Synthesizing New Walk and Climb Motions from a Single Motion Captured Walk Sequence

Shrinath Shanbhag
CDAC Mumbai
shrinath@ncst.ernet.in

Sharat Chandran
IIT Bombay
sharat@cse.iitb.ac.in

Abstract

We describe a method to dynamically synthesize believable variable stride and variable foot lift walk and climb motions from a single motion captured walk sequence. Our method is driven by motion capture and guided by a simple kinematic walk model . The method allows control in the form of stride and lift parameters. It generates a range of variations while maintaining individualistic nuances of the captured performance.

1. Introduction

Walking is a fundamental motion for us humans. The number of variations of *"the walk"* an individual is capable of performing is potentially infinite. Even so, animated virtual characters in interactive applications such as games, are driven by a severely limited repertoire of walk sequences. Usually just two sequences, one for run and one for walk. As a result we see walk animation that does not adapt to the environment and is visually jerky. Moreover, climbs are many a times synthesised as walk or run sliding along an incline.

Most current games use motion capture (mocap) driven character animation. Using mocap allows games to easily capture and playback any motion that an actor can perform. This is a key property missing from key-framing and physically based simulation systems. However mocap based systems possess one significant drawback - the lack of dynamic control. Such control is inherent in procedural methods. Such control is not essential in certain applications, such as in animation catering to movies. However applications where the playback environment is different from the capture environment suffer from its absence. In addition, interactive applications like games demand dynamic control and adaptation facilities.

Recently, games have started employing hybrid schemes that combine motion capture and physically based simulation, for animation. For example *rag-doll* dynamics simulations is often employed to simulate falling and bouncing in real time. However, these schemes treat objects as passive and totally governed by Newton's laws of motion. This is not the case for a virtual humanoid character, that is active. By active we mean, it can initiate motion on its own. Most attempts at physically synthesizing motion have succeeded in generating only simple motions. It is still difficult to create such controllers for simulating a wide variety of human motion. Hence the dependence on pre-created motion.

Rapid increase in processing power and storage capacities in modern machines has opened up a new avenue. Applications can employ larger databases of motion captured sequences. Schemes for organizing such databases as graphs and synthesizing motion on demand are now available. However, these schemes are more adept at generating high-level motion specifications. Adapting captured motion to a new environment, demands low level control. Moreover many such schemes still need human supervision.

In this paper, we describe a new method that uses a simple kinematic walk model to derive variable stride and variable lift walk and climb limb motion from a single motion captured walk sequence. The synthesis can be controlled programatically. The next section describes related work in this area. The following sections then describe our simple kinematic walk model, our method for synthesizing new motion and our ideas for future work.

2. Related Work

The history of research in humanoid animation dates back more than 15 years. Motion synthesis methods can be broadly classified as kinematics based, dynamics based and constraint based methods. In addition there are also hybrid methods which mix one or more of the other techniques. More recently, methods based on motion captured sequences have been developed.

2.1. Gait Synthesis Techniques

Multon et al [15] provide an excellent survey of computer animation of human walking. Metaxas and Sun [20] describe a low-level gait generator based on sagittal elevation angles, which allows curved locomotion to be created easily. They also describe an inverse motion mapping algorithm, that allows motion to be adapted to uneven terrains.

In addition they describe a higher level control frame work that allows motion requirements to be specified at a high level by sketching the desired path. Hodgins et. al [8] describe an algorithm that allows simulation of running, bicycling and vaulting. The simulation is achieved through control algorithms that cause physically realistic models to perform the desired behaviour. Faloustos et al [4] describe a method to combine various physically based simulation controllers into a unified framework.

2.2. Mocap based Motion Editing Techniques

Researchers working in this field have proposed a number of innovative techniques adapting signal processing methods or employing constraint based solvers. Bruderlin et al [3] have successfully applied techniques from image and signal processing domains to designing, modifying and adapting animated motions. Unuma et al [21] describe a method for modeling human figure locomotions with emotions. Herein Fourier expansions of experimental data of actual human behaviors serve as a basis from which to interpolate or extrapolate the human locomotions. Witkin et al [22] describe a simple technique for editing captured animation based on warping of the motion parameter curves. Gleicher et. al [6][5]use space-time constraint formulations to modify captured motion and retargeting motion to new characters with different segment lengths. Gleicher [6] provides a comparison of constraint based motion editing methods. Lee et al [14] describe a hierarchical framework for adapting existing motion of humanoids to externally specified constraints. Gleicher et. al [19] describe a method that takes into consideration physical principles to touch up synthetically generated motion, so as to make it more plausible.

2.3. Mocap Based Synthesis Techniques

Motion synthesis techniques create new motion from existing motion data. Gleicher et al [11], Lee et al [13], Forsyth et al [1] describe techniques to create new motion sequences from a corpus of motion data. Each technique essentially clusters similar motion into nodes. The next phase builds a graph of nodes, where each edge represents a transition between nodes. A walk through the cluster node graph results in synthesis of new motion sequences. The techniques differ in metrics used for clustering, pruning schemes and control criteria for node walk. Bregler et al [17],[18] describe a scheme for synthesizing missing degrees of freedom and adding details to specified degrees of freedom, for a roughly specified motion sequence. Their method uses the various correlation between the various degrees of freedom (DOF's) within each motion sequences. Forsyth et. al [2] describe a technique using a novel search method based around dynamic programming to interactively synthesize motion from annotations. The user paints a timeline with annotations and the system assembles the motion sequence by extracting suitable frames from a motion database. Taehoon Kim et al [10] describe a scheme that allows synthesis of rhythmic motion driven by sound beats. Gleicher et al [7] present a technique that preprocesses a corpus of motion capture examples into a set of short clips that can be concatenated to make continuous streams of motion. The resulting simple graph structure can be used in virtual environments where control and responsiveness are more important than accuracy.

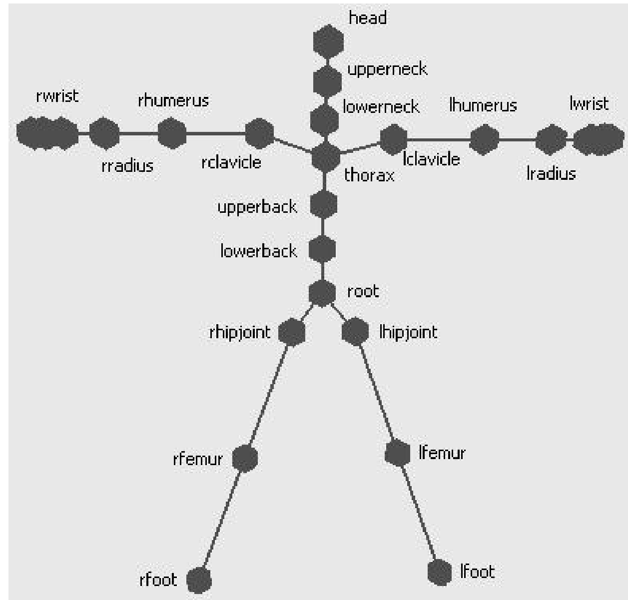

Figure 1: Skeletal heirarchy

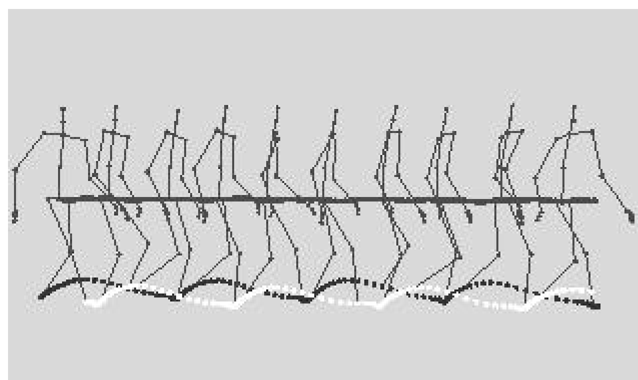

Figure 2: Base motion captured walk sequence

3. Our method

A skeletal hierarchy, such as in Figure 1, is used to acquire motion capture data. Motion data consists of a bundle of motion signals. Each signal represents a sequence of sampled values for each degree of freedom (DOF). Motion signals are sampled at discrete instances of time with a uniform sampling interval to yield a motion clip. In each frame, the sampled values of the different DOFs at each joint determine the configuration of an articulated figure for that frame. Often the root of the skeletal hierarchy contains six DOFs (three for translation and three for rotation about x, y and z axis), whereas rest of the nodes contain three DOFs (only rotation). Scaling is mostly unused.

Our base motion sequence is a motion captured straight line walk, shown in Figure 2. The aim is to create believable motion as against simulating physically accurate behaviour. We preprocess this sequence to obtain relative displacement vectors of joints at each frame. We account for inherent foot plant constraints. We identify all frames that contain either the right or the left or both feet planted. We annotate these frames accordingly. For each frame we also compute a lift vector for each foot.

3.1. Computing Relative Displacement Vector

We compute per frame relative displacement vectors for the *root node*, *left foot node* and *right foot node*. The displacement vector is computed as follows:

Let P_{j_i} and $P_{j_{(i+1)}}$ be the world positions of joint j at frame i and $i+1$ respectively. Then the relative displacement vector D_{j_i} for joint j at frame i is given by:

$$D_{j_i} \leftarrow P_{j_{(i+1)}} - P_{j_i} \tag{1}$$

3.2. Identifying Foot Plant Frames

We identify foot plant constraints as follows. We identify frames with zero crossings for vertical displacement (the y axis in our case). We select all frames which are close to the ground, within a given threshold. This forms the seed set of foot plant frames. For most normal walk sequences, we observe that the foot is placed on the ground for more than one frame. However, in a motion captured sequence, the foot positions may not coincide exactly due to foot skate. [12] describe a technique to identify and correct foot skate. We use a simpler method. From the initial set of foot plant frames obtained above, we sequentially search in either direction and cluster frames, near the seed foot plant frame, where the magnitude of the displacement vector is below a given threshold value. We stop the search at the first frame that fails the test. We then cluster together, like foot plant frames based on their sequence in the clip.

3.3. Computing Lift Vector

Computing the lift vector for the base sequence, involves projecting the feet position on the ground plane, and computing the vector difference of the two positions.Let $P_{j\,i}$ and

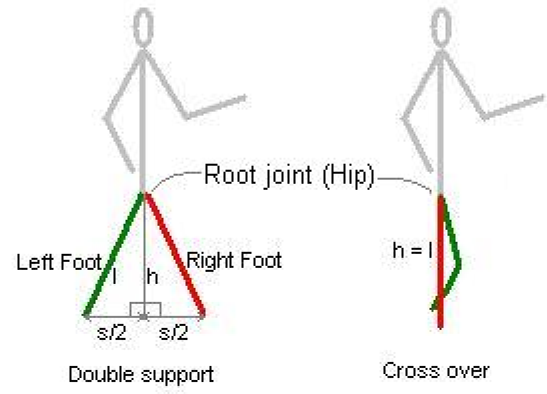

Figure 3: Simple Kinematic Walk Model

$P'_{j\,i}$ be the world positions of joint j, representing a foot, at frame i and its projection on the ground plane respectively. Then the lift vector $L_{j\,i}$ for joint j at frame i is given by:

$$L_{j\,i} \leftarrow P_{j\,i} - P'_{j\,i} \tag{2}$$

Traversing through the frames sequentially, we identify frames with local maximas of the foot lift vector. We find the mean magnitude, L_m, of magnitudes of local maxima of foot lift vectors. This is used to determine the foot lift scaling factor as explained in Section 5.2.

4. Kinematic Model for Walk

Human walking is a process of locomotion in which the erect, moving body is supported by first one leg and then the other. As the moving body passes over the supporting leg, the other leg is swinging forward in preparation for its next support phase. One foot or the other is always on the ground, and during that period when the support of the body is transferred from the trailing to the leading leg there is a brief period when both feet are on the ground. [9] describes human walking in more detail.

We use a simple kinematic model for human walking that allows us to estimate root-ground clearance during the gait cycle. Figure 3 shows ground clearance values for the root joint. The minimum occurs during double support stance. The maximum occurs during the crossover stance. From the figure 3 we have

$$h = \begin{cases} \sqrt{l^2 - (\frac{s}{2})^2} & \text{for double step stance} \\ l & \text{for crossover stance} \end{cases} \tag{3}$$

where s is the stride and l is the length of the legs measured from hip to heel.

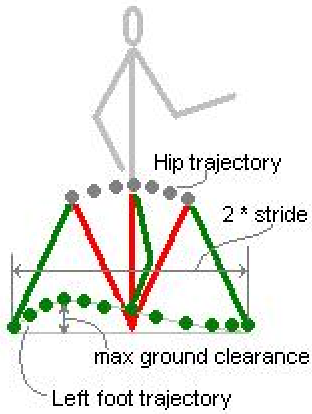

Figure 4: Root and Foot trajectories in the saggital plane

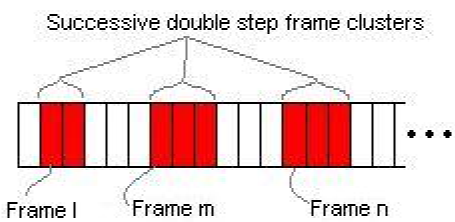

Figure 5: Frame m and Frame n

The root joint, trajectory between two double step for the saggital plane (X-Y plane in our case) is a sinusoidal wave [9]. If R_i is the position of the root joint for the i^{th} frame, m is the first frame of a double step cluster and n is the first frame of the immediately succeeding double step cluster as in Figure 5, then the y position of the root joint is given by:

$$R_i.y = h_{min} + \delta h * |sin(\theta)| \tag{4}$$

$$\text{where} \begin{cases} \theta = \frac{\pi*(i-m)}{m-n}, m \leq i \leq n \\ h_{min} = \sqrt{l^2 - (\frac{s}{2})^2} \\ \delta h = l - h_{min} \end{cases}$$

The trajectory in the transverse plane (Z-X plane in our case) is obtained from the captured sequence.

The feet trajectory is obtained directly from the motion capture sequence. Figure 4 shows the trajectories for the root joint and the left foot joint, for a half walk cycle. In the second cycle, the root joint trajectory repeats itself and the right foot follows a trajectory similar to the one shown for left foot.

5. Synthesis

In this section we describe our synthesis of variable stride, variable foot lift and climb. Our synthesis is driven by the base sequence and guided by our simple kinematic walk model. We synthesize motion only for the lower body and playback the upperbody animation as originally recorded.

5.1. Varying stride

We define a resultant foot stride vector S, which is computed by adding individual frame displacement vectors of frames for either left or right foot joint from frame l to frame n, where l and n are as explained in Figure 5. The magnitude of S gives twice the stride. The stride for left and right feet in a rythmic walk motion are equal. The stride of the captured walk sequence is varied by directly scaling the relative displacement vectors D_{j_i}, with a scaling factor s. The displacement vectors of the root joint are scaled by the same scaling factor. The trajectory of the root joint in the saggital plane is obtained from equation 4.

$$S = \sum_{i=l}^{n} D_{j_i} \tag{5}$$

$$s = \frac{\textit{2*(new stride)}}{|S|} \tag{6}$$

A new trajectory is obtained for the root joint and foot joints as follows:

for $i = l$ to $(n-1)$
$P_{j_{(i+1)}} \leftarrow P_{j_i} + s * D_{j_i}$

Smoothly varying the scale factor s creates a smoothly varying stride. This may be used to synthesize a accelerating or retarding walk sequence. The DOF angles for each joint in the foot kinematic chain are recomputed, for each frame, by a two link analytical *inverse kinematics solver*[16].

5.2. Varying Foot Lift

We vary foot lift by computing new positions for the feet joints by scaling the frame foot lift vectors by a scale factor l. The scale factor is obtained as:

$$l = \frac{\textit{new lift}}{L_m} \tag{7}$$

where L_m is the mean magnitude of local maxima of foot lift vectors as described in Section 3.3.

For a motion sequence containing n frames, the new foot trajectory is computed as

for $i = 0$ to $(n-1)$
$P_{j_i} \leftarrow P'_{j_i} + l * L_{j_i}$

where P'_{j_i} is the projection of joint position P_{j_i} on the ground plane.

The computation for foot lift and stride are combined as follows

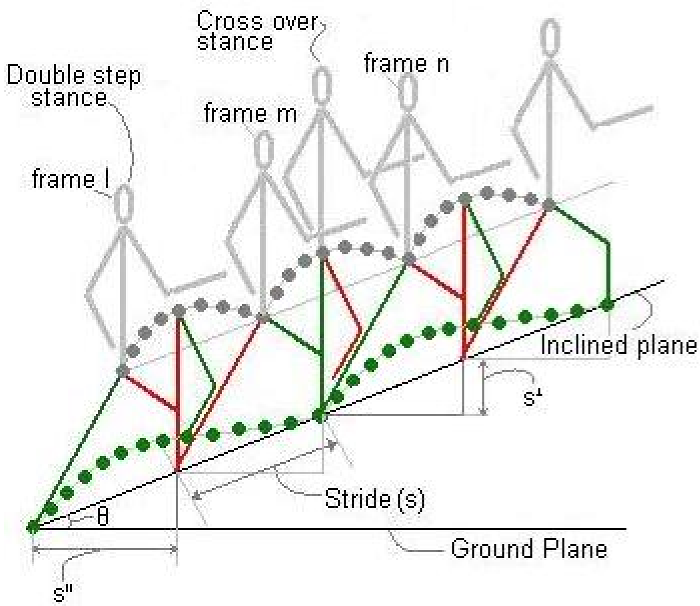

Figure 6: Modelling climb

for $i = l$ to $(n-1)$
$$P_{j_{(i+1)}} \leftarrow P_{j_i} + s * D_{j_i}$$
$$P_{j_{(i+1)}} \leftarrow P'_{j_{(i+1)}} + l * L_{j_{(i+1)}}$$

The DOF angles for each joint in the foot kinematic chain are recomputed, for each frame, using the IK solver.

5.3. Synthesizing climb

Here we describe the synthesis of climb motion, along a plane which makes an angle θ with the horizontal ground plane, as illustrated in Figure 6. Let s be the desired stride along the inclined plane. The distance parallel to the ground plane is given by $s^{\|}$. The corresponding rise in height in ground level is given by $s^{\perp}$.

$$s^{\|} = s * cos(\theta)$$
$$s^{\perp} = s * sin(\theta) \quad (8)$$

For synthesizing the climb motion, we need to estimate the path of the root joint and the feet joints of our articulated body. For this we use the simple kinematic model described in section 4. We first synthesize a sequence with stride scaled to $s^{\|}$. We use this modified sequence for further modifications as described later.

Consider successive double step frames as in frames l, m, n shown in Figure 5. The geometry of these frames is as visualized in Figure 6. To compute the trajectory of the root joint, note that the root undergoes an additional vertical displacement of magnitude $s^{\perp}$ between each successive double step frame. The trajectory of the root from frame l to m is computed by first displacing the root position at m in the perpendicular direction by $s^{\perp}$. The per frame displacement for in between frames is linearly interpolated as follows:

for $i = l$ to m
$$P_{j_i}.y \leftarrow P_{j_i}.y + \frac{(i-l)*s^{\perp}}{(m-l)}$$

where P_{j_i} is the position of the root joint in the i^{th} frame of the modified sequence.

Trajectory for the left and right foot joints can be calculated in a similar manner. Note that the left foot, shown in green color in Figure 6, accumulates an additonal vertical displacement of magnitude $2 * s^{\perp}$ between frames l and m. For frames m to n, the right leg, shown in red, accumulates this vertical displacement. The legs climb alternately, with the planted leg, maintaining its height. The trajectory of the left leg joint from frames l to m can be computed as follows:

for $i = l$ to m
$$P_{j_i}.y \leftarrow P_{j_i}.y + \frac{(i-l)*2*s^{\perp}}{(m-l)}$$

where P_{j_i} is the position of the left leg joint in the i^{th} frame of the modified sequence. The DOF angles for each joint in the foot kinematic chain are recomputed, for each frame, using our IK solver. The upper body posture for the climb, synthesized by our method, needs further adaptation to look real. [19] describe a method which can be directly applied as a post process to our synthesis.

6. Conclusion and Future Work

We have described the synthesis of rythmic walk along a straight line. In real life, people walk along curvilinear paths, sometimes with continously varying gradients. Relative motion of an articulated body, is invariant to global rotation around the vertical axis of lateral symmetry [11][13]. Therefore walking along a level curved path with small to medium curvature can be approximated with multiple linear segments with appropriate rotation around the Y axis, with the root joint as the origin. For an inclined curved path, θ changes continuously and therefore $s^{\|}$ and $s^{\perp}$ need to be recomputed for each step in the approximation. Climbing a uniformly spaced staircase is equivalent to climbing an inclined plane. Motion synthesis for climbing down an inclined plane can be constructed similarly. In future we plan to address the problem of dynamic motion planning and adaptation.

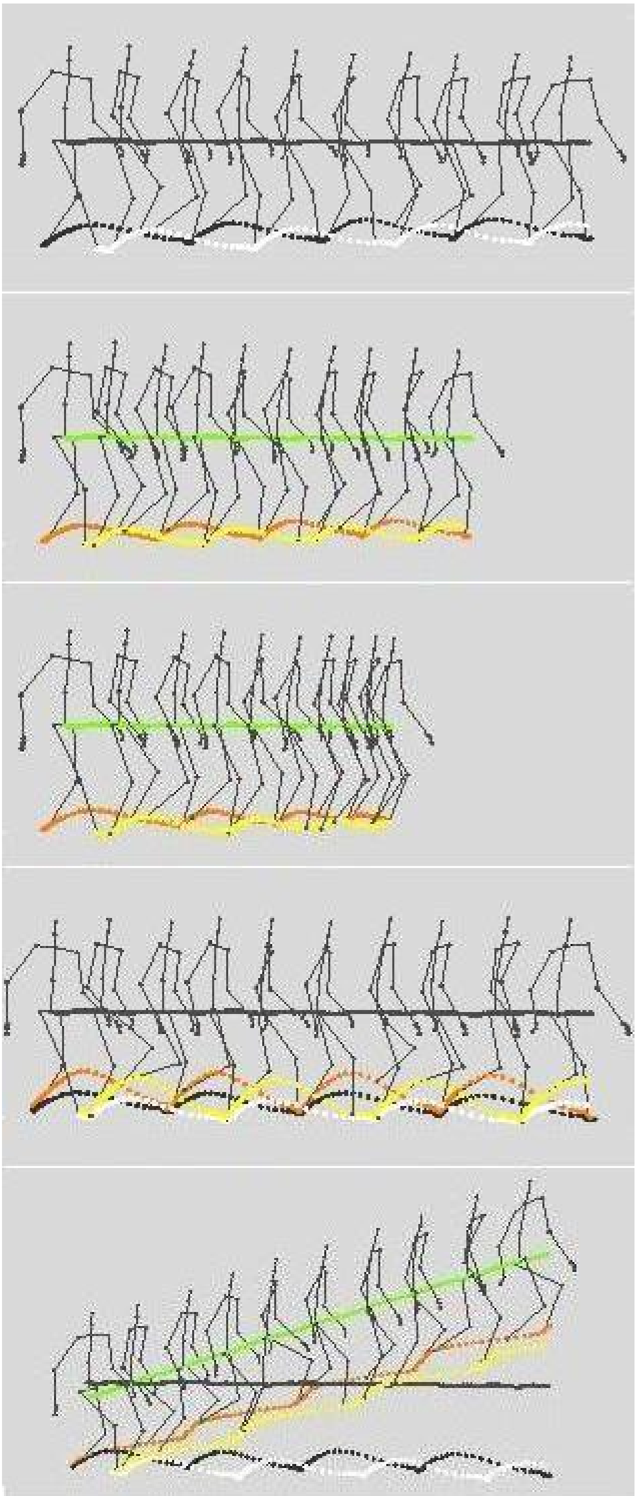

Figure 7: Synthesized walk: From top to bottom - base sequence, uniformly scaled stride, continuously varying stride, scaled foot lift with base sequence trajectory superimposed, climb with base sequence trajectory superimposed

References

[1] O. Arikan and D. A. Forsyth. Interactive Motion Generation from Examples. In *Proc. of Siggraph '02*, pages 483 – 490, 2002.

[2] O. Arikan, D. A. Forsyth, and J. F. O'Brien. Motion Synthesis from Annotations. In *Proc. of Siggraph '03*, pages 483 – 490, 2003.

[3] A. Bruderlin and L. Williams. Motion Signal Processing. In *Proc. of Siggraph '95*, 1995.

[4] P. Faloutsos, M. van de Panne, and D. Terzopoulos. Composable controllers for physics-based character animation. In *Proceedings of Siggraph '01*, August 2001.

[5] M. Gleicher. Retargetting Motion to New Characters. In *Proc. of Siggraph '98*, 1998.

[6] M. Gleicher. Comparing Constraint-based Motion Editing Methods. *Graphical Model*, 63(2):107 – 134, 2001.

[7] M. Gleicher, H. J. Shin, L. Kovar, and A. Jespen. Snap together motion: Assembling run-time animation. In *2003 Symposium on Interactive 3D Graphics*, April 2003.

[8] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating human athletics. In *Proceedings of Siggraph '95*, 1995.

[9] V. T. Inman, H. Ralston, and F. Todd. *Human Walking*. Lippincott Williams & Wilkins, 1989.

[10] T. Kim, S. I. Park, and S. Y. Shin. Rythmic-motion synthesis based on motion-beat analysis. In *Proc. of Siggraph '03*, 2003.

[11] L. Kovar, M. Gleicher, and F. Pighin. Motion Grahs. In *Proc. of Siggraph '02*, 2002.

[12] L. Kovar, J. Schreiner, and M. Gleicher. Footskate cleanup for motion capture editing. In *Proceedings of the 2002 ACM Symposium on Computer Animation (SCA)*, July 2002.

[13] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard. Interactive Control of Avatars Animated with Human Motion Data. In *Proc. of Siggraph '02*, 2002.

[14] J. Lee and S. Y. Shin. A Hierarchial Approach to Interactive Motion Editing for Human like Figures. In *Proc. of Siggraph '99*, 1999.

[15] F. Multon, L. France, M.-P. Cani-Gascuel, and G. Debunne. Computer animaton of human walking: A survey. Technical Report 3441, INRIA, June 1988.

[16] K. Perlin. Two link inverse kinematics. GDC 2002 course notes `http://mrl.nyu.edu/p̃erlin/gdc/ik/talk.html`, March 2002.

[17] K. Pullen and C. Bregler. Animating by Multi-level Sampling. In *Proc. of IEEE Computer Animation 2000*, 2000.

[18] K. Pullen and C. Bregler. Motion Capture Assisted Animation: Texturing and Synthesis. In *Proc. of Siggraph '02*, 2002.

[19] H. J. Shin, L. Kovar, and M. Gleicher. Physical touch-up of human motion. In *Pacific Graphics 2003*, October 2003.

[20] H. C. Sun and D. N. Metaxas. Automating gait generation. In *Proceedings of Siggraph '01*, August 2001.

[21] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion based human figure animation. In *Proceedings of Siggraph '95*, 1995.

[22] A. Witkin and Z. Popovic. Motion Warping. In *Proc. of Siggraph '95*, 1995.