Sequence Generation

Our sample set includes a wide variety of styles. Since the quantitative aspects of each style differ in speed, foot steps, and the range of body movements, we represent each sample with a processed clip we term a *loop sequence* that is specified from multiple foot-plants detected from unlabeled samples.

5.1 Foot-Plant Detection

A foot-plant is an important physical constraint that must be precisely detected with two boundary points. Manual labeling is a laborious task even for a short locomotion sequence; furthermore, boundary points are difficult to detect precisely

with a fixed threshold value as the sample motion contains noise and retargeting errors [3]. For this reason, we initially approximate foot-plants and provide a new ankle trajectory that replaces step transition periods such as heel-strike and toe-off moments (Section 7).

Assuming each sample has been captured with a reasonable noise level, a low-pass filter like the Savitzky-Golay filter [13] can be used to smooth out local peaks of noise from motion curves of ankle joints while maintaining the high peaks which possibly belong to an unconstrained frame. For our sample clips, 12 frames (0.1 frames per second) for the filtering window size with degree 4 of the smoothing polynomial preserve the global shape of significant peaks in target data.

Once ankle joint data are smoothed out, boundary points of each foot-plant can be roughly selected from two geometric thresholds: the height and translational speed of foot. We derived the first threshold at 10 ~ 20% of the vertical range of the extreme high and low ankle positions and the second threshold from the average body speed of the sample locomotion sequence. Due to the lenient smoothing applied by the filter, noisy frames can still exist within a detected foot-plant. Outliers like these tend to last only a few frames; thus, we can include them as a part of a foot-plant by checking their neighboring frames. For each noisy frame, we look at N_f frames on both sides of its position and add $1/(2N_f + 1)$ to the probability of it being a foot-plant frame whenever its neighbor is a foot-plant frame. We include a noisy frame into a foot-plant if its probability is over 0.5. Here, we set N_f with 1/4 of the length of foot-plant.

5.2 Sequential Cycle Construction

Once each sample clip is labeled with foot-plants, the start and end of a loop sequence can be specified as any frame within a foot-plant. We select the middle frame since it is typically a moment during which the foot stands fully on the ground. For the sequence alignment during a motion transition, each loop sequence starts and ends within a foot-plant of the left foot, requiring at least three foot-plants detected from the sample clip as shown in Figure 2.

Loop sequences are constructed so that they can be concatenated repeatedly and will yield smooth motion without further blending. Formally, this requires

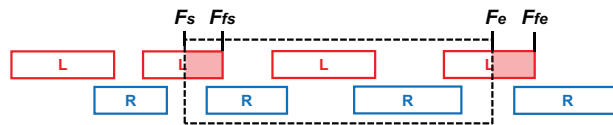


Fig. 2. Specification of a loop sequence: The shaded area is interpolated to create the smooth loop continuity

continuity between the starting frame, F_s , and the ending frame, F_e , of the loop sequence. This is ensured by pre-blending the ends of the loop sequence in order to ensure this continuity; however, the size of blending window, N_b , differs for each loop sequence constructed from samples as follows:

$$N_b = \min(F_{f_s} - F_s, F_{f_e} - F_e),$$

where F_{f_s} and F_{f_e} are the last frame of the starting foot-plant and of the ending foot-plant respectively. Thus, after blending N_b frames between $F_s + i$ and $F_e + 1 + i$, we append the first $N_b/2$ blended frames to F_e and then replace N_b frames from F_s with the remaining $N_b/2$ blended frames in order to guarantee smooth continuity throughout the sequence. For the i th blending frame, $0 \leq i < N_b$, we linearly interpolate the root position, R_i , and perform spherical linear interpolation on j th joint rotation, q_i^j , as follows:

$$R_i = w(i)R_{a,i} + (1 - w(i))R_{b,i}, \quad (1)$$

$$q_i^j = \text{slerp}(q_{R_{a,i}}^j, q_{R_{b,i}}^j, w(i)), \quad (2)$$

where $w(i)$ is a sinusoidal function for blending weights, which should monotonically increase with a range of $[0, 1]$. To ensure the smooth continuity of a loop sequence, we used $w(i) = 1 - \frac{1}{2}\cos(\frac{i+0.5}{N_b}\pi) + \frac{1}{2}$. Here, $R_{a,i}$ and $R_{b,i}$ are the i th frame after $F_e + 1$ and after F_s respectively. Since we complete the construction process by replacing the first N_b frames of a loop sequence with second $N_b/2$ frames of the blended frames, the y position of $F_s + i$ is used for $R_{a,i}$ if $i \geq N_b/2$ and of $F_e + 1 + i$ is used for $R_{b,i}$ if $i < N_b/2$. This way, we maintain the original vertical movements of the root for the output synthesis, adjusting to a new root position on an input path.

6 Locomotion Synthesis

An initial output locomotion is generated by concatenating a loop sequence repeatedly or blending with other sequences to transition between styles. Provided with the transition timings specified by an animator as shown in the Figure 3, all loop sequences are fit to an arc-length parameterized motion path.

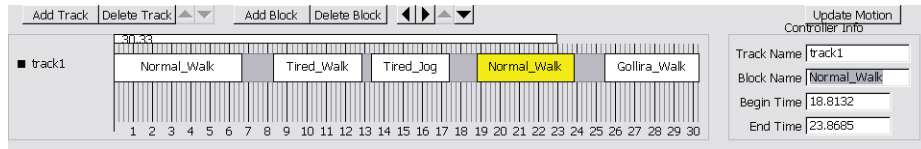


Fig. 3. Temporal placements of loop sequences on a motion path via the timing editor

6.1 Loop Sequence Alignment

Each loop sequence constructed from a sample set differs in the number of foot-plants and the body speed. In order to make a natural transition from one sequence to another, two sequences should be blended with correspondence in time based on their key-times. These key-times can be easily set from the two boundary points of each foot-plant detected in Section 5.1. For loop sequence alignment, we utilized the dynamic time-warping technique suggested by Rose *et al.* [14]. With their method, the normalized key-times are used to map the actual key-times T of each sequence, defined as $\{K_1, \dots, K_{N_k}\}$, to a generic time $t \in [0, 1]$. Here, $N_k = 2N_{fp}$ and N_{fp} is the number of foot-plants. However, this linear mapping must maintain a consistent number of keys, N_k , for all sample sequences, but this often varies depending on the sampled style. To deal with this problem, we determine the correspondence of each loop sequence to a shorter clip and use this to calculate the correspondence in time of the full loop sequences. For example, if there is a transition between two sequences, S_a and S_b , as shown in Figure 4, we use the shorter clip, S_w , as an intermediary clip such that each cycle (the black vertical lines in Figure 4) of S_a and S_b is aligned with S_w to make the correspondence in key-times from S_a to S_b .

Therefore, for a given t , its corresponding frame F can be calculated for each sequence as follows,

$$F(t) = Fi + \frac{F_{i+1} - F_i}{K_{i+1} - K_i}(t - T_i),$$

where F_i is a key-frame corresponding to K_i , $1 \leq i < N_k$.

6.2 Adaptive Blending Length

Unlike graph-based approaches [1,7,8,9,22], our system allows an animator to start a transition at any point within the loop sequence and uses an adaptive blending length to ensure a natural style change during a motion transition. Since one stylistic locomotion can be distinguished from others with a small number of point configurations [5], we used the end-effectors of the body as such differentiating points: the head, two hands, and two foot positions.

If a motion transition is requested between two poses, P_a within sequence S_a and P_b within sequence S_b , we first find the end-effector that has the largest positional difference between P_a and P_b . Let this value be ΔD_i for i th end-effector. Given the frame rate for capturing sample locomotion, fr (frames per second), we can estimate the blending length, ℓ_b , as follows:

$$\ell_b = \left(\frac{\Delta D_i}{|T_{i,a} - T_{i,b}|} \right) \left(\frac{1}{fr} \right),$$

where $T_{i,a}$ and $T_{i,b}$ is the average distance traveled by i th end-effector in S_a and S_b respectively. This ℓ_b ensures a gradual change of all end-effectors without disturbing the physical coherence of joints.

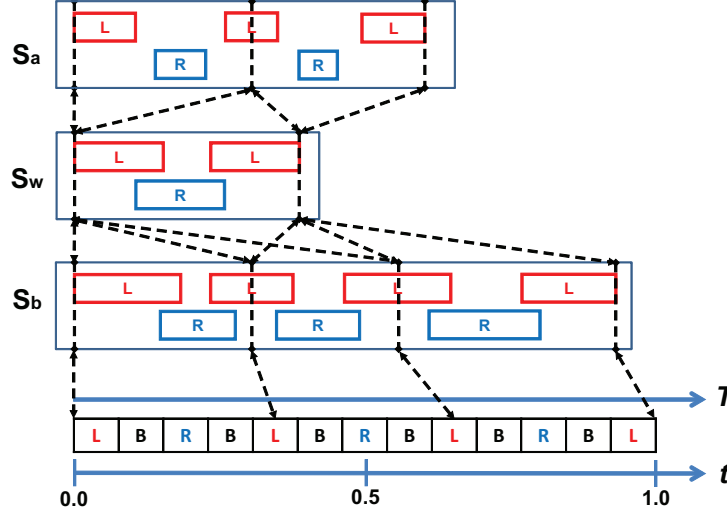


Fig. 4. Loop sequence alignment: L and R represents a left and a right foot on the ground while B represents either both feet on ground or in flight. The dashed lines represent the key-time correspondences of each cycle.

6.3 Displacement Mapping

Once all loop sequences are temporarily specified on an arc-length parameterized motion path, $\hat{\mathbf{p}}(t)$, we now determine the position and orientation of the root for each output frame. To incorporate the varied speed between the sequences into the output generation, the *speed* parameter s_i in the i th sequence is estimated as ℓ_i/N_i , where ℓ_i is an overall length of the reference root line, a linear least squares fit of the root trajectory, and N_i is the total number of frames in a loop sequence. Based on s_i , the current arc length on $\hat{\mathbf{p}}(t)$ determines the root position, $r_m(t)$, and orientation, $q_m(t)$, by interpolating between two sample points on $\hat{\mathbf{p}}(t)$.

For each frame in the sample sequence, we now map the local displacement of the root from its base motion path to the arbitrary output motion path in order to preserve the continuity and global shape of root trajectory from the original sample. Given $r_m(t)$ and $q_m(t)$ from the parametric path at a generic time t , the output root is derived as follows,

$$r_o(t) = r_m(t) + (r_i(t) - \hat{r}_i(t)),$$

$$q_o(t) = q_m(t)\hat{q}_i^{-1}(t)q_i(t),$$

where $r_i(t)$ and $q_i(t)$ are the projected root position and orientation respectively. $\hat{q}_i(t)$ is the orientation aligned to the tangential direction at $\hat{r}_i(t)$ on the reference root trajectory. Notice that the addition of the difference between $r_i(t)$ and $\hat{r}_i(t)$ reflects the side-to-side weight shift of the sample locomotion into the output animation.

7 Output Locomotion Enhancement

An output animation generated so far contains a character following a specified motion path, but lacking two key elements of physical validity. First, foot-skating is significant throughout the output animation, especially during foot-plant transitions. Second, the character does not make any physical reaction to the path curvature. In this section, we enforce such physical validity without using an expensive physics-based simulation.

7.1 Foot Constraint Enforcement

With the foot-plants detected in Section 5.1, we enforce the foot constraints by adopting the low-level IK routine from [10] due to its online performance and simplicity of implementation. However, rooting constrained frames at a single position introduces a motion discontinuity between an unconstrained frame, F_u , and a constrained frame, F_c , after and before each foot-plant. We eliminate such discontinuities by interpolating the frames to produce a new ankle trajectory. As the rate of ankle movements follows an ease-in and out curve between F_u and F_c , we apply the same interpolation function (Equation 1 and 2) with a slightly different weight function, $w(i) = 1 - \cos(0.5 \frac{i+0.5}{N_b} \pi)$, $0 \leq i < N_b$. Here, N_b is the number of interpolated frames between F_u and F_c and determined based on the Euclidean distance between F_u and F_c divided by the average distance traveled by an ankle joint during each foot-plant. We ensure the smoothness of a new joint trajectory by connecting between F_u and F_c with a Hermite spline. Here, the two tangents at F_u and F_c are determined as $\hat{T}_{F_u} = 0.5 \frac{|F_u - F_c|}{|T|} T$ and $\hat{T}_{F_c} = 0.1 \frac{|F_u - F_c|}{|T|} T_y$, where T is a tangent obtained from F_u to its closest neighbor frame, and T_y is the y -component of T . The constrained positions of ankle over a new trajectory is shown in Figure 5. We apply a similar constraint enforcement for the leg swivel rotation in order to prevent an undesirable spinning motion of the rooted foot on the ground. For this, an average swivel angle of a foot-plant is used to orient the rooted foot's direction and gradually changes its rotation angle over the new ankle trajectory to prevent any rotation discontinuity.

7.2 Postural Adjustments

Each loop sequence constructed from a straight-path sample set contains a uniform body speed and foot step size. Because of this, an output animation generated from the sequence will lack any physical reaction to the path curvature such as reduced body speed and stride length. Additionally, a character should lean toward the instantaneous center of curvature of the path in order to resist the centripetal force.

For reduced speed and stride length, a set of scaled versions of each loop sequence is generated with progressively shorter strides. This is done by scaling the distance between the foot-plants and the vertical range of the foot and then reconstructing motion comparable to the original using IK techniques based on

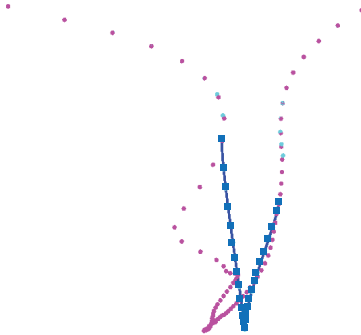


Fig. 5. Foot Constraint Enforcement: Original ankle positions (round dots) are enforced to the constrained positions (rectangle dots) on a Hermite spline

[10]. We derive the reduced speed parameter for each scaled loop sequence as described in Section 6.3. The final locomotion is generated by sampling across this family of scaled loop sequences to continuously adjust stride length. Each output frame is selected from one of the scaled sequences based on current curvature of the path.

To automate this adjustment process, the difference of tangential direction between \hat{p}_i and \hat{p}_{i+1} on the arc-length parameterized path is precalculated to determine the degrees of curvature throughout the path. The largest degree is used to determine the number of scaled sequences that need to be constructed in advance. Based on our experiments, we scaled down 1% of the original step size per 0.2 degree difference, to a maximum of 50%, and found no significant motion discontinuity in the output animation. We limit the scaling to 50% since further reduction can impact the physical coherence between foot and torso joints.

The ZMP provides a useful correction guide for maintaining physically plausible postures. Since we only need to adjust ankle joint rotation to mimic body lean during a turning motion, the ZMP-based balance adjustment suggested by [15] is utilized to calculate the rotation value for the ankle joint. For the actual ZMP calculation, we used a 4th order central difference with 0.5 seconds as the sampling rate for the acceleration term.

8 Experimental Results

Our experiments were performed on an Intel Core 2 QuadTM 2.4GHz PC with 2GB memory. As a sample set, we used various short locomotion clips as shown in Table 1, where their style differs from each other in a frame duration and number of foot steps. All input clips are sampled at the rate of 120 frames per second and share the same skeletal structure. The skeletal model is defined in a Y-up coordinate frame and includes 6 DOFs for root position and orientation and 42 DOFs for body joint orientations: 12 for the torso and head, 9 for each arm, and 6 for each leg.

Table 1. Number of frames and foot steps for sample locomotion clips and corresponding loop sequences

	Walking	Jogging	Excited	Balancing	Distressed	Energetic	Tired
Sample Frames	296	253	530	1283	558	341	801
Sample Steps	5	6	8	7	7	7	9
Loop Sequence Frames	143	100	365	922	364	226	387
Loop Sequence Steps	2	2	4	6	4	2	4

Our experiments start with constructing loop sequences from various stylistic samples and comparing them with shorter two-step sequences. For the comparison, two versions of a loop sequence are constructed with two steps and multiple steps respectively as shown in the Table 1. The accompanying video shows that the sequences with multiple steps generate physically (balancing) and stylistically (excited) more convincing results than the two step cycle does.

Next, we compare motion transitions using adaptive blending lengths to ones with fixed duration. In this demonstration, two comparisons were made with various samples while the transition point was chosen randomly. For a fixed blending length, we used 0.33s (40 frames) since this value has been widely used in previous approaches [20]. The adaptive blending length takes about 2.22 ~ 3.89s for the transition between the walking and balancing sample and about 1.32 ~ 1.78s for the transition between energetic and tired sample. In both cases, the blending length varies as the transition can start at any point within each loop sequence. The accompanying video shows that the adaptive blending length achieves gradual motion changes; thus, it generates physically more plausible poses than the fixed blending length does.

To demonstrate the effects of our locomotion enhancement for realistic turning motions, we next compared the constant turning motion with one enhanced by our postural adjustments. In this demonstration, we used walking and jogging samples on a sinusoidal motion path as body speed affects the reaction to centripetal force. For the scaled adjustment, it took less than a minute to generate 50 scaled versions from the samples. The accompanying video shows that applied adjustments generate physically preferable turning motion for both scaled and ZMP-adjusted cases.

Our last result demonstrates an expressive animation generated from a set of stylistic samples and an arbitrary motion path. After the motion path is interactively specified, no user intervention has taken place during the output locomotion synthesis. We used a number of different stylistic samples for output animation as shown in the Figure 6, also in the accompanying video. In this demonstration, it took about 6.8s to generate 2401 frames for the first output and about 9.2s to generate 2951 frames for the second output, producing over 300 frames per second excluding the rendering time.

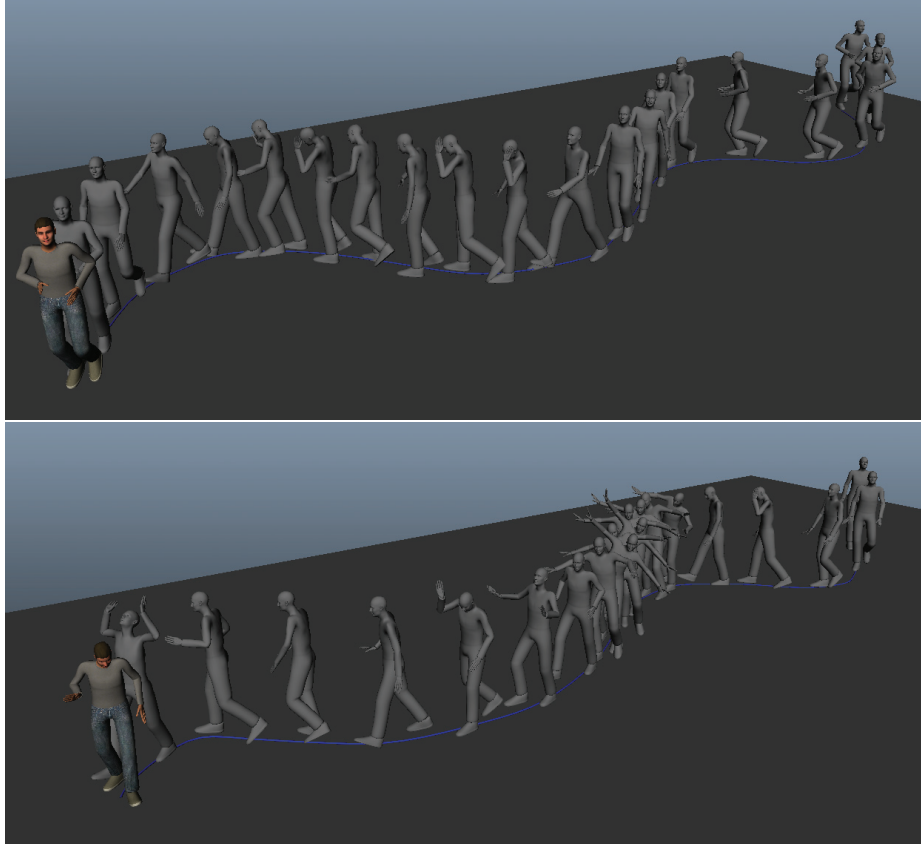


Fig. 6. Output Animations: First animation is generated from jogging, walking, and distressed samples. Second animation is generated from walking, distressed, balancing, and excited samples.

9 Conclusions

In this paper, we introduced an animation system for expressive locomotion that generates a continuous output sequence from a set of sample clips and an arbitrary motion path. Conscious of the high cost of preparing motion captured data, our system only requires a single sample of straight-path locomotion for each style modeled, keeping the size of the sample set to a minimum. The system focuses on automating the generation of expressive animation from a wide variety of stylistic locomotion samples. It does this by introducing a concise representation of each style, a loop sequence, which encapsulates its key style in multiple foot steps. Automation is further supported through: foot-plant detection from unlabeled samples, estimation of an adaptive blending length for a natural style change, and post-processing to enhance physical realism for turning motions on

a curved path. Our experimental results show that our system is capable of generating output animation at an interactive speed, making our system suitable for both automated and user-controlled virtual characters in various interactive applications.

Currently, our prototype system only supports samples containing cyclic foot steps. To extend the sample set to less cyclic motions like dance and ballet, more sophisticated foot-plant detection is required to anticipate the foot pattern with additional contact points such as toe and heel.

In addition, the system generates a set of scaled versions of loop sequences in order to enhance the turning motions. These redundant samples are linearly generated and controlled for an input path curvature; thus, the result can be physically implausible for certain outlier path conditions, such as radical turns. Directly editing of the kinematic controls of a character might complement our approach without using additional data.

Acknowledgments. Financial support for this research was provided in part by NSF grant 0845529 to University of California, Davis and through a software donation from Autodesk. Thanks to Jonathan Graham for model support and Hilary Bryan for motion capture support.

References

1. Arikan, O., Forsyth, D.A.: Interactive motion generation from examples. *ACM Transactions on Graphics* 21(3), 483–490 (2002)
2. Bruderlin, A., Williams, L.: Motion signal processing. In: *Proceedings of ACM SIGGRAPH 1995*, pp. 97–104 (1995)
3. Glardon, P., Boulic, R., Thalmann, D.: Robust on-line adaptive footplant detection and enforcement for locomotion. *Visual Comput.* 22(3), 194–209 (2006)
4. Gleicher, M.: Motion path editing. In: *Proceedings of ACM Symposium on Interactive 3D Graphics*, pp. 195–202 (2001)
5. Johansson, G.: Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics* 14(38), 201–211 (1973)
6. Kovar, L., Gleicher, M.: Flexible automatic motion blending with registration curves. In: *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 214–224 (2003)
7. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. *ACM Transactions on Graphics* 21(3), 473–482 (2002)
8. Kwon, T., Shin, S.Y.: Motion modeling for on-line locomotion synthesis. In: *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 29–38 (2005)
9. Lee, J., Chai, J., Reitsma, P.S.A., Hodgins, J.K., Pollard, N.S.: Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics* 21(3), 491–500 (2002)
10. Neff, M., Kim, Y.: Interactive editing of motion style using drives and correlations. In: *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 103–112 (2009)

11. Oshita, M.: Smart motion synthesis. *Computer Graphics Forum* 27(7), 1909–1918 (2008)
12. Park, S.I., Shin, H.J., Kim, T.H., Shin, S.Y.: On-line motion blending for real-time locomotion generation. *Computer Animation and Virtual Worlds* 15(3-4), 125–138 (2004)
13. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical Recipes in C++: The Art of Scientific Computing*, 2nd edn. Cambridge University Press, New York (2002)
14. Rose, C., Cohen, M.F., Bodenheimer, B.: Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications* 18(5), 32–40 (1998)
15. Shin, H.J., Kovar, L., Gleicher, M.: Physical touch-up of human motions. In: *Proceedings of Pacific Conference on Computer Graphics and Applications*, pp. 194–203 (2003)
16. Sun, H.C., Metaxas, D.N.: Automating gait generation. In: *Proceedings of ACM SIGGRAPH*, pp. 261–270 (2001)
17. Tak, S., Ko, H.: A physically-based motion retargeting filter. *ACM Transactions on Graphics* 24(1), 98–117 (2005)
18. Treuille, A., Lee, Y., Popović, Z.: Near-optimal character animation with continuous control. *ACM Transactions on Graphics* 26(3), article 7 (2007)
19. Unuma, M., Anjyo, K., Tekeuchi, T.: Fourier principles for emotion-based human figure animation. In: *Proceedings of ACM SIGGRAPH*, pp. 91–96 (1995)
20. Wang, J., Bodenheimer, B.: Computing the duration of motion transitions: An empirical approach. In: *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 335–344 (2004)
21. Witkin, A., Popović, Z.: Motion warping. In: *Proceedings of ACM SIGGRAPH*, pp. 105–108 (1995)
22. Zhao, L., Normoyle, A., Khanna, S., Safonova, A.: Automatic construction of a minimum size motion graph. In: *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 27–35 (2009)