Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2005)
K. Anjyo, P. Faloutsos (Editors)

Motion Modeling for On-Line Locomotion Synthesis

Taesoo Kwon and Sung Yong Shin†

Korea Advanced Institute of Science and Technology

Abstract

In this paper, we propose an example-based approach to on-line locomotion synthesis. Our approach consists of two parts: motion analysis and motion synthesis. In the motion analysis part, an unlabeled motion sequence is first decomposed into motion segments, exploiting the behavior of the COM (center of mass) trajectory of the performer. Those motion segments are subsequently classified into groups of motion segments such that the same group of motion segments share an identical footstep pattern. Finally, we construct a hierarchical motion transition graph by representing these groups and their connectivity to other groups as nodes and edges, respectively. The coarse level of this graph models locomotive motions and their transitions, and the fine level mainly captures the cyclic nature of locomotive motions. In the motion synthesis part, given a stream of motion specifications in an on-line manner, the motion transition graph is traversed while blending the motion segments to synthesize a motion at a node, one by one, guided by the motion specifications. Our main contributions are the motion labeling scheme and a new motion model, embodied by the hierarchical motion transition graph, which together enable not only artifact-free motion blending but also seamless motion transition.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Animation

1. Introduction

Demand for on-line, real-time motion synthesis has ever been increasing, propelled mainly by computer games and VR applications. Recent technological advancement in motion capture and reuse increases the potential for on-the-fly motion synthesis to replace playback of prerecorded motions according to scenarios. There have been two main research streams in real-time motion synthesis: motion blending [GR96, PSS02, PSS04, RCB98, WH97, KG04] and motion rearrangement [AF02, AFO03, GJH00, KGP02, LCR*02, LWS02, PB02, TH00]. The former increases efficiency and controllability, whereas the latter retains naturalness embedded in captured motions.

A hybrid approach has recently been proposed to combine the idea of motion blending and that of motion rearrangement. In this approach, motion transition graphs are instrumental in combining these ideas, as demonstrated in locomotive motion generation [PSS02, PSS04] and rhythmic motion synthesis [KPS03]. The origin of these graphs is traced back to “verb graphs" in [RCB98], which were used mainly for motion style control. Later, Park et al. [PSS02, PSS04] enhanced them for on-line motion blending.

A node in a motion transition graph represents a group of basic motions of an identical structure, and an edge represents the transition from a blended motion to a blended motion (possibly including self-transition). Given a stream of motion specifications, the graph is traversed from node to node, while blending motions at nodes and making transitions at edges.

To facilitate the hybrid approach, the major premise is the availability of labeled motion data satisfying the following two properties.

- The group of motions at each node should have an identical structure for motion blending.
- The group of motions at a node should transit seamlessly to that of motions at a node connected by an edge (possibly a self-edge).

A straightforward way to prepare these motion data would be to direct a performer to repeat motions with an identical structure but different aspects. Aside from their accuracy, however, the resulting captured motions would be quite unnatural. A better approach would be to let the

† syshin@jupiter.kaist.ac.kr

subject perform any desired actions within a given motion category to capture an unlabeled motion sequence, and then to acquire labeled motion clips from the sequence. In [RCB98, PSS02, PSS04], it was assumed that labeled motion clips are available. Kim et al. [KPS03] proposed an automatic labeling scheme for rhythmic motions to construct their motion transition graphs, exploiting motion beats and rhythmic patterns embedded in the motions.

In this paper, inspired by the work in [KPS03], we propose a novel on-line motion synthesis approach for non-rhythmic motions, in particular, locomotive motions such as running, walking, and standing, together with their transitions. Unlike rhythmic motions, reference temporal patterns such as motion beats and rhythmic patterns are not available for these motions in general. Exploiting biomechanical results on human COM (center of mass) trajectories and footstep patterns, we first cut an unlabeled motion sequence into motion segments to identify motion units called basic movements. We then classify those motion segments into groups of motion segments such that the same group of motion segments share an identical structure while extracting their parameters simultaneously. Finally, we construct a motion model, based on a novel structure called a hierarchical motion transition graph. We also enhance the motion blending scheme in [PSS02, PSS04] to better handle motion dynamics and transition.

2. Related Work

There have been rich results on real-time motion synthesis based on motion capture and reuse. Our work is directly related to those on motion blending to facilitate on-line locomotion generation.

Guo and Robergé [GR96] and Wiley and Hahn [WH97] initiated motion blending by providing techniques for interpolating example motions sampled regularly in parameter spaces. Rose et al. [RCB98] proposed a framework of motion blending based on scattered data interpolation with radial basis functions. They introduced a verb graph for motion transition. Later, Sloan et al. [SRC01] adopted cardinal basis functions for further acceleration. Despite its superb efficiency, this approach was intended for real-time, but not for on-line motion synthesis.

Park et al. [PSS02, PSS04] have enhanced the framework of Rose et al. [RCB98] for on-line locomotion blending. Their most important contribution is arguably to model labeled motion clips available in a motion library, by incorporating motion rearrangement [AF02, AFO03, GJH00, KGP02, LCR*02, LWS02, PB02, TH00] into the framework of motion blending, based on a motion transition graph. However, the authors did not address how to obtain the labeled motion clips to construct the motion transition graph.

Recently, Kim et al. [KPS03] also adapted the motion transition graph [PSS02] for on-line rhythmic motion synthesis. Given an unlabeled example motion sequence, their approach decomposes the sequence into a set of motion segments called “basic movements" and clusters them to obtain a collection of labeled sets of basic movements by exploiting their rhythmic structures. The authors modeled the rhythmic motion sequence as a motion transition graph, where a node and an edge represent a set of labeled movements and the transition from one labeled movement set to a movement set (possibly itself). Unfortunately, this method is not applicable to non-rhythmic motions.

Although there have been rich research results on motion segmentation in computer vision and computer grahpics, a few results have dealt with full 3D human motions. Bindiganavale and Badler [BB98] employed the notion of zero-crossing in order to detect moments of interaction with environments. Fod et al. [FMJ02] presented a scheme to find a set of motion primitives from a sequence of motion data, based on principal component analysis (PCA) and K-means clustering. They segmented the motion sequence at the moments of zero-crossing of the angular acceleration. On top of the zero-crossing moments, Kim et al. [KPS03] exploited regular temporal patterns for labeling rhythmic motions, which is not easily generalized for other types of motions.

Arikan et al. [AFO03] employed a support vector machine classifier to interactively annotate motion data. Although this semi-automatic classifier works well for motion annotation, the results may not be precise enough for motion blending. Jenkins and Mataric [JM03], Barbit et al. [BSP*04] proposed automatic motion classification methods. However, their goals are not automatic motion blending and transition. Exploiting the temporal correspondence between motions, Kovar and Gleicher [KG04] proposed an automatic scheme to extract motion segments that are similar to a query motion. This scheme assumes the availability of query motions; otherwise, they would be specified manually. As pointed out by the authors, the scheme sometimes requires manual filtering to obtain blendable motions and does not take into account motion transition.

3. Overview

Generalizing the work in [KPS03], our framework for on-line locomotion generation consists of two major components: example motion preprocessing and run-time motion synthesis.

In example motion preprocessing, a motion transition graph is constructed from unlabeled, captured motions. Our main contributions lie in motion modeling; in particular, a motion labeling scheme and a hierarchical motion transition graph. The motion labeling scheme spans both motion segmentation and classification. Given an unlabeled motion sequence of a human-like articulated figure, the sequence is decomposed into motion segments based on the variation of the vertical component of the COM trajectory of the figure. Those motion segments are then classified into groups of basic movements such that the motions in the same group share

an identical motion type, exploiting the biomechanical observations on footstep patterns. We also propose a new motion transition graph to incorporate a motion hierarchy and transition nodes. The former is to represent the cyclic nature of locomotion: cyclic motions at the coarse level such as running and walking are represented by combining primitive motion units at the fine level, that is, basic movements. The latter is for seamless transitions among locomotive motions. To our knowledge, the proposed approach provides the first automatic modeling scheme for locomotive motions allowing both motion blending and transition.

In runtime motion synthesis, given a stream of motion specifications in an on-line manner, our system generates the corresponding locomotive motions while traversing our new motion transition graph. We address issues in motion dynamics and transition that arise from the cyclic nature of locomotion. In particular, we achieve a convincing motion prescribed by a given parameter vector, exploiting locomotion-specific characteristics, such as dependency between speed and turning angles, learned from example motion data. We also add acceleration as a new parameter to reflect some dynamical aspects of locomotion such as leaning due to sudden speed change.

The remainder of this paper is organized as follows: In Section 4, we describe how to construct our motion transition graph with an emphasis on motion labeling. In Section 5, we present our on-line locomotion synthesis scheme based on the motion transition graph. We present results in Section 6 and discuss issues and limitations in Section 7. We finally conclude this paper in Section 8.

4. Motion Analysis

In this section, we propose an automatic method to construct a motion transition graph, given unlabeled locomotion data. The locomotion data is first decomposed into motion segments, and then these segments are classified into groups of basic movements of identical structures. The collection of basic movement groups and their connectivity are mapped onto the node and edge sets of a motion transition graph, respectively.

4.1. Preliminaries

We represent a captured (unlabeled) human motion $\mathbf{M}$ as a sequence of postures sampled at discrete times called frames. The posture at each frame is described by a tuple, $(\mathbf{p}, \mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_J)$, where $\mathbf{p} \in \mathcal{R}^3$ and $\mathbf{q}_1 \in \mathcal{S}^3$ specify the position and orientation of the root, which is the pelvis in our case, $\mathbf{q}_j \in \mathcal{S}^3$ gives the orientation of joint j, and J is the number of joints in $\mathbf{M}$.

Motion Half Cycles: Our criteria for motion segmentation are three-fold:

1. Every motion segment should be simple enough to have an intuitive parametrization.
2. Every motion segment should be long enough to contain meaningful motion semantics.
3. An important motion feature should not be split into consecutive motion segments.

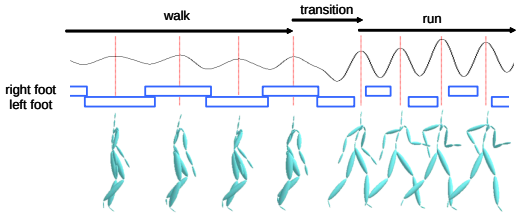

Figure 1: *The vertical component of the COM and corresponding poses*

Locomotive motions such as walking and running exhibit an inherent cyclic nature. Because of this nature, locomotion cycles would be apparent candidates for segmentation units that satisfy these criteria. However, each cycle of such a motion is composed of two half cycles initiated by left and right footsteps, respectively. Moreover, the characteristics of the half cycles are quite different to violate criterion 1 if they are parameterized as a single unit. Hence, we choose motion half cycles as basic segmentation units for cyclic motions such as walking and running.

Biomechanical Observations: For motion segmentation, we rely on biomechanical literature [Per92, Win90] , guided by criterion 3. As illustrated in Figure 1, it is well-known in biomechanics that motions such as walking and running have quite different COM trajectory patterns.

A running motion is composed of two stages: constrained and unconstrained stages. In the constrained stage, one of the feet contacts the ground, which causes physical interaction involving contact and friction forces. Thus, this stage has important motion features. In the unconstrained stage, neither foot contacts the ground. With no external forces exerted except gravity, the COM trajectory exhibits a parabolic curve. Thus, the COM trajectory has a single vertical peak(local maximum) in this stage. The motion segment, which is delineated by the frames with a pair of consecutive peaks of unconstrained stages, forms a half cycle of a run motion. This half cycle completely contains a constrained stage with important motion features, thus satisfying criterion 3.

Unlike a running motion, a walking motion consists of a single constrained stage, and thus yields a quite different COM trajectory pattern. By a similar analysis to a running motion, however, we can see that a pair of consecutive peaks(at the mid-stance of a single foot support phase) of constrained stages delineate a half cycle containing important motion features(at a double foot support phase).

Regardless of locomotion types, the vertical velocity of the COM vanishes at a peak. Thus, a peak of the COM occurs near the middle of a single limb support phase of a constrained stage or a flight phase (with no limbs supported) of

an unconstrained stage. Thus, we can identify the type of locomotion at the peak, which will be exploited in motion classification. On the contrary, critical interactions with the ground occur either in the double limb support phase while walking or in the single limb support phase while running, thus resulting in important motion features.

A transition motion between two different motions shares the biomechanical characteristics with both of the motions. For example, the first half of a walk-to-run transition motion resembles a walking motion and the last half resembles a running motion, as illustrated in Figure 1. The COM further sinks down to a valley at the constrained stage after the walking motion to store the energy for the unconstrained stage of the running motion.

4.2. Motion Segmentation

We use the behavior at the COM trajectory of an articulated figure to segment an unlabeled motion sequence **M**. Let $\mathbf{c}_\mathbf{M}(t)$ be the COM trajectory that gives the COM at time t, that is,

$$\mathbf{c}_\mathbf{M}(t) = \frac{\sum_i m_i \mathbf{r}_i(t)}{\sum_i m_i}, \tag{1}$$

where m_i and $\mathbf{r}_i(t)$ are the mass and COM position, respectively, for link i of the articulated figure at time t. The mass of each link is obtained based on biomechanical data in [Win90, SP05]. The velocity $\mathbf{v}_M$ of the COM of the figure is

$$\mathbf{v}_\mathbf{M}(t) = \frac{d\mathbf{c}_\mathbf{M}(t)}{dt}. \tag{2}$$

The vertical velocity of the COM vanishes at all peak of the COM trajectory $\mathbf{c}_\mathbf{M}(t)$, which is a zero-crossing moment of the velocity. We can detect all peaks by computing the discrete vertical velocity of the COM at every frame.

Without loss of generality, suppose that the motion sequence M is initiated and ended by a pair of peaks. Otherwise, we can cut off the initial and final segments, each of which is a short segment. Every motion segment, which is delineated by a pair of consecutive peaks, is not necessarily a half cycle since we need to consider non-cyclic motions such as standing motions and transitions motions.

Let $\mathbf{T} = \{t_0, t_1, \cdots, t_r\}$ be the sequence of peaks embedded in $\mathbf{c}_\mathbf{M}(t)$. We first identify all standing motion segments by detecting every frame with a stand pose. Excluding the standing motion segments, we cut the remaining portion of **M** into motion segments with the sequence of peaks. Together with the standing motion segments, they give the set of motion segments, $\mathbf{S} = \{\mathbf{s}_0, \mathbf{s}_1, \cdots, \mathbf{s}_k\}$ such that $\mathbf{M} = \mathbf{s}_0||\mathbf{s}_1||\cdots||\mathbf{s}_k$, where $||$ is a concatenation operator. In principle, $\mathbf{s}_i, 0 \le i \le k$ should be a half cycle if it is neither a standing motion segment nor a transition motion. However, this may not be true in practice due to the approximation error of the COM trajectory and the noise from motion capture. We will return to this issue while classifying the motion segments in **S**.

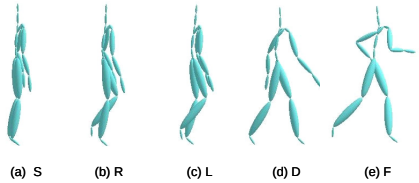

Figure 2: *The representative pose of each phase*

4.3. Motion Classification

In this section, we describe how to classify the set of motion segments, $\mathbf{S} = \{\mathbf{s}_0, \mathbf{s}_1, \cdots, \mathbf{s}_k\}$ into a collection of groups of basic movements of identical structures so that the basic movements in the same group are blendable without artifacts. To achieve this goal, we exploit biomechanical observations on footstep patterns.

Footstep Patterns: Footsteps characterize locomotive motion stages. In particular, a foot contacts the ground in a constrained stage, and neither foot contacts the ground in an unconstrained stage. The unconstrained stage has a single flight phase. However, depending on the foot (feet) contacting the ground, the constrained stage is classified into three phases: a left foot support phase, a right foot support phase, and a double support phase. To address a standing motion, we add a stand phase. In summary, we have five locomotion phases characterized with footstep patterns: a left foot support phase (L), a right foot support phase(R), a double support phase(D), a flight phase(F), and a stand phase(S), where the symbols in parentheses denote the corresponding phases. The representative pose of each phase is illustrated in Figure 2.

String Mapping: Our strategy of motion labeling is to encode each motion segment in S as a string of symbols in $\Sigma = \{\mathrm{L}, \mathrm{R}, \mathrm{D}, \mathrm{F}, \mathrm{S}\}$, and to classify the motion segments with the same string into the same group. The reason that we use the strings rather than the motion segments is two-fold: First, we can avoid troublesome time-warping to align the characteristic features. Second, string processing is more robust than numerical computation. To assign a string to a motion segment, we first extract the footsteps from the motion segment, employing the constraint detection scheme in [LP02], based on the observation that a contact foot on the ground maintains its height at the ground level and zero velocity for some consecutive frames. This scheme worked so well that few footsteps are missing in our experiments.

Given the footstep sequence, we can identify all phases in each motion segment. We first mark the portion of the motion **M** containing the S phases, based on the fact that an S phase consists of a sequence of poses with a fixed COM position that lasts longer than a threshold time. We then detect the F and D phases to mark the remaining portion of

M. A distinct D phase occurs when both feet contact the ground for a time interval. In an F phase, the feet are above the ground, which one might expect to be easily recognizable. In practice, some D phases could be misclassified into F phases at times, since a foot contact status is checked rather conservatively. To prevent this misclassification, we use the COM trajectory together with the foot contact status. Suppose that no foot contact the ground for a portion of **M**. If the peak of the COM trajectory is closer to the center of the portion than the valley is, then this portion is identified as an F phase. Otherwise, it is identified as a D phase. Finally, we identify the L and the R phases to further flag the unmarked portion of **M**. In both phases, only one foot contacts the ground. While scanning the motion segment, we divide it into phases and assign a symbol to each phase to obtain a string.

We would have a total of 17 strings encoding all basic movements as summarized in Table 1. For example, a walking motion has two strings, LDR and RDL, representing half cycles beginning with left and right support feet, respectively. These half cycles end with right and left support feet, respectively after a double support phase. A walk-to-run transition motion also has two strings, LDRF and RDLF, which are not half cycles. These strings show the characteristics of transition motions; in particular, the three-symbol prefix of each string encodes a walk motion, and the single-symbol suffix a run motion. All strings are self-explanatory, and thus are not further described here.

Table 1: *The strings for basic movements*

strings	type code	motion type	basic movement (ground contact status)	half cycle
LDR	WK	walk	initial support by left foot	yes
RDL			initial support by right foot	yes
FLF	RN	run	middle support by left foot	yes
FRF			middle support by right foot	yes
S	ST	stand	-	no
LDRF	WR	walk to run transition	initial support by left foot	no
RDLF			initial support by right foot	no
FRDL	RW	run to walk transition	final support by left foot	no
FLDR			final support by right foot	no
LDRS	WS	walk to stand transition	initial support by left foot	no
RDLS			initial support by right foot	no
SRDL	SW	stand to walk transition	final support by left foot	no
SLDR			final support by right foot	no
SLDRF	SR	stand to run transition	initial support by left foot	no
SRDLF			initial support by right foot	no
FRDLS	RS	run to stand transition	final support by left foot	no
FLDRS			final support by right foot	no

Refinement: Ideally, a motion segment would be a basic movement, that is, a string encoding a motion segment would be a string listed in Table 1. However, we may have unexpected strings that encode some motion segments in **S**. Therefore, those strings should be post-processed. Remember that the unlabeled motion sequence **M** consists of motion segments $\mathbf{s}_i, 0 \le i \le k$ in **S**, that is, $\mathbf{M} = \mathbf{s}_0||\mathbf{s}_1||\cdots||\mathbf{s}_k$. Without loss of generality, the string $f(\mathbf{s}_i), 0 \le i \le k$ that encodes a motion segment $\mathbf{s}_i$ is not in Table 1. Consider a tuple, $(f(\mathbf{s}_{i-1}), f(\mathbf{s}_i), f(\mathbf{s}_{i+1}))$, where $f(\mathbf{s}_{-1}) = f(\mathbf{s}_{k+1}) = \varepsilon$ denotes an empty string. Empirically, such a motion segment $\mathbf{s}_i$ is most likely from a high-to-low speed transition such as a run-to-walk, a run-to-stand, or a walk-to-stand transition. Specifically, a small COM peak may occur while making the transition. Such a peak would divide a transition segment into smaller ones, resulting in invalid strings. Therefore, we concatenate these strings by ignoring the troublesome peak. The new string will be $f(\mathbf{s}_{i-1}||\mathbf{s}_i)$ or $f(\mathbf{s}_i||\mathbf{s}_{i+1})$, whichever is a valid string encoding a high-to-low speed transition.

An invalid string also is produced very rarely in a low-to-high transition by missing a COM peak in a single foot or a flight phase. In this case, we split $\mathbf{s}_i$ into two parts, $\mathbf{s}_i^1$ and $\mathbf{s}_i^2$ such that $f(\mathbf{s}_i^1)$ and $f(\mathbf{s}_i^2)$ are valid, by enumerating all possible cases.

After the suggested string refinement, we rarely encounter invalid strings. The remaining invalid strings, if any, are discarded. Further refinement would result in a motion segment, whose COM trajectory is quite different from the others with the same string, which would later cause artifacts in motion blending.

Let **B** be the sequence of motion segments after refinement, that is,

$$\mathbf{B} = (\mathbf{m}_0, \mathbf{m}_1, \cdots, \mathbf{m}_b). \tag{3}$$

M is not necessarily equal to $\mathbf{m}_0||\mathbf{m}_1||\cdots||\mathbf{m}_b$ since some motion segments $\mathbf{s}_i, 1 \le i \le k$ in **M** could be discarded during refinement. However, $\mathbf{m}_i, 0 \le i \le b$ is a basic movement, and thus $f(\mathbf{m}_i)$ is a valid string. By string mapping and refinement, we have classified the motion segments in **B** into a maximum of 17 groups of basic movements such that the basic movements in each group share the same string. For future reference, we denote **C** as the collection of such movement groups, that is,

$$\mathbf{C} = (\mathbf{g}_0, \mathbf{g}_1, \cdots, \mathbf{g}_c), \tag{4}$$

where $\mathbf{g}_i, 0 \le i \le c < 17$ is a group of motion segments.

4.4. Motion Parametrization and Keytime Extraction

In this section, we first describe how to parameterize a motion segment and then explain how to extract the keytimes for feature alignment.

Parametrization: A motion segment can be parameterized in different ways depending on how it will be used. Our objective for motion parametrization is on-line locomotion control. Therefore, motion parameters should include the information such as the type of motion, the destination or direction of the motion, and the speed of the motion.

We identify a motion segment **m** in the set **B** with a parameter vector,

$$\mathbf{p}(\mathbf{m}) = (t(\mathbf{m}), ft(\mathbf{m}), v(\mathbf{m}), \theta(\mathbf{m}), a(\mathbf{m})), \tag{5}$$

where $t(\mathbf{m})$, $ft(\mathbf{m})$, $v(\mathbf{m})$, $\theta(\mathbf{m})$ and $a(\mathbf{m})$ denote the motion type, the foot contact status, the speed, the turning angle, and the acceleration of $\mathbf{m}$, respectively. The motion type and the foot contact status are directly obtained from Table 1 by table search with the string representing $\mathbf{m}$. We will explain later how to derive $v(\mathbf{m})$, $\theta(\mathbf{m})$ and $a(\mathbf{m})$.

Through the manner in which we have segmented the unlabeled motion $\mathbf{M}$, a motion segment $\mathbf{m}$ is well characterized by its own COM trajectory $\mathbf{c_m}(t)$. We extract the parameters of $\mathbf{m}$ from $\mathbf{c_m}(t)$. Let $\bar{\mathbf{c}}_\mathbf{m}(t)$ be the projection of $\mathbf{c_m}(t)$ onto the ground. The speed $v(\mathbf{m})$ is defined as follows:

$$v(\mathbf{m}) = \frac{1}{N(\mathbf{m})} \sum_{j=1}^{N(\mathbf{m})} ||\bar{\mathbf{c}}_\mathbf{m}(j) - \bar{\mathbf{c}}_\mathbf{m}(j-1)||, \qquad (6)$$

where $N(\mathbf{m})$ is the number of frames in $\mathbf{m}$ and $\bar{\mathbf{c}}_\mathbf{m}(0)$ is linearly extrapolated. $v(\mathbf{m})$ approximates the average speed along $\bar{\mathbf{c}}_\mathbf{m}(t)$.

Letting $\mathbf{T_m}(j) = \frac{\bar{\mathbf{c}}_\mathbf{m}(j) - \bar{\mathbf{c}}_\mathbf{m}(j-1)}{||\bar{\mathbf{c}}_\mathbf{m}(j) - \bar{\mathbf{c}}_\mathbf{m}(j-1)||}$, the acceleration parameter $a(\mathbf{m})$ is given by

$$a(\mathbf{m}) = \frac{1}{N(\mathbf{m})} \sum_{j=1}^{N(\mathbf{m})} \left(\frac{d\bar{\mathbf{c}}_\mathbf{m}(j)}{dt} - \frac{d\bar{\mathbf{c}}_\mathbf{m}(j-1)}{dt} \right) \cdot \mathbf{T_m}(j), \quad (7)$$

which approximates the average tangential acceleration of $\bar{\mathbf{c}}_\mathbf{m}(t)$. If $\bar{\mathbf{c}}_\mathbf{m}(j) = \bar{\mathbf{c}}_\mathbf{m}(j-1)$, then $\mathbf{T_m}(j)$ is assumed to be a zero vector. By incorporating $a(\mathbf{m})$, forward (or backward) leaning arising from sudden speed change is implicitly parameterized.

Finally, to define the turning angle $\theta(t)$, let $(i_0, i_1, \cdots, i_h)$ be a maximal subsequence of frames in $\mathbf{m}$ such that $\bar{\mathbf{c}}_\mathbf{m}(i_j) \neq \bar{\mathbf{c}}_\mathbf{m}(i_{j-1})$ for all $1 \leq j \leq h$. Then,

$$\theta(\mathbf{m}) = \frac{1}{N(\mathbf{m})} \sum_{j=1}^{h} \sin^{-1} ||\mathbf{T_m}(i_j) \times \mathbf{T_m}(i_{j-1})||. \quad (8)$$

$\theta(\mathbf{m})$ approximates the average turning angle for $\mathbf{m}$. Since $\mathbf{m}$ consists of a small number of frames and the turning angle at each frame is very small, $\theta(\mathbf{m})$ is unambiguously defined. In addition to explicitly parameterizing the average turning angle, $\theta(\mathbf{m})$ also reflects implicitly lateral leaning due to turning.

Keytime Extraction: Every keytime of a motion segment is a moment at which a motion feature occurs. The keytimes in locomotion are the moments of heel-strikes and toe-offs, which are used to time-warp the motions to be blended for feature alignment. These occur only at the start and end frames of every L or R phase, which has already been identified. Thus, it is easy to detect the keytimes. Since every COM peak initiates or ends a motion segment, the peaks can also be considered as keytimes.

4.5. Motion Transition Graph Construction

To reflect the cyclic nature of locomotion, we propose a motion transition graph with two-level hierarchy as shown in Figure 3. The building blocks at the coarse level are locomotive motions and nine transition motions (see Figure 3 (a)), while those at the fine level are motion segments (see Figure 3 (b)).

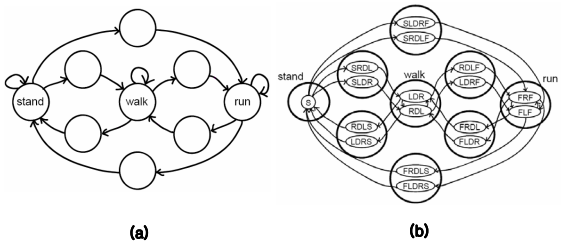

Figure 3: *Motion transition graph*

Let $\mathcal{G} = \big(\mathcal{N}(\mathbf{N}_c, \mathbf{N}_f), \mathcal{A}(\mathbf{A}_c, \mathbf{A}_f)\big)$ denote a motion transition graph, where $(\mathbf{N}_c, \mathbf{A}_c)$ represents the conventional motion transition graph [PSS02, PSS04, KPS03]. The node set $\mathcal{N}$ consists of two type of node sets, that is, the node set $\mathbf{N}_c$ and node set $\mathbf{N}_f$, which represent the building blocks at the coarse level and those at the fine level, respectively. Accordingly, the edge set $\mathcal{A}$ is composed of two types of edge sets, the edge set $\mathbf{A}_c$ and the edge set $\mathbf{A}_f$, which connects their corresponding nodes, respectively.

Our strategy is first to construct the fine-level graph according to $\mathbf{B}$ and $\mathbf{C}$ as defined in Equations 3 and 4, respectively, and then to complete the graph by adding the structure at the coarse level as illustrated in Figure 3 (a).

Guided by Table 1, we initially construct the fine-level graph as shown in Figure 3. Every movement group in $\mathbf{C}$ gives rise to a node in $\mathbf{N}_f$. If there is a pair of motion segments, $\mathbf{m}_p$ and $\mathbf{m}_q$ in $\mathbf{B}$ such that $\mathbf{m}_p \in \mathbf{g}_i$, $\mathbf{m}_q \in \mathbf{g}_j$, and $\mathbf{m}_p || \mathbf{m}_q \in \mathbf{M}$, then $(\mathbf{m}_p, \mathbf{m}_q)$ is mapped onto an edge in $\mathbf{A}_f$ if it is not yet added.

To add the coarse-level structure composed of $\mathbf{N}_c$ and $\mathbf{A}_c$, we first classify the fine-level nodes in $\mathbf{N}_f$ into the coarse-level nodes in $\mathbf{N}_c$ according to Table 1. Then, the coarse-level edges in $\mathbf{A}_c$ are obtained accordingly. Specifically for an ordered pair of coarse-level nodes (g,h), (g,h) is an edge in $\mathbf{A}_c$, if there are a pair of fine-level nodes x and y such that x and y are respectively classified into g and h, and (x,y) is an edge in $\mathbf{A}_f$.

Finally we point out that our graph construction scheme can be easily extended for multiple unlabeled motion sequences since they can be concatenated into a single sequence.

5. Motion Synthesis

In this section, we present an on-line method for locomotion synthesis. Given a stream of motion specifications, the motion transition graph is traversed to synthesize the corresponding sequence of locomotive motions, one by one, in an on-line manner. The coarse-level transition nodes and all fine-level nodes of the graph are transparent for the user.

Thus, a motion specification prescribes only the locomotive motions at coarse-level nodes such as walking, running and standing in terms of motion parameters. The motion specifications for basic movements are derived automatically from the motion context. Given the motion specification for a basic movement, the motion segments assigned to the corresponding fine-level node are blended to generate the basic movement, which is appended to the previously synthesized motion sequence.

5.1. Motion Specification

An coarse-level locomotion node is chosen according to a motion specification, that is, the parameter vector of a locomotive motion (full cycle) $\bar{\mathbf{m}}$,

$$(t(\bar{\mathbf{m}}), v(\bar{\mathbf{m}}), \theta(\bar{\mathbf{m}})), \tag{9}$$

where $t(\bar{\mathbf{m}}), v(\bar{\mathbf{m}})$, and $\theta(\bar{\mathbf{m}})$ denote the motion type, the speed, and the average signed turning angle of $\bar{\mathbf{m}}$, respectively. $\theta(\bar{\mathbf{m}})$ is regarded as the average turning angles for the two half cycles contained in $\bar{\mathbf{m}}$. These are the motion parameters specified manually by the user. We describe how to prescribe the motion specification for a basic movement (half cycle) $\mathbf{m}$, which is a motion segment in $\bar{\mathbf{m}}$. The parameter vector for $\mathbf{m}$ is

$$(t(\bar{\mathbf{m}}), ft(\mathbf{m}), v(\bar{\mathbf{m}}), \theta(\mathbf{m}), a(\mathbf{m})). \tag{10}$$

Here, $ft(\mathbf{m}), \theta(\mathbf{m})$, and $a(\mathbf{m})$ are the foot contact status, the turning angle, and the acceleration of the basic movement to be synthesized. These three parameters are extracted from the motion context while the others are directly obtained from the specification of the motion $\bar{\mathbf{m}}$.

There are three cases depending on the cyclic nature of motions: a standing motion, a cyclic motion such as walking or running, and a transition motion. If a basic movement $\mathbf{m}$ belongs to a standing motion, then the parameters $ft(\mathbf{m}), \theta(\mathbf{m})$, and $a(\mathbf{m})$ are trivially determined, since there are no footsteps, no turning, and no acceleration. We concentrate on the other two cases.

Cyclic Motions: We first describe how to obtain the motion specification for a cyclic motion. The two parameters, $t(\bar{\mathbf{m}})$ and $v(\bar{\mathbf{m}})$ are directly obtained from the motion specification prescribed by the user.

Given a user-provided motion specification for $\bar{\mathbf{m}}$, suppose that a new specification for a basic movement $\mathbf{m}$ is to be derived. Letting $\mathbf{m}_*$ be the basic movement synthesized at the current step, the parameter $a(\mathbf{m})$ of the next basic movement $\mathbf{m}$ is derived as follows:

$$a(\mathbf{m}) = \begin{cases} \bar{a}(\mathbf{m}) & \text{, if } \mathbf{m}_* \text{ is unavailable,} \\ (1-\gamma)a(\mathbf{m}_*) + \gamma\frac{v(\bar{\mathbf{m}})-v(\mathbf{m}_*)}{N(\mathbf{m}_*)} & \text{, otherwise,} \end{cases} \tag{11}$$

where $\bar{a}(\mathbf{m})$ is the average of the acceleration parameters over the motion segments assigned to the next fine-level node specified by $\mathbf{m}$, $N(\mathbf{m}_*)$ is the number of frames in $\mathbf{m}_*$, and γ is a constant that is tuned by the user. $\mathbf{m}_*$ is unavailable whenever the motion type $t(\bar{\mathbf{m}})$ changes. By trial and error, we have found that $\gamma = .5$ works well in our experiments.

To acquire the remaining two parameters, $ft(\mathbf{m})$ and $\theta(\mathbf{m})$, we investigate the structure of a cyclic motion. Such a motion consists of an alternating sequence of half cycles each of which is embodied as a basic movement.

Parameter $ft(\mathbf{m})$ is derived by exploiting this cyclic nature. If the previous basic movement is not available, $ft(\mathbf{m})$ is chosen at random according to a probability distribution of footsteps estimated from the motion segments assigned to current fine-level node. Otherwise, there are two sub-cases depending on $\mathbf{m}_*$. Suppose that $\mathbf{m}_*$ is a basic movement that belongs to a cyclic motion. Then, $ft(\mathbf{m})$ is "left" if $ft(\mathbf{m}_*)$ is "right," and vice versa. Otherwise, $\mathbf{m}_*$ must be a transition movement. Thus, $\mathbf{m}$ and $\mathbf{m}_*$ share the same foot contact status, that is, $ft(\mathbf{m}) = ft(\mathbf{m}_*)$. (See Table 1).

Parameter $\theta(\mathbf{m})$ is also derived based on the observation that the two half cycles shows different turning behaviors. For example, while walking, a human turns clockwise or counter-clockwise depending on the support foot. Accordingly, the signed turning angle oscillates. Let $\mathbf{m}_L$ and $\mathbf{m}_R$ be the half cycles of a motion initiated by the left and right support foot, respectively. Then, the user-specified parameters $v(\bar{\mathbf{m}})$ and $\theta(\bar{\mathbf{m}})$ at the coarse level can be interpreted as

$$v(\bar{\mathbf{m}}) = .5(v(\mathbf{m}_L) + v(\mathbf{m}_R)) \tag{12}$$

$$\theta(\bar{\mathbf{m}}) = .5(\theta(\mathbf{m}_L) + \theta(\mathbf{m}_R)). \tag{13}$$

Empirically, it turns out that the difference, $\theta(\mathbf{m}_L) - \theta(\mathbf{m}_R)$ is highly correlated with $v(\bar{\mathbf{m}})$ and $\theta(\bar{\mathbf{m}})$. By regression analysis, we obtain the linear model

$$\theta(\mathbf{m}_L) - \theta(\mathbf{m}_R) = c_1 v(\bar{\mathbf{m}}) + c_2\theta(\bar{\mathbf{m}}) + c_3. \tag{14}$$

From Equations 13 and 14, we obtain

$$\theta(\mathbf{m}_L) = \theta(\bar{\mathbf{m}}) + .5\,(c_1 v(\bar{\mathbf{m}}) + c_2\theta(\bar{\mathbf{m}}) + c_3) \tag{15}$$

$$\theta(\mathbf{m}_R) = \theta(\bar{\mathbf{m}}) - .5\,(c_1 v(\bar{\mathbf{m}}) + c_2\theta(\bar{\mathbf{m}}) + c_3)\,. \tag{16}$$

Thus,

$$\theta(\mathbf{m}) = \begin{cases} \theta(\bar{\mathbf{m}}) + .5\,(c_1 v(\bar{\mathbf{m}}) + c_2\theta(\bar{\mathbf{m}}) + c_3) & \text{, if } \mathbf{m} = \mathbf{m}_L, \\ \theta(\bar{\mathbf{m}}) - .5\,(c_1 v(\bar{\mathbf{m}}) + c_2\theta(\bar{\mathbf{m}}) + c_3) & \text{, otherwise} \end{cases} \tag{17}$$

Transition Motions: A transition motion is implicitly derived upon receiving a new user-prescribed specification for a locomotive motion $\bar{\mathbf{m}}$ while synthesizing a motion $\mathbf{m}_*$. The parameters for the basic movement $\mathbf{m}$ can be specified in a straightforward manner. The motion type is obtained from Table 1, given $\mathbf{m}_*$ and $\bar{\mathbf{m}}$. If $\mathbf{m}_*$ is not a standing motion, then the other parameters are simply copied from those of $\mathbf{m}_*$. Otherwise, those parameters are generated in a similar manner as for a cyclic motion by ignoring $\mathbf{m}_*$, as though it were unavailable.

An abrupt transition to a new user-specified motion may result in an unnatural motion, which is not physically plausible. We avoid such a motion artifact by incrementally varying the numerical parameters such as speed v and turning angle θ toward those of the new motion specification.

5.2. Motion Blending

We adapt the framework of on-line motion blending in [PSS04, KPS03] to generate a basic movement at an fine-level node, given a parameter vector. Based on multidimensional scattered data interpolation, this framework is composed of four parts: parametrization, weight computation, time-warping, and posture blending.

In parametrization, the motion segments that are assigned to an fine-level node are placed at the points in the parameter space defined by three numerical parameters, speed, turning angle, and acceleration. Given a stream of motion specifications supplied by the user in an on-line manner, they are stored in a queue. Our system dequeues a motion specification one at a time and converts this into a sequence of motion specifications for basic movements as described in Section 4.

Provided with the motion specification of a basic movement, the corresponding fine-level node is visited to compute the contribution of each motion segment stored in this node. We use a scheme of inverse distance weighted interpolation [ACP02, KG04] instead of that of scattered data interpolation based on cardinal basis functions [PSS04, KPS03]. The latter have exhibited quite unstable behavior, since many motion segments at each fine-level node of a cyclic motion have similar parameter vectors. For weight computation, we set $k = 30$ to choose the k-nearest neighbors.

We adopt the incremental time-warping scheme in [PSS04] to align the motion features of the basic movements to be blended. The keytimes have been extracted in Section 4.4. Finally, the time-warped movements are blended using the incremental posture blending scheme [PSS04]. Unlike the original scheme, the root trajectory is also synthesized by blending, based on the registration curve in [KG03].

The synthesized movement is adapted to the environment in an on-line manner to prevent foot sliding and penetration, employing an on-line motion retargeting scheme [SLSG01] after computing the target foot position at each frame by blending the foot positions of the basic movements using the technique given in [PSS04].

5.3. Motion Stitching

Although our motion synthesis scheme produces a basic movement of high quality at every fine-level node, some jerkiness may be observed at knots between adjacent synthesized movements since these are blended separately. The jerkiness mainly arises from the error due to time-warping and motion retargeting. Employing the scheme in [GSKJ03], we take a look-ahead strategy to address this problem: The next movement is computed in the middle of the playback of the current movement so that the two segments are stitched seamlessly at the knot while propagating the error both forward and backward. This guarantees the C^1 continuity for each degree of freedom. Finally, foot skating is cleaned up by using the approach by Kovar et al. [KGS02].

6. Results

We performed experiments on an athlon PC (AMD athlon 64 2GHz processor and 2GB memory). Our human-like articulated figure had 54 DOFs (degrees of freedom): 6DOFs for the pelvis, 6DOFs for the spine, 9DOFs for each limb, 3DOFs for the neck, and 3DOFs for the head. The sampling rate of the example motion data were 60Hz or 120Hz. We first show results for motion modeling and then those for motion synthesis. Finally, we provide some preliminary results for future generalization.

6.1. Motion Modeling

To demonstrate the effectiveness of our motion modeling scheme, we performed experiments on locomotive motions. The results are summarized in Tables 2 and 3. In all experiments, it takes less than one minute to obtain the motion transition graph from an unlabeled motion sequence.

We captured two locomotive motion sequences, referred to as Locomotions A and B. Locomotion A is sampled at a rate of 60Hz and 3.2 minutes long. Locomotion B is at 120Hz and 7.4 minutes long. In capturing these motion sequences, the motion performer was allowed to make any desired combination of locomotive motions while varying speed and turning angle. Both motion sequences are created by concatenating several motion clips, each of which is about one minute long. We applied our motion labeling scheme to those motion sequences and constructed their corresponding motion transition graphs. Since a pair of consecutive motion segments may be from different motion clips, special case is needed in building fine-level edges (see Section 4.5).

Table 2: *Motion labeling results*

	# of motion clips	length	# of motion segments		# of groups
			before refinement	after refinement	
Loco. A	3	3.2 min	353	338	15
Loco. B	7	7.4 min	866	838	17

Table 3: *Motion labeling results*

strings	counts		strings	counts	
	Loco.A	Loco.B		Loco.A	Loco.B
LDR	89	187	LDRS	2	3
RDL	90	194	RDLS	9	2
FLF	48	185	SRDL	3	7
FRF	51	185	SLDR	9	9
S	20	35	SLDRF	0	2
LDRF	2	10	SRDLF	3	2
RDLF	2	4	FRDLS	6	4
FRDL	1	2	FLDRS	0	5
FLDR	3	2	total	338	838

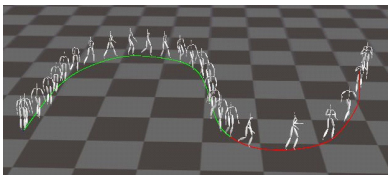

Figure 4: *Pathfollowing*

For locomotion A, we obtained 338 motion segments, which were classified into 15 groups. Every motion segment was mapped onto one of the groups listed in Table 1. However, not every group had a motion segment assigned to it since some basic movements did not appear in the example motion sequence. Thus, the resulting motion transition graph missed two nodes, in particular, the SLDRF and FLDRS nodes. For locomotion B, we obtained similar results as shown in Tables 2 and 3. Unlike Locomotion A, all groups have one or more motion segments assigned to them. In Locomotion B, we observed that the number of walk-to-stand transitions is significantly smaller than that of stand-to-walk, since many motion clips that were concatenated in Locomotion B were initiated by a standing motion and ended by a walking motion.

6.2. Motion Synthesis

To visually verify the validity of our motion modeling scheme, we first performed experiments for locomotive motion synthesis. The motion transition graph is traversed in an on-line manner, guided by a stream of motion specifications.

Our first experiments were conducted to show that our motion synthesis scheme preserves the path-following capability of the scheme in [PSS04]. As illustrated in Figure 4, given a trajectory curve to follow, the speed and turning angle of each motion specification were derived from the tangent vector at the current point sampled from the curve and the angle between the tangent vectors at the previous and current points while moving from one end-point to the other along the curve. The color of a curve segment represents a prescribed motion type. Motion transition occurs whenever color changes.

The next experiments were to show that our motion synthesis scheme also preserves the on-line, real-time capability. For these experiments, we used two types of input devices, an analog joystick and a mouse. For the joystick, the stick position was sampled in an on-line manner to derive the two motion parameters, that is, the speed and turning angle for a desired motion. The motion type was specified with buttons. For the mouse, the cursor position was sampled to compute the speed and turning angle. The motion type was specified with the keyboard. As shown in Figure 5 and also in the accompanying video, our scheme synthesized the prescribed motions on the fly. Even without any code optimization, this scheme produced more than 1000 frames per second.

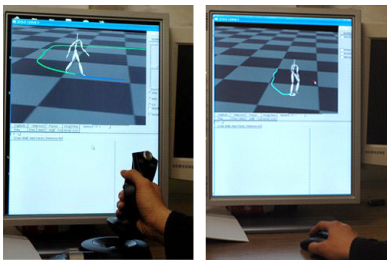

Figure 5: *On-line motion synthesis*

The above two kinds of experiments together with our experience confirmed that the proposed motion synthesis scheme preserves the two important capabilities of the motion synthesis scheme [PSS04]. Furthermore, incorporating acceleration as a parameter, our scheme better reflects some dynamical aspects of locomotion such as leaning, as observed in the accompanying video. From a motion modeling point of view, these results support that our motion labeling scheme automatically produces labeled motion clips, which are comparable to manually-labeled motion clips in quality.

7. Discussion

Relying on footsteps and the COM trajectory of a human-like articulated figure, our current motion modeling scheme is applicable to human locomotive motions. Our scheme is extremely robust to avoid any human adjustment. However, the unlabeled example motion should be captured from a real performer but not created manually by an animator. In the latter case, the example motion may not exhibit the biomechanical observations that our scheme heavily rely on. While focusing on locomotive motions with time-varying speed and turning angle, our scheme cannot handle stylistic variation of motions due to emotional change [RCB98].

For motion synthesis, we adapt the previous framework for on-line motion blending for our hierarchical motion transition graph. Our non-trivial contribution is to derive the parameter vector of a basic movement from the motion context prescribed by the user. To derive the acceleration parameter using Equation 11, we need to tune γ in preprocessing by trial and error. The weight computation can be accelerated employing the K-nearest search algorithm in [AMN*94].

8. Conclusion

In this paper, an example-based, on-line locomotion synthesis method is proposed. We modeled an unlabeled example motion in terms of labeled motion segments called basic movements so that motion blending and motion transition can be performed effectively in accordance with on-line motion specifications by the user. In particular, we proposed a hierarchical motion transition graph to represent a locomotion model. We also addressed the issues in motion dynamics and transition that arise from the cyclic nature of locomotions. The key component of our method is the motion labeling scheme. Exploiting biomechanical observation data on

human COM trajectories and footstep patterns, our scheme decomposes the example motion into groups of motion segments such that the motion segments in the same group share an identical footstep pattern. Moreover, based on string processing rather than numerical computation, our scheme is extremely robust. In the future, we would like to generalize our scheme for other footstep-driven motions such as dancing, boxing, and martial arts.

Acknowledgements

This research was supported by the Korea Science and Engineering Foundation (KOSEF R01-2004-000-10413-0).

References

[ACP02] ALLEN B., CURLESS B., POPOVIC Z.: Articulated body deformation from range scan data. *ACM Transactions on Graphics 21*, 3 (July 2002), 612–619.

[AF02] ARIKAN O., FORSYTH D. A.: Interactive Motion Generation from Examples. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002) 21*, 3 (July 2002), 483–490.

[AFO03] ARIKAN O., FORSYTH D. A., O'BRIEN J. F.: Motion synthesis from annotations. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003) 22*, 3 (July 2003), 402–408.

[AMN*94] ARYA S., MOUNT D. M., NETANYAHU N. S., SILVERMAN R., WU A.: An optimal algorithm for approximate nearest neighbor searching. In *Proceedings of 5th ACM-SIAM Sympos. Discrete Algorithms* (1994), pp. 573–582.

[BB98] BINDIGANAVALE R., BADLER N. I.: Motion abstraction and mapping with spatial constraints. In *Proceedings of International Workshop CAPTECH'98* (1998), pp. 70–82.

[BSP*04] BARBIC J., SAFONOVA A., PAN J., FALOUTSOS C., HODGINS J., POLLARD N.: Segmenting motion capture data into distinct behaviors. In *In the Proc. of Graphics Interface* (2004), pp. 185–194.

[FMJ02] FOD A., MATARIC M. J., JENKINS O.: Automated derivation of primitives for movement classification. *Autonomous Robots 12*, 1 (2002), 39–54.

[GJH00] GALATA A., JOHNSON N., HOGG D.: Learning variable-length markov modles of behavior. *Computer Vision and Image Understanding 81* (2000), 398–413.

[GR96] 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* (Aug. 1996), pp. 95–107.

[GSKJ03] GLEICHER M., SHIN H. J., KOVAR L., JEPSEN A.: Snap-together motion: Assembling run-time animation. *ACM Transactions on Graphics 22*, 3 (July 2003), 702–702.

[JM03] JENKINS O., MATARIC M.: Automated derivation of behavior vocaburaries for autonomous humanoid motion. In *In Proc. of AAMAS'03* (2003), pp. 225–232.

[KG03] KOVAR L., GLEICHER M.: Flexible automatic motion blending with registration curves. In *Eurographics/SIGGRAPH Symposium on Computer Animation* (2003), Breen D., Lin M., (Eds.), Eurographics Association, pp. 214–224.

[KG04] KOVAR L., GLEICHER M.: Automated extraction and parameterization of motions in large data sets. vol. 23, pp. 559–568.

[KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion Graphs. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002) 21*, 3 (July 2002), 473–482.

[KGS02] KOVAR L., GLEICHER M., SCHREINER J.: Footskate cleanup for motion capture editing. In *Proceedings of ACM SIGGRAPH Symposium on Computer Animation* (July 2002).

[KPS03] KIM T. H., PARK S. I., SHIN S. Y.: Rhythmic-motion synthesis based on motion-beat analysis. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003) 22*, 3 (July 2003), 392–401.

[LCR*02] 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 (Proc. SIGGRAPH 2002) 21*, 3 (July 2002), 491–500.

[LP02] LIU C. K., POPOVIC Z.: Synthesis of complex dynamic character motion from simple animations. In *SIGGRAPH 2002 Conference Proceedings* (2002), Hughes J., (Ed.), Annual Conference Series, ACM Press/ACM SIGGRAPH, pp. 408–416.

[LWS02] LI Y., WANG T., SHUM H.: Motion Texture: A Two-Level Statistical Model for Character Motion Synthesis. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002) 21*, 3 (July 2002), 465–472.

[PB02] PULLEN K., BREGLER C.: Motion Capture Assisted Animation: Texturing and Synthesis. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002) 21*, 3 (July 2002), 501–508.

[Per92] PERRY J.: *Gait analysis: Normal and Pathological Function*. Delmar Learning, 1992.

[PSS02] PARK S. I., SHIN H. J., SHIN S. Y.: On-line locomotion generation based on motion blending. In *Proceedings of ACM SIGGRAPH Symposium on Computer Animation* (July 2002), pp. 105–111.

[PSS04] PARK S. I., SHIN H. J., SHIN S. Y.: On-line motion blending for real-time locomotion generation. *Computer Animation and Virtual Worlds 15* (Sept. 2004), 125–138.

[RCB98] ROSE C., COHEN M. F., BODENHEIMER B.: Verbs and adverbs: Mulidimensional motion interpolation. *IEEE Computer Graphics and Applications 18*, 5 (Sept. 1998), 32–40.

[SLSG01] SHIN H. J., LEE J., SHIN S. Y., GLEICHER M.: Computer puppetry: An importance-based approach. *ACM Trans. Graph. 20*, 2 (2001), 67–94.

[SP05] SULEJMANPAŠIĆ; A., POPOVIĆ J.: Adaptation of performed ballistic motion. *ACM Trans. Graph. 24*, 1 (2005), 165–179.

[SRC01] SLOAN P., ROSE C. F., COHEN M. F.: Shape by example. In *Proceedings of 2001 ACM Symposium on Interactive 3D Graphics* (2001), pp. 135–144.

[TH00] TANCO L. M., HILTON A.: Realistic synthesis of novel human movements from a database of motion capture examples. In *Proceedings of the IEEE Workshop on Human Motion* (2000), pp. 137–142.

[WH97] WILEY D. J., HAHN J. K.: Interpolation synthesis for articulated fiture motion. *IEEE Computer Graphis and Applications 17*, 6 (1997), 39–45.

[Win90] WINTER D. A.: *Biomechanics and Motor Control of Human Movement*. John Wiley and Sons Inc, 1990.