Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2012)
P. Kry and J. Lee (Editors)

Component-based Locomotion Composition

Yejin Kim[†] and Michael Neff[‡]

Department of Computer Science and Program for Cinema and Technocultural Studies, University of California, Davis

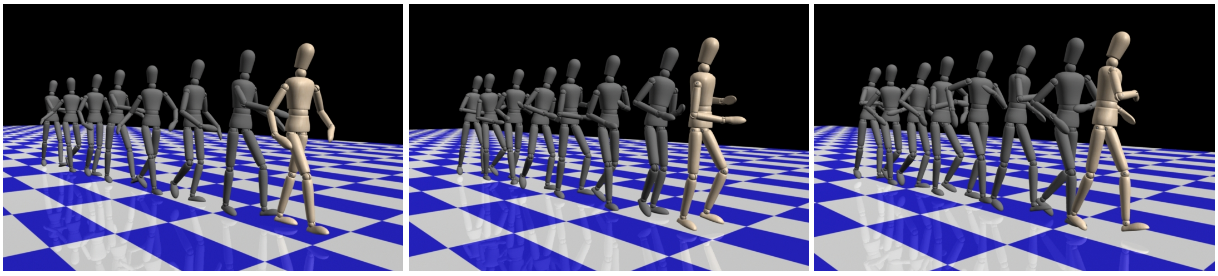


Figure 1: *Diverse locomotion outputs (in a happy style) synthesized by a linear mixture of combinatorial components decomposed from the 10 example set.*

Abstract

When generating locomotion, it is particularly challenging to adjust the motion's style. This paper introduces a component-based system for human locomotion composition that drives off a set of example locomotion clips. The distinctive style of each example is analyzed in the form of sub-motion components decomposed from separate body parts via independent component analysis (ICA). During the synthesis process, we use these components as combinatorial ingredients to generate new locomotion sequences that are stylistically different from the example set. Our system is designed for novice users who do not have much knowledge of important locomotion properties, such as the correlations throughout the body. Thus, the proposed system analyzes the examples in a unsupervised manner and synthesizes an output locomotion from a small number of control parameters. Our experimental results show that the system can generate physically plausible locomotion in a desired style at interactive speed.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Human locomotion is one of the most fundamental activities that a person performs on a daily basis, yet still a challenge to animate with an articulated character due to the correlated relationship across high degrees of freedom (DOFs). For this reason, the direct acqusition of locomotion through motion capture data becomes a natural choice, with example-based approaches that modify existing motion data into variants of the example set. Previously, great effort has been put into developing editing techniques based on statistical [CFP03, MH02, MH03, LPL08, SCF06, UGB*04] or multilinear analysis [LXPR11, Vas02, VT05], generative models [MLC10], or direct replacement of DOFs [AGH03, AW00, PB02, HKG06, IF04, Osh08]. However, most of these mainly focus on extracting and transferring style elements between two examples. Monolithic blending between pairs of examples produces a quite limited range of stylistic variation, especially when trying to maintain the physical constraints and correlations embedded in the original example data. This motivates us to explore a finer grained decomposition of motion, in which components that govern particular

[†] e-mail: rokkim@ucdavis.edu
[‡] e-mail: neff@cs.ucdavis.edu

DOI: 10.2312/SCA/SCA12/165-173

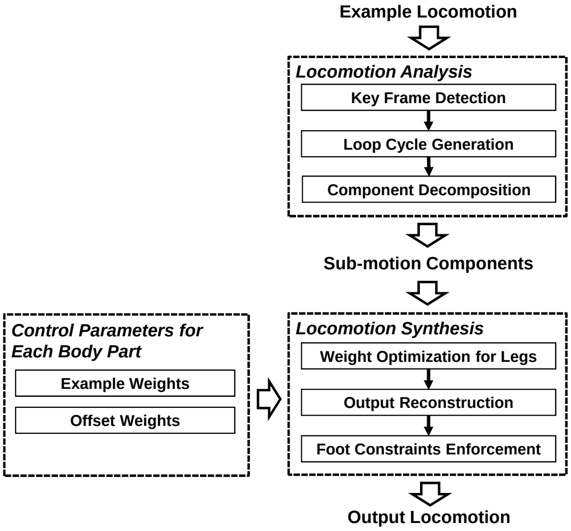


Figure 2: *Overview of the locomotion composition with sub-motion components*

aspects of movement for various parts of the body can be simultaneously combined from the example set, increasing the stylistic diversity of the output locomotion.

In this paper, we introduce a component-based system for human locomotion composition that takes a set of example locomotion clips as input and allows a user to synthesize new locomotion sequences by recombining a set of *sub-motion components* decomposed from separate body parts. As shown in Figure 2, our system consists of two processes: locomotion analysis and synthesis. In the locomotion analysis, each example sequence is first segmented into a loop cycle based on the detected key frames. Given a set of loop cycles, the system divides the body into several parts and applies *independent component analysis* (ICA) to each part in order to decompose the cycles into *independent components* (ICs) for each body section. Here, we call ICs *sub-motion components* since each component represents a partial motion in the body part and can be recombined with the corresponding components of other examples to define a new style. During the synthesis process, important physical constraints, such as foot contacts, are maintained by solving an optimization on the given weights for the leg components in order to generate a physically plausible output.

Our system makes two main contributions. First, we apply a linear statistical method to the locomotion composition process and represent a human locomotion as a linear mixture of sub-motion components. Grouping the related DOFs into separate body parts and decomposing them into editable components enables synthesis with rich varieties of style combinations from a small example set. Second, to generate physically plausible output, we developed an optimization method that weights individual components to satisfy the given physical constraints. Thus, the system requires minimal user intervention and provides automation for quick composition. Our experimental results show that a user can generate various stylistic locomotion at interactive speed, simply by specifying two control parameters, example and offset weights, for each input example in the body parts.

The remainder of this paper is organized into the following sections: We begin with a survey of previous approaches for editing and synthesizing human locomotion in Section 2. The locomotion analysis is explained in Section 3, and the synthesis part is detailed in Section 4. After experimental results are demonstrated in Section 5, we conclude this paper with a discussion of further investigations in Section 6.

2. Related Work

Locomotion synthesis from example-based approaches has been actively studied. An unsupervised statistical analysis such as *principal component analysis* (PCA) or ICA is widely adopted to represent a motion sequence as a linear mixture of decomposed components. Using PCA, Urtasun *et al.* [UGB*04] trained a locomotion model with a large number of examples by using external parameters such as speed and height and approximate the PCA coefficients to generate an output sequence with different speeds and heights. Mori and Hoshino [MH02, MH03] were the first to apply ICA to analyze motion capture data and to synthesize locomotion output by interpolating the corresponding ICs between two examples. Cao *et al.* [CFP03] used ICA to extract emotions and content from an example set of facial animation and combine them linearly to generate new facial animation while Shapiro *et al.* [SCF06] applied a similar approach to a spatial representation of human locomotion in order to transfer gross style from one example to another. Combining ICA with invariant feature subspaces, Liu *et al.* [LPL08] tried to perform style transfer more precisely in an automated way. In our system, we also utilize ICA to decompose the example set into editable components for the locomotion synthesis; however, our system focuses on combining ICs from more than two examples in order to expand the style diversity of outputs. Furthermore, our system can handle motion decomposition with an angular representation of the motion and style transfer between the examples with a large difference in body speed.

Multilinear analysis based on *tensor* decomposition is another technique to separate a locomotion sequence into multiple elements such as style and content. Vasilescu [Vas02] adopted the multilinear model to stylize the walking motions of a new character. Later, Vasilescu and Terzopoulos [VT05] developed this into the multilinear ICA (MICA) model to detect some of the image factors not distinguishable by conventional ICA during facial recognition. Recently, Liu *et al.* [LXPR11] applied MICA to human motion data and syntheiszed a new motion performed by unknown actors by reconstructing motion vectors from a low-dimensional parameter space. Min *et al.* [MLC10] used multilinear analysis to construct a generative model which augments the example set with speed variations and allows locomotion to be edited with motion parameters controlled by a user. It is useful to use multilinear models to decompose a human locomotion into a small of number of controllable parameters which can synthesize an output by editing the example set; however, the multi-linearity in this kind of statistical model requires a relatively large number of examples to be captured from different actions and multiple actors in order to form N vector spaces to be analyzed. In addition, it is mainly designed for style transfer from existing actors to unknown ones while our system aims for locomotion composition from multiple examples captured from any actor.

Some research focuses on expanding the size of a motion database by replacing specific body parts or DOFs of one example with another. Pullen and Bregler [PB02] followed this strategy to add details of one example to the output by filling the sparse DOFs of the key framed sequence in the frequency domain. Al-Ghreimil and Hahn [AGH03] applied body part replacement to throwing and walking examples such that a different throwing action is extracted as a partial motion and combined with a base walking motion to generate the variants. Ikemoto and Forsyth [IF04] used a direct transplantation of limbs between two motion examples. In their method, they designed a rule-based classifier that evaluates the naturalness of the transplanted results. Ashraf and Wong [AW00] divided a character body into a upper and a lower half and maintained consistently synchronized locomotion between the two halves via the decoupled interpolation. In a similar spirit, Heck *et al.* [HKG06] introduced a layered approach that focuses on preserving the cross-body correlation during the upper and lower body combination. Later, Oshita [Osh08] tried to maintain the correlation by adding torso vibrations, extracted from an action, to the lower body locomotion. In this type of motion composition, maintaining the kinematic correlation between different body parts, especially the upper against the lower one, becomes key to synthesizing a physically plausible locomotion. In our system, we preserve the correlation by detecting a series of foot phases which snapshot key moments of coordinated movements between the arms and legs into frames and then decompose the frames into the components per body part. This provides a user fine control of style composition by allowing combinations of the corresponding components from multiple examples without destroying the correlation preserved in the key frames.

3. Locomotion Analysis

Human locomotion is a complicated, but constrained activity. Assuming a locomotion cycle in the example set starts and ends with two consecutive foot steps of the same foot, we can assign *key frames* based on the important moments of foot contact on the ground called *foot phases* and use these as input data to the component decomposition. We detect 4 different key frames for each foot for the following foot phases: *heel-strike* (HS), *foot-flat* (FF), *push-off* (PO), and *toe-off* (TO). There are several benefits to using these for the locomotion analysis. First, the series of key frames encapsulates the gross style of the example, while its small number accelerates the decomposition process in ICA. Missing details between the key frames in the example sequence are stored as posture offsets and will be added later as part of the locomotion reconstruction. Next, the coordinated movements of arm and leg swings between different examples are aligned in the time domain with the corresponding key frames. Furthermore, the foot-plants are easily specified from the given key frames.

Foot Phases	Acceleration		Position	
	Ankle	Toe	Ankle	Toe
Heel-Strike (HS)	≈ 0	> 0	lowest	-
Foot-Flat (FF)	≈ 0	≈ 0	-	-
Push-Off (PO)	> 0	≈ 0	lowest	-
Toe-Off (TO)	> 0	> 0	-	lowest

Table 1: *Detection of foot phases: First, the positional acceleration of two foot joints crossing zeroes or being near zero is used to detect a frame as one of the foot phases. Next, if multiple frames are detected for the same foot phase, the geometric position of either an ankle or a toe joint is compared across all candidates to select a key frame. For the FF phase, we simply set the first detected one as the key frame.*

3.1. Key Frame Detection

The precise starting moment for each foot phase is ambiguous to detect from the unlabeled examples due to the noise in the data and retargeting errors [KGP02]. We expedite the process by detecting zero-crossings of acceleration of two foot joints, the ankle and toe, as shown in Table 1. A spatial filter, Laplacian of Gaussian, is applied prior to the detection to reduce the noise in the acceleration data. Depending on the style of foot contact, it is possible to detect multiple frames belonging to the same phase, where their acceleration crosses zero or approaches near zero. In these cases, the system selects one based on the geometric positions of two joints as shown in right columns of Table 1 or lets a user select a key frame from the multiple candidates.

3.2. Loop Cycle Generation

Looping a locomotion cycle is useful so that it can be concatenated repeatedly and will yield smooth locomotion without further blending [RGBC96]. Formally, this requires continuity between the starting frame, HS$_i$, and the ending frame, HS$_{i+1}$, of the loop cycle as shown in Figure 3. This is

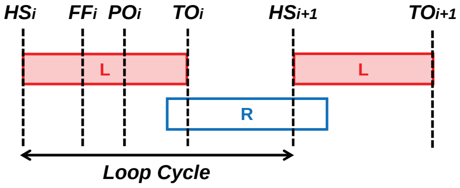


Figure 3: *Specification of a loop cycle: The shaded areas (red) are interpolated together and replaced the first stance phase to create the smooth loop continuity.*

ensured by pre-blending the start stance phase of one foot to the next stance phase of the same foot with the size of blending window, $N_b = min(\text{TO}_i - \text{HS}_i, \text{TO}_{i+1} - \text{HS}_{i+1})$. For the ith blending frame, $0 \leq i < N_b$, we linearly interpolate the root positions and perform *SLERP* [Sho85] on the joint rotations. We complete the loop by replacing the frames in the start stance phase, between HS_i and TO_i, with the blended frames. When the periods of two stance phases are significantly different, we first apply the dynamic time-warping technique [RCB98] to map the second stance to the first one based on the key-times of TO, FF, PO, and TO. This way, we preserve the foot constraints in the example stances during the interpolation.

3.3. Component Decomposition

The motion of an articulated character with a structure of hierarchical joints can be represented by interpolating a series of key frames and adding posture offsets as follows,

$$\text{M}(t) = [p(t) + \bar{p}(t), q_1(t) + \bar{q}_1(t), \ldots, q_n(t) + \bar{q}_n(t)], \quad (1)$$

where p is the world position of the root and q_i is the rotation of the ith joint with respect to its parent, represented by a unit quaternion. Here, $\bar{p}$ and $\bar{q}_i$ define transitional and rotational offsets which are the difference between the linearly interpolated frames between key frames and the corresponding frames in the original loop cycle. These posture offsets capture details of the example style and can be used as a separate editing parameter during the synthesis process.

Before a set of loop cycles is decomposed into sub-motion components via ICA, we first divide the character's body into several parts as shown in Figure 4. Grouping DOFs of a specific body part for the partial replacement of motion from one to another is an efficient technique [IF04, HKG06, Osh08] to expand the size of motion database. Inspired by this, our system extracts separate components from each body part, providing a user with detailed controls of mixing components from different examples and different parts of the body, yielding a rich range of style for each body part. Note that the root joint is excluded from the head and torso group since it is closely coupled with leg swings and needed to generate a base locomotion. During the synthesis process, the leg parts will be optimized to maintain the proper foot contacts. This will be further detailed in Section 4.

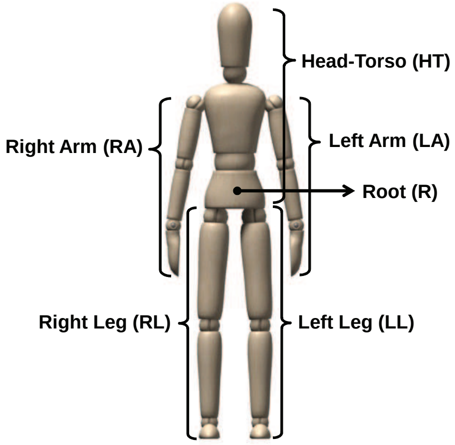


Figure 4: *Hierarchical ICA on separate body parts.*

Given a loop cycle for each example, ICA requires the key frames of all examples to be combined together as input in order to extract sub-motion components where each component represents a shared stylistic feature from each body part. For example, if we have N_k key frames in a loop cycle, we place them in the temporal order in the cycle and then concatenate all the loop cycles into an input matrix. In fact, we use the rotational components, q_i, of the key frames to form the initial version of this.

It is important to determine an appropriate joint angle representation to use with ICA because ICA is a statistical technique to solve for a linear combination of the source (hidden) variables [HO00]. Thus, using $q_i \in \mathbb{S}^3$ as input to the linear space of ICA in $\mathbb{R}^3$ can introduce undesirable artifacts in the joint rotations during the synthesis. Other joint rotation representations, such as Euler angles and exponential maps, are not suitable for a continuous motion [Gra98], while a direct input of joint positions in Euclidean coordinates does not maintain original bone lengths so it requires an additional post-processing step to preserve the physical constraints [SCF06]. For these reasons, we convert each q_i of key frames to its linearized approximation, $v_i \in \mathbb{R}^3$, by using the logarithm map and later convert back to q_i from v_i via the exponential map during the synthesis process [LS02].

Let N_l be the total number of loop cycles used as input to ICA. The number of total frames, N_f, from the concatenated loop cycles is $N_k \times N_l$. Given N_d DOFs in a body part, we can construct the following input matrix, $\mathbf{V}$, for ICA,

$$\mathbf{V} = \begin{bmatrix} v_{0,0} & v_{0,1} & \cdots & v_{0,N_f} \\ v_{1,0} & v_{1,1} & \cdots & v_{1,N_f} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N_d,0} & v_{N_d,1} & \cdots & v_{N_d,N_f}, \end{bmatrix} \quad (2)$$

Body Part (DOFs)	Sub-motion Components			
	ICs	SA	CO	TR
Torso-Head (12)	6	1, 3, 5, 6	2	4
Left Arm (9)	12	1, 2, 4, 6-8, 12	3, 5, 10, 11	9
Right Arm (9)	12	3-6, 9-11	1, 2, 7, 12	8
Left Leg (7)	8	1, 3, 5-7	2, 4, 8	-
Right Leg (7)	8	1-3, 5, 7	4, 6, 8	-

Table 2: *Sub-motion components decomposed from 10 examples: The number under sagittal (SA), coronal (CO), and transverse (TR) column is the component number categorized by its main direction of movements in that plane.*

where v_{ij} is the ith DOF in a key frame belonging to the {HS, FF, PO, TO} for both left and right foot. Due to its stability and fast performance, we adopted the FastICA algorithm [HO97] for the decomposition, which is based on the maximization of non-Gaussianity of input data.

A separating matrix, $\mathbf{W}$, is calculated by the FastICA algorithm, that allows $\mathbf{V}$ to be decomposed into the sub-motion components, $\mathbf{C}$, as follows:

$$\mathbf{C} = \mathbf{W}\mathbf{P}(\mathbf{V} - \hat{\mathbf{V}}), \tag{3}$$

where $\hat{\mathbf{V}}$ is a mean of the key frames for each DOF and $\mathbf{P}$ is a PCA whitening matrix which eliminates insignificant DOFs in the cycles and facilitates the decomposition calculation in the FastIC algorithm by providing a reduced dimensionality. The number of decomposed components and their characteristics are summarized in Table 2.

4. Locomotion Synthesis

Reconstructing a locomotion sequence from a set of sub-motion components is a straightforward process. Given the separating matrix, $\mathbf{W}$, we first recover the key frames belonging to each loop cycle as follows:

$$\mathbf{V} = \hat{\mathbf{V}} + \mathbf{W}^{-1}\mathbf{C}, \tag{4}$$

where $\mathbf{C}$ contains N_c components and the length of each component is N_k frames long, representing the key frames of one loop cycle. Note that $v_{ij} \in \mathbf{V}$ is converted back to the unit quaternion values, q_{ij}, via the exponential map during this reconstruction. A loop cycle can be reconstructed by interpolating the N_k temporally separated key frames from $\mathbf{V}$ and adding posture offsets to their in-between frames as in Equation 1.

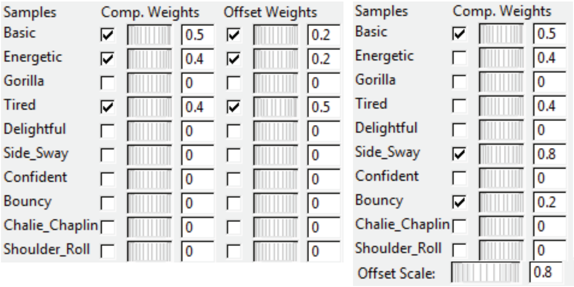
4.1. Control Parameters

The larger design question is how to mix or recombine the components available from the various input motions in order to generate new motion. Two corresponding components can be manipulated in a number of different ways. Mori and Hoshino [MH02, MH03] interpolated the components to vary motion style while Shapiro *et al.* [SCF06] merged or swapped the components to transfer the style of one locomotion to another. Both cases show that the components can be directly manipulated for modifying the input style. Inspired by this, we turn a large number of DOFs that must be set multiple times during a cycle, into a set of components that each span the entire cycle, making the editing process easier. However, due to the hierarchical decomposition, as shown in Table 2, our system generates a relatively large number of components to be controlled by a user. In this case, the visual evaluation of individual components [SCF06] for the sake of composition can be time-consuming. For this reason, we initiates the output synthesis based on user-specified weight values, w_j^R, for each input example j to be included, where R indicates the root body part. The user can also specify potentially different weights, o_j^R, which are used to linearly interpolate input examples in order to calculate the posture offsets discussed in Section 3.3. Next, weight values, w_j^p, for each body part p of the input examples are specified, along with a scaling value, s^p, for the offsets added to the output sequence, where $p \in$ {TH, LA, RA, LL, RL} (Figure 4). These interfaces are shown in Figure 5.

The output sequence is generated by first reconstructing a *base sequence* from using w_j^R for all components, which determines the overall speed and style of the output motion. This base sequence is further edited by adding desired torso-head, arm, and leg movements with w_j^p and then completed by adding the offsets scaled by s^p. These scaled offsets are dynamically time-warped to the output sequence if $o_j^R \neq w_j^R$. The offsets are an optional control that adds detail to the motion. Initially, the system distributes w_j^p across the sub-motion components in each body part. However, for the legs, w_j^{LL} and w_j^{RL}, should be distributed carefully to avoid a physically implausible output from the reconstruction. This becomes difficult, especially with the multiple components extracted from examples with different speeds. For example, if the components of leg parts from the walking motion are simply weighted with the corresponding components from the running motion, the output sequence will contain floating leg swings which are temporally unaligned with the ground contacts. Thus, we provide an optimization method for solving component weights for the foot parts which maintains reasonable ground contacts.

4.2. Component Ranks

Unlike PCA, the individual components from the input data set have no order of significance in their ranks. This means that we can reconstruct a locomotion sequence with any order of components in $\mathbf{C}$ from Equation 4. On the other hand, if we can prioritize the components based on their contribution to the desired output style in advance, we can facilitate the synthesis process by trying to distribute a weight value to



Samples	Comp. Weights		Offset Weights	
Basic	☑	0.5	☑	0.2
Energetic	☑	0.4	☑	0.2
Gorilla	□	0	□	0
Tired	☑	0.4	☑	0.5
Delightful	□	0	□	0
Side_Sway	□	0	□	0
Confident	□	0	□	0
Bouncy	□	0	□	0
Chalie_Chaplin	□	0	□	0
Shoulder_Roll	□	0	□	0

Samples	Comp. Weights	
Basic	☑	0.5
Energetic	□	0.4
Gorilla	□	0
Tired	□	0.4
Delightful	□	0
Side_Sway	☑	0.8
Confident	□	0
Bouncy	☑	0.2
Chalie_Chaplin	□	0
Shoulder_Roll	□	0
Offset Scale:		0.8

Figure 5: *User interfaces for the root part (left) and other body parts (right) to specify example and offset weights*

the highest ranked component first. For example, if we want more gorilla leg motion and we know that component 4 of the gorilla leg motion is particularly significant to this style, we can prioritize adding this component. However, it is difficult to evaluate such component contributions until the locomotion sequence is actually reconstructed with a selected component and visually inspected. It is from this perspective that we reconstruct test sequences with the purpose of automatically ranking each component's style contribution. When each test sequence, reconstructed by excluding one component at a time, is compared against the original loop cycle sequence, we can estimate how much each of the excluded components has influenced the range of movements in the original sequence.

Let C_{ij} denote the ith component of a particular body part from the jth loop cycle. Using Equation 4, we derive the positional differences of joints, $\Delta\mathbf{p}_i$, as follows,

$$\Delta\mathbf{p}_i = \mathbf{p}(\mathbf{V}_j) - \mathbf{p}(\hat{\mathbf{V}}_j + \sum_{i=1}^{N_c} \mathbf{W}_i^{-1}\mathbf{C}_{ij}), \tag{5}$$

where $\mathrm{C}_{ij} = 0$ for the ith test sequence and $\mathbf{p}(\cdot)$ estimates the joint positions from the rotational data in key frames. Thus, a larger $\Delta\mathbf{p}_i$ indicates that the excluded C_{ij} has more influence on the range of the movements in $\mathbf{V}_j$. These $\Delta\mathbf{p}_i$ values are used to rank the components.

4.3. Component Weight Optimization

An optimization process is used to adjust the w_j^P for two leg parts to achieve physically plausible motion. Let w_{kj} be the optimal weight for the kth leg component of the jth loop cycle, $\mathbf{C}_{kj}$, in each of the leg parts. Our objective is to find the w_{kj} such that the positional difference of foot constraints between the reconstructed sequence from w_{kj} and the base sequence from w_j^R is minimized. Specifically, if $d(w_{kj})$ is the distance of the foot constraints of the output sequence from the base one, we estimate w_{kj} for each component by solving the nonlinear optimization as follows,

$$\min_{w_{kj}} \sum_{j=1}^{N_l} d(w_{kj}), \tag{6}$$

where $k = [1 \ldots N_c]$ and N_c is the number of components in the left and right leg part respectively. To solve this, we utilize an iterative numerical approach, the conjugate gradient method [PTVF02]. Note that the rank of $\mathbf{C}_{kj}$ is predetermined from the previous section. We conduct the optimization from the highest to the lowest rank. Finally, the output sequence is reconstructed from Equation 4 with $\mathrm{C}_k = \sum_{j=1}^{N_l} w_{kj}\mathbf{C}_{kj}$ for the kth component in each of the leg parts.

4.4. Foot Constraint Enforcement

The visual artifact of foot-skating is often observed in the output locomotion due to errors in ankle positions in the example set or the misplacement of foot constraints during the composition. We utilize the lower-body IK routine suggested in [NK09, KN11] to enforce the foot constraints derived between the foot phases detected in Section 3.1.

Based on the period of stance phases, from HS to FF, we set the root foot position as the half way point between the start and end of the stance phases with the lowest height achieved during the stance phase. Holding the foot fixed for this duration can result in a discontinuity at the edge of each step. In most cases, this discontinuity takes place over a very short period of time; thus, linear interpolation between a constrained and an unconstrained frame is sufficient for generating a smooth transition between steps. Similarly, we apply the same strategy to the swivel rotations of the root leg to prevent spinning when the foot is on the ground.

5. Experimental Results

Experiments were performed on an Intel Core 2 Quad$^{\mathrm{TM}}$ 2.4GHz PC with 8GB memory. We used the same skeleton structure for all results consisting of 50 DOFs: 6 for the pelvis position and orientation, 12 for the torso and head orientation, 9 for each arm orientation, and 7 for each leg orientation. As the sample set, we captured 10 different locomotion clips which vary in style and speed as listed in Figure 5. All these input clips are sampled at the rate of 120 fps.

Component Decomposition

The decomposition of the given examples is summarized in Table 2. The number of components decomposed from each body part is actually larger than the number of DOFs except the torso-head part. This is mainly because the style difference between the given examples comes from the arm and leg parts while the torso and head movements are relatively subtle. It is worth noting that each component spans the entire motion, whereas the individual DOFs would need to be set many times during a cycle to create locomotion. Observations of the components reveal that corresponding components taken across different samples exhibit common stylistic feature, while individual components within a sample differ from each other. For example, the component 3's of the left arm part from the basic, energetic, and tired examples exhibit the similar arm swing in the coronal plane while the component 5's from the same examples exhibit the different arm swing in the sagittal plane. Similarly, the component 1's of the right leg part from the examples show a foot lift while the component 7's of the same group show a forward foot swing in the sagittal plane. A comparison between the components are shown in the accompanying video.

Overall, the number of components containing the movements in the sagittal plane is larger than two other planes. We only observed a few sub-motions in the transverse plane due to the characteristic feature of the forward movements throughout the example set. It took about 150s to decompose the 10 examples into a total of 480 components, 48 per example; however, it takes less than 20ms to reconstruct any motion from the decomposed components. Thus, the system is suitable for an interactive locomotion generation.

Locomotion Composition

We first tested the locomotion composition with different component weights on a single example. Using the basic example, we generated three different outputs with weights of 0.5, 1.0, and 1.5 at the same speed as in the example. The weight of 1.0 generates the same motion as the example motion while 0.5 and 1.5 generates smaller steps and larger steps respectively. Due to maintaining same body speed, we can observe reduced arm and leg swings with increased step frequency from the output with 0.5 and an increased swing size with a reduced frequency of steps from the output with 1.5. Both outputs maintain physically plausible foot contacts on the ground. These results are compared in Figure 6 and demonstrated in the accompanying video.

Next, we compared the outputs synthesized from the given weights against ones from the optimized weights in the leg parts. In this comparison, we used the gorilla and tired examples, exhibiting a large difference in the body speed. We synthesized two outputs with different speed: Using the speed from the tired example, we combined the torso-head and arm parts from the tired example with the leg parts from the gorilla example. Using the speed from the gorilla example, we combined them in an opposite way, the upper body of the gorilla with the under body of the tired one. We also did the opposite combinations. As shown in Figure 7, the outputs synthesized from the linear weights result in the floating or penetrating foot contacts while ones synthesized from the optimal weights show physically better results.

In addition, we generate different styles from the examples by scaling the weights for the given examples. To synthesize the basic walk in a relaxed style, we reduced the body speed and incorporated the examples with large rotations in the torso-head and arm parts in order to strengthen the weight

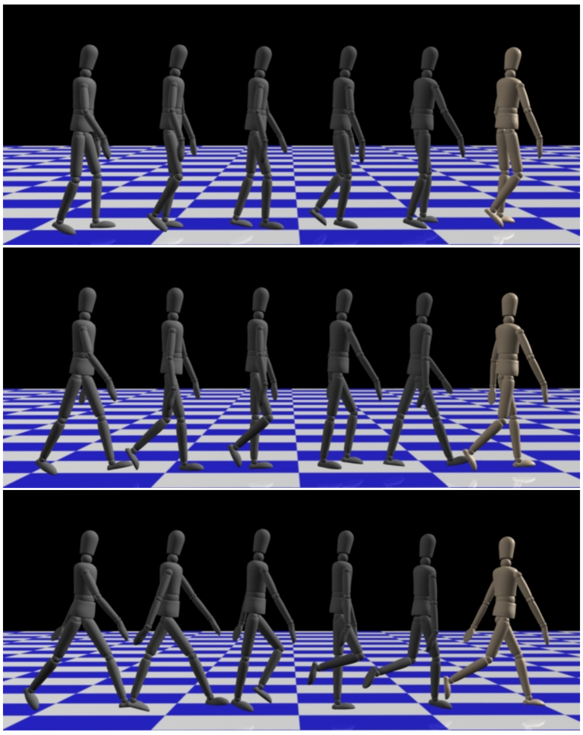


Figure 6: *Basic walks synthesized with component weights of 0.5 (top), 1.0 (middle), and 1.5 (bottom)*

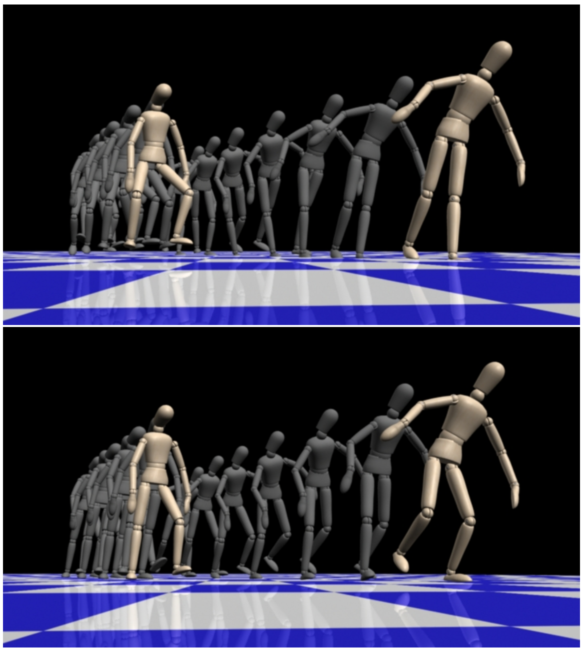


Figure 7: *Gorilla walks synthesized with given weights (top) and optimized weights (bottom): The optimal weights produce physically plausible results.*

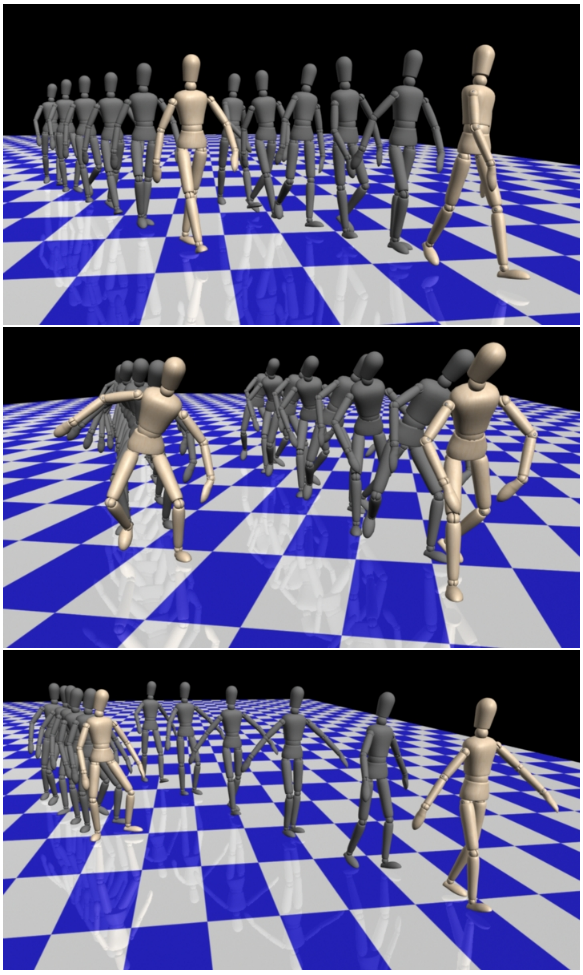


Figure 8: *Examples (left) are modified into a different style (right): Basic walk in a relaxed style (top), gorilla walk in a tired style (middle), and Charlie-chaplin walk in a rigid style (bottom)*

shift between the steps in the output. On the other hand, the gorilla walk in a tired style used the examples with decreased rotations in the torso-head and arm parts. Finally, we scaled down the offsets from the Charlie-chaplin example and replaced its body movements from more restricted examples such as the confident and basic examples to synthesize the Charlie-chaplin walk in a rigid style. Figure 8 compares these outputs and Table 3 shows more details on the examples and the weights used to achieve the style change.

User Case Study

Our system has been tested by a number of novice users who are not experienced in computer animation field. We asked a number of different undergraduate and graduate students to generate an output, given the instruction of producing a walk in a happy style by using the 10 examples. About two thirds of the participants successfully generate a reasonable result in about 20 to 30 minutes. The participants tended to use the examples from basic, energetic, delightful, bouncy, and shoulder roll during the synthesis. Some results are included in Figure 1 and also in the accompanying video.

Body Part	Examples (Weights)		
	Relaxed	Tired	Rigid
Root	Energetic (0.5) Tired (0.8)	Gorilla (0.8) Tired (0.5)	Charlie (0.8) Confident (0.4)
	Offset		
	Energetic (0.5) Tired (0.5)	Gorilla (1.0) Tired (0.5)	Charlie (0.5) Confident (0.8)
Torso-Head	Basic (1.0) Energetic (0.5)	Gorilla (0.5) Tired (0.3) Basic (0.5)	Basic (0.2) Confident (0.2)
	Offset Scale		
	0.1	1.0	0.3
Arms	Energetic (0.5) Tired (0.5) Delightful (1.0)	Gorilla (0.8) Tired (0.5) Shoulder (0.2)	Bouncy (0.2) Basic (0.4) Gorilla (0.4)
	Offset Scale		
	1.0	1.0	0.3
Legs	Basic (0.2) Tired (0.8) Energetic (0.4)	Gorilla (0.8) Tired (0.5)	Basic (0.4) Confident (0.2)
	Offset Scale		
	1.0	0.5	0.5

Table 3: *Examples with their weight values used to modify the input example into a different style*

6. Conclusion

In this paper, we introduced a component-based system for locomotion composition that is capable of synthesizing output sequences that are stylistically different from the example set. To provide rich varieties of output style from the small example set, the proposed system expanded the number of combinatorial components by decomposing the examples into separate body parts and maintained the key correlation throughout the body by using the detected foot phases. This way, a user can efficiently generate a desired output with a small number of control parameters while the system solves for the physically plausible locomotion by finding the optimal weights for the leg components. Our experimental results show that our system is suitable for novice users who do not have much experience in motion generation and want to compose new locomotion clips from multiple examples in a quick and intuitive way.

Currently, our system works best for the cyclic locomotion sequences, where the correlation is most strong between the arm and leg swings. For acyclic examples, the system should detect multiple foot cycles depending on the key style to detect and looks for additional information from other end effectors such as a head and hands, which establish more comprehensive correlations throughout the body. For such examples, the decomposition via ICA can result in a larger number of sub-motion components so it might require steps to reduce the dimensionality of components by grouping similar components via a nearest neighbor search algorithm or applying localized PCA which captures the spatially distinct changes of style better.

In addition, our system only enforces the physical constraints during the foot contacts. Imposing user-specified constraints on other body parts during the qualitative edits will require a postprocessing motion filter such as the per-frame constraint solver of [TK05], which enhances the physical validity of the synthesized locomotion.

Acknowledgments

Financial support for this research was provided in part by NSF grant 0845529 and through a software donation from Autodesk. Thanks to Jonathan Graham for model support.

References

[AGH03] Al-Ghreimil N., Hahn J. K.: Combined partial motion clips. In *In the 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision* (Feb 2003). 1, 3

[AW00] Ashraf G., Wong K. C.: Generating consistent motion transition via decoupled framespace interpolation. In *Proceedings of Eurographics* (2000). 1, 3

[CFP03] Cao Y., Faloutsos P., Pighin F.: Unsupervised learning for speech motion editing. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 225–231. 1, 2

[Gra98] Grassia S.: Practical parameterization of rotations using the exponential map. *Graphics Tools 3*, 3 (1998), 29–48. 4

[HKG06] Heck R., Kovar L., Gleicher M.: Splicing upper-body actions with locomotion. *Computer Graphics Forum 25*, 3 (2006), 459–466. 1, 3, 4

[HO97] Hyvarinen A., Oja E.: A fast fixed-point algorithm for independent component analysis. *Neural Computation 9*, 7 (1997), 1483–1492. 5

[HO00] Hyvarinen A., Oja E.: Independent component analysis: Algorithms and applications. *Neural Networks 13*, 4–5 (2000), 411–430. 4

[IF04] Ikemoto L., Forsyth D.: Enriching a motion collection by transplanting limbs. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), pp. 99–108. 1, 3, 4

[KGP02] Kovar L., Gleicher M., Pighin F.: Motion graphs. *ACM Transactions on Graphics 21*, 3 (2002), 483–490. 3

[KN11] Kim Y., Neff M.: Automating expressive locomotion generation. *Transactions on Edutainment VII, LNCS 7145* (2011), 48–61. 6

[LPL08] Liu G., Pan Z., Lin Z.: Style subspaces for character animation. *Computer Animation and Virtual Worlds 19*, 3–4 (2008), 199–209. 1, 2

[LS02] Lee J., Shin S. Y.: General construction of time-domain filters for orientation data. *IEEE Transactions on Visualization and Computer Graphics 8*, 2 (2002), 119–128. 4

[LXPR11] Liu G., Xu M., Pan Z., Rhalibi A. E.: Human motion generation with multifactor models. *Computer Animation and Virtual Worlds 22*, 4 (2011), 351–358. 1, 2

[MH02] Mori H., Hoshino J.: Independent component analysis and synthesis of human motion. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* (2002), pp. 3564–3567. 1, 2, 5

[MH03] Mori H., Hoshino J.: Ica-based interpolation of human motion. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation* (2003), pp. 453–458. 1, 2, 5

[MLC10] Min J., Liu H., Chai J.: Synthesis and editing of personalized stylistic human motion. In *Proceedings of Symposium on Interactive 3D Graphics* (2010), pp. 39–46. 1, 3

[NK09] Neff M., Kim Y.: Interactive editing of motion style using drives and correlations. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), pp. 103–112. 6

[Osh08] Oshita M.: Smart motion synthesis. *Computer Graphics Forum 27*, 7 (2008), 1909–1918. 1, 3, 4

[PB02] Pullen K., Bregler C.: Motion capture assisted animation: Texturing and synthesis. *ACM Transactions On Graphics 21*, 3 (2002), 501–508. 1, 3

[PTVF02] Press W. H., Tukolsky S. A., Vetterling W. T., Flannery B. P.: *Numerical Recipes in C++: The Art of Scientific Computing*, second ed. Cambridge University Press, New York, NJ, 2002. 6

[RCB98] Rose C., Cohen M. F., Bodenheimer B.: Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications 18*, 5 (1998), 32–40. 4

[RGBC96] Rose C., Guenter B., Bodenheimer B., Cohen M.: Efficient generation of motion transitions using spacetime constraints. *Proceedings of ACM SIGGRAPH* (1996), 147–154. 3

[SCF06] Shapiro A., Cao Y., Faloutsos P.: Style components. In *Proceedings of Graphics Interface* (2006), pp. 33–39. 1, 2, 4, 5

[Sho85] Shoemake K.: Animating rotation with quaternion curves. In *Proceedings of ACM SIGGRAPH* (1985), pp. 245–254. 4

[TK05] Tak S., Ko H.: A physically-based motion retargeting filter. *ACM Transactions on Graphics 24*, 1 (2005), 98–117. 9

[UGB*04] Urtasun R., Glardon P., Boulic R., Thalmann D., Fua P.: Style-based motion synthesis. *Computer Graphics Forum 23*, 4 (2004), 799–812. 1, 2

[Vas02] Vasilescu M. A. O.: Human motion signatures: Analysis, synthesis, recognition. In *Proceedings of International Conference on Pattern Recognition* (2002), pp. 456–460. 1, 2

[VT05] Vasilescu M. A. O., Terzopoulos D.: Multilinear independent components analysis. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005), pp. 547–553. 1, 2