Real-Time Character Control for Wrestling Games

Edmond S.L. Ho and Taku Komura

Institute of Perception, Action and Behaviour
School of Informatics, University of Edinburgh

Abstract. This paper proposes a method to simulate the real-time interactions of tangling motions by two virtual wrestlers in 3D computer games. The characters are controlled individually by two different players - one player controls the attacker and the other controls the defender. We make use of the topology coordinates which are effective to synthesize tangling movements. The attacker's movements are simulated by changing the topology coordinates at every frame, and the defender is controlled to escape from such an attack by inverse kinematics. The experimental results show the methodology can simulate realistic competitive interactions of wrestling in real-time, which is difficult by previous methods.

Keywords: character animation, motion capture.

1 Introduction

Wrestling is a major field in 3D computer games. The motions in wrestling involve close contacts between the characters such as squeezing and locking. Simulating such motions is not easy as they require a great amount of close contacts and collision avoidance. In fact, in most of the wrestling games, such motions are designed carefully in advance by the animators and the game players have little control over the characters once complex tangling motions are started. This is completely different from real wrestling - wrestlers have lots of degrees of freedom to escape from the attackers, and attackers need to carefully select their motion to lock the defender. Such complex interactions of the wrestlers are rarely simulated in the existing wrestling games.

In this paper, we make use of the topology coordinates [5] in order to simulate such complex interactions in real-time. The attacker is controlled so that its topology coordinates approach to those of the target configuration. The player can switch the attacks according to the configuration of the attacker and the defender. On the other hand, the defender can escape from such attacks by kinematically controlling the body.

Our interface provides large degrees of freedom to the game players while minimizing the complexity of control, which can increase the attractiveness of wrestling games.

2 Related Work

We first briefly review the recent wrestling games and their interfaces. Next, we review researches of two topics that are related to character control in wrestling games - real-time character control and close interactions of multiple characters.

A. Egges, R. Geraerts, and M. Overmars (Eds.): MIG 2009, LNCS 5884, pp. 128–137, 2009.

Wrestling games have been attracting millions of game players around the world and has become one of the major categories in computer games. In the old games, the attacks were limited to hits such as punches, kicks or chops. Therefore, the users could only repeatedly press the buttons to give large damage to the opponent character.

The recent advanced games allow the characters to conduct complex tangling attacks such as back drops, rear-choke hold and full nelson attacks. In order to launch such motions, the user is supposed to press the button at the correct pose and timing, or select the part to attack by the pointing device [1]. Motions that involve tangling are usually just replayed during run-time as real-time editing of such motions can easily result in collision and penetration of the body segments. The attractiveness of the games will be greatly enhanced if the game players have access to the details of the tangling motions during the game play.

Real-time character control: Here we review a number of techniques which are useful for real-time control of the virtual wrestlers. We first review techniques of inverse kinematics (IK) which is a basic technique to edit character motions, and then their extensions to handle dynamics and tangling motions.

Inverse kinematics (IK) [20,13,14,23,21] is a basic technique that is often used for real-time control of characters. Methods to control characters of arbitrary morphology [7,3] have also been proposed. IK methods can be divided into (1) CCD-based approaches [9], (2) analytical approaches [11], (3) particle-based approaches [7,3] and (4) quadratic programming based approaches [20,13,21].

Among these approaches, the quadratic programming based approach has an advantage to simulate wrestling motions as it can enclose constraints based on dynamics [22,8,16] and topological relationships [5] into the solver. Our approach is built on top of Ho and Komura [5], which propose to handle tangling motions by adding constraints into an optimization-based IK solver.

Multiple Character Interactions: The simulation of interactions between multiple characters has many applications such as computer games, virtual environments and films. Liu et al.[12] simulate the close dense interactions of two characters by repetitively updating the motion of each character by spacetime constraints. Lee and Lee [10] simulate the boxing match by using reinforcement learning. Treuille et al [19] also use reinforcement learning to simulate the pedestrians avoiding each other. Shum et al [17] use min-max search to find the optimal action in a competitive environment. They also propose a real-time approach based on an automatically produced finite state machine [18]. These researches do not handle very close interactions such as holding or wrestling. Ho and Komura [4] generate wrestling motions by finding the topological relationship of characters from the template postures and use PD control to simulate movements where the topological relationship is kept the same. If we want to simulate a scene where the topological relationship of characters changes in time, we cannot apply such a method. A method to dynamically update the postures and the topological relationships is required. [6] propose to evaluate the similarity of character postures based on the topological relationships. When equivalent postures are found, the postures are linearly interpolated at the level of generalized coordinates. However, no method has enabled game players to interactively change the topological relationship of virtual characters in real-time. We propose such an approach in this paper.

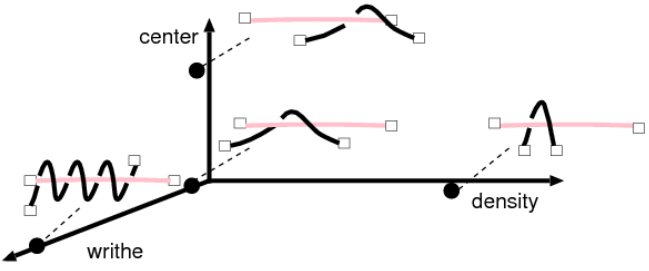


Fig. 1. The three axes in topology space : writhe, center and density. The center, which specifies the central location of the twist relative to each strand, is actually composed of two scalar parameters, although it is represented by a single axis in this figure. The density tells which strand is playing the major role to compose the twist.

3 Methodology

We first briefly review the topology coordinates [5], which is the basic technique used to control the tangling motions. Next, we explain how the topology coordinates can be applied to the control of virtual wrestlers in computer games.

3.1 Topology Coordinates

Topology coordinates enable characters to tangle their bodies with other characters without conducting global path planning methods. The topology coordinates are composed of three attributes, the writhe, center and density. The first attribute **writhe** counts how much the two curves are twisting around each other. Writhe can be calculated by using Gauss Linking Integral (GLI) [15] by integrating along the two curves γ_1 and γ_2 as:

$$GLI(\gamma_1, \gamma_2) = \frac{1}{4\pi} \int_{\gamma_1} \int_{\gamma_2} \frac{d\gamma_1 \times d\gamma_2 \cdot (\gamma_1 - \gamma_2)}{\|\gamma_1 - \gamma_2\|^3} \tag{1}$$

where $\times$ and $\cdot$ are cross product and dot product operators, respectively. The GLI computes the average number of crossings when viewing the tangle from all directions.

Curves can twist around each other in various ways. In order to further specify the status of the two chains, we introduce the other two attributes, **center** and **density**. Examples of changing these attributes for a pair of strands are shown in Figure 1. The center, which is composed of two scalar parameters, explains the center location of the twisted area, relative to each strand. The density, which is a single scalar parameter, explains how much the twisted area is concentrated at one location along the strands. When the density is zero, the twist is spread out all over the two strands. When the density value is either very large or very small, we can say one strand is playing a major role to compose the twist, as it is twisting around the other strand which is kept relatively straight (Figure 1). When the density turns from negative to positive, or vice versa, the strand playing the major role switches.

3.2 Controlling the Characters

We represent the bone structure of the virtual wrestlers by a set of line segments. Therefore, now we will mathematically define the topology coordinates of serial chains. Let us assume we have two chains S_1 and S_2, each composed of n_1 and n_2 line segments, connected by revolute, universal or gimbal joints (Figure 2). In this case, we can compute the total writhe by summing the writhes by each pair of segments:

$$w = GLI(S_1, S_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} T_{i,j} \tag{2}$$

where w represents the writhe, $T_{i,j}$ is the writhe between segment i on $S1$ and j on $S2$. Let us define a $n_1 \times n_2$ matrix $\mathbf{T}$ whose (i, j)-th element is $T_{i,j}$, and call this the **writhe matrix**. The writhe matrix explains how much each pair of segments from S_1 and S_2 contributes to the total writhe value. Various twists of two serial chains and the corresponding writhe matrices are shown in Figure 3.

The topology coordinates can be updated by changing the distribution of the elements in the writhe matrix using basic operations such as rotation, translation and scaling. Rotating the elements results in changing the density. Translating the elements results in changing the center. Scaling the whole matrix results in changing the writhe. Let us define these operations by $R(M,d)$, $Tr(M,c)$ and $S(M,w)$, respectively, where M is the input matrix and (d,c,w) are topology coordinates, each of which representing the density, center and writhe, respectively.

Rather than directly manipulating the writhe matrix of the characters, we first compute an ideal, desired writhe matrix and try to minimize the difference of the character's writhe matrix and the desired writhe matrix. The desired writhe matrix T_d that corresponds to topology coordinates (d,c,w) is computed by sequentially applying $R()$, $Tr()$ and $S()$ to a matrix $\mathbf{I}$, which is a $n_1 \times n_2$ matrix who has values evenly distributed at the $(n_2+1)/2$-th column if n_2 is odd, or at both the $n_2/2$ and $n_2/2+1$-th column if it is even:

$$\mathbf{T} = S(Tr(R(\mathbf{I}, d - \frac{\pi}{4}), \mathbf{c}), w). \tag{3}$$

where

$$\mathbf{I} = \begin{cases} \begin{pmatrix} 0 \cdots, \frac{1}{n_1}, \cdots, 0 \\ \vdots \\ 0 \cdots, \frac{1}{n_1}, \cdots, 0 \end{pmatrix} (n_2 \text{ is odd}) \\ \begin{pmatrix} 0 \cdots, \frac{1}{2n_1}, \frac{1}{2n_1}, \cdots, 0 \\ \vdots \\ 0 \cdots, \frac{1}{2n_1}, \frac{1}{2n_1}, \cdots, 0 \end{pmatrix} (n_2 \text{ is even}) \end{cases} \tag{4}$$

and $\frac{pi}{4}$ is an offset to adjust the density d due to its definition [5].

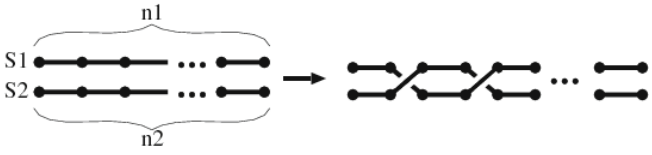


Fig. 2. Twisting a chain of line segments around each other

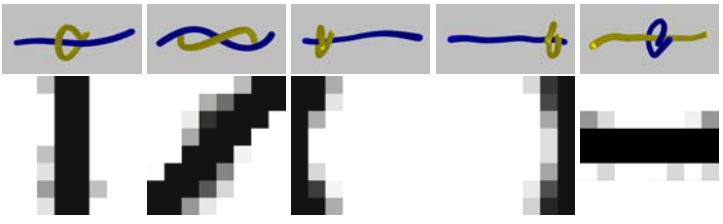


Fig. 3. (upper) Tangles with different density and center, and (lower) the distribution of elements with large absolute values in the corresponding writhe matrices. The darkness represents the amplitude of the absolute value.

Once the desired writhe matrix T_d is computed, the character is guided to the desired posture by updating the generalized coordinates so that the writhe matrix of the character T becomes similar to the desired writhe matrix T_d. This problem is solved by quadratic programming:

$$\min_{\Delta\mathbf{q_1},\Delta\mathbf{q_2},\delta} \|\Delta\mathbf{q_1}\|^2 + \|\Delta\mathbf{q_2}\|^2 + \|\delta\|^2 \text{ s.t.} \tag{5}$$

$$\Delta\mathbf{T} = \frac{\partial\mathbf{T}}{\partial\mathbf{q_1}}\Delta\mathbf{q_1} + \frac{\partial\mathbf{T}}{\partial\mathbf{q_2}}\Delta\mathbf{q_2} \tag{6}$$

$$|T_{i,j} + \Delta T_{i,j}| \leq \sigma(1 \leq i \leq n_1, 1 \leq j \leq n_2) \tag{7}$$

$$\mathbf{T} + \Delta\mathbf{T} - \mathbf{T^d} + \delta = \mathbf{0} \tag{8}$$

$$\mathbf{r} = \mathbf{J_1}\Delta\mathbf{q_1} + \mathbf{J_2}\Delta\mathbf{q_2} \tag{9}$$

where $(\mathbf{q_1}, \mathbf{q_2})$ are the generalized coordinates of the two chains, $(\Delta\mathbf{q_1}, \Delta\mathbf{q_2})$ are their updates to be made at this iteration, $\Delta\mathbf{T_d}$ is the update of the writhe matrix, σ is a threshold, that is set to 0.2 in our experiments to avoid the segments to approach too close to each other, δ is a vector of slack parameters introduced to minimize the difference of the desired writhe matrix and that of the controlled characters, Equation (9) represents the other kinematical constraints which can be linearized with respect to $(\Delta\mathbf{q_1}, \Delta\mathbf{q_2})$ when the movement is small, such as the movements of any parts of the body in Cartesian coordinates or the center of mass. $\mathbf{J_1}, \mathbf{J_2}$ are the Jacobians of this constraint, and $\mathbf{r}$ is the linearized output of this constraint. The updated generalized coordinates $(\mathbf{q_1} + \Delta\mathbf{q_1}, \mathbf{q_2} + \Delta\mathbf{q_2})$ correspond to the target topology coordinate at the next time step, $(w + \Delta w, d + \Delta d, \mathbf{c} + \Delta\mathbf{c})$. By solving Equation (5) at every time step, we simulate the interactions of two virtual wrestlers.

3.3 Real-Time Control of Wrestlers

In this subsection, we explain how to simulate the interactions of two wrestling characters, each controlled by different game players. We assume one character is attacking the other character by moves that involve a lot of tangling.

Let us first give an overview of the computation of the characters' motions. The motion of each character is computed sequentially by two different quadratic programming problems. The attacker's movement is guided by the topology coordinates - once the attack is specified, the attacker tries to tangle its body with the defender's body according to the target configuration. The attack can be switched in the middle if the player thinks a different attack is more effective under the current configuration. The defender needs to escape from the attacks by controlling the body segments involved in the tangling process. The player uses a pointing device to move a body segment and the posture of the defender is computed by inverse kinematics based on quadratic programming.

Now the method to control the attacker is explained. The attacker's motion is determined based on the defender's current posture and the target topology coordinates in the next time step. Let us assume the configurations of the attacker and defender are represented by $\mathbf{q_1}$ and $\mathbf{q_2}$, respectively. Solving Equation (8) is not a good idea to compute the motion of the attacker, as the updates of the attacker's movements takes into account the movements of the defender in the next time step. This makes the defender difficult to escape from the attacker. Therefore, we solve the following problem for computing the motion of the attacker:

$$\min_{\Delta\mathbf{q_1},\delta} \|\Delta\mathbf{q_1}\|^2 + \|\delta\|^2 \text{ s.t.} \tag{10}$$

$$\Delta\mathbf{T} = \frac{\partial\mathbf{T}}{\partial\mathbf{q_1}}\Delta\mathbf{q_1} \tag{11}$$

$$|T_{i,j} + \Delta T_{i,j}| \leq \sigma(1 \leq i \leq n_1, 1 \leq j \leq n_2) \tag{12}$$

$$\mathbf{T} + \Delta\mathbf{T} - \mathbf{T^d} + \delta = \mathbf{0} \tag{13}$$

$$\mathbf{r_1} = \mathbf{J_1}\Delta\mathbf{q_1} \tag{14}$$

where $\mathbf{r_1}$ represents the kinematic parameters of the attacker. This technique adds an effect of physiological delay for launching a response motion with respect to the defender's movements, which increases the realism of the interaction. It also adds an essence of a game play to the interaction between the two virtual wrestlers as the attacker may not be able to achieve the target topology coordinates if the defender is controlled well.

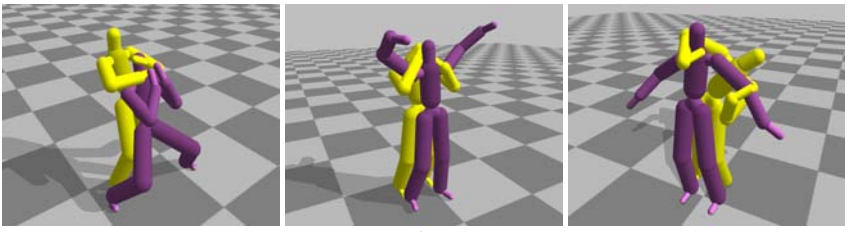
Next, the defender's movement is computed by solving the following inverse kinematics problem:

$$\min_{\Delta\mathbf{q_2}} \|\Delta\mathbf{q_2}\|^2 \text{ s.t.} \tag{15}$$

$$\Delta\mathbf{T} = \frac{\partial\mathbf{T}}{\partial\mathbf{q_2}}\Delta\mathbf{q_2} \tag{16}$$

$$|T_{i,j} + \Delta T_{i,j}| \leq \sigma(1 \leq i \leq n_1, 1 \leq j \leq n_2) \tag{17}$$

$$\mathbf{r_2} = \mathbf{J_2}\Delta\mathbf{q_2}. \tag{18}$$



(a) (b) (c)

Fig. 4. Various wrestling interactions created by the proposed method

where $\mathbf{r_2}$ represents the kinematic parameters of the defender, based on the input from the pointing device and any other kinematic constraints. The GLI between every segment pairs are still considered to avoid penetrations.

The attacker's motion is computed first by solving *Equation* (10) and the attacker's posture is updated. Then, the defender's motion is computed by solving *Equation* (15). Once both character's motion is updated, the time counter is increased.

The key point of controlling the defender is to move the body such that it can efficiently escape from the attacks. Such movements are those which can reduce the writhe value by little motion. For example, when the attacker starts to shift to a configuration of a rear-choke hold (Figure 4 (a)) , which is an attack to squeeze the neck from behind, the most efficient way to escape is to knee down at the last moment, which requires little movement. On the other hand, if the attacking player can predict such escaping moves of the defender and switch to another move that can make use of such movements, the player can efficiently tangle the attacker's body to the defender.

4 Experimental Results

We have simulated a number of interactions between two virtual wrestlers both controlled by game players. Those include rear-choke hold (Figure 4 (a)), Full Nelson hold (Figure 4 (b)) and a number of different squeezing motions from the back. The virtual wrestlers start from separate postures and the attacker approaches to the defender to tangle its body with it.

In our demos, we used a human character model of 42 DOF. All DOFs are used when controlling the characters. For the human character model, when one character is controlled, we can obtain an interactive rate of 40 frames per second when using a Pentium IV 3.2 GHz PC. When two characters are simultaneously controlled, we can still obtain a frame rate of 30 frames per second. We use ILOG CPLEX 9.1 [2] as our quadratic programming solver.

The movement of the attacker is controlled by specifying the topology coordinates. In the first animation, the attacker performed an attack by tangling i) it's right arm with the neck of the defender and ii) it's left arm with the left arm of the defender. Without controlling the defender, this attack can be performed easily by changing the topology coordinates of the attacker as shown in Figure 5.

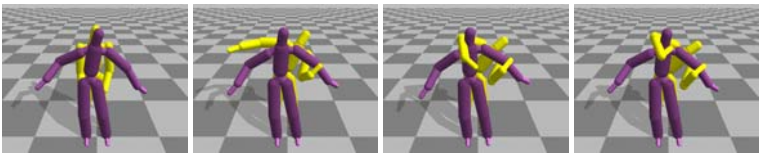


Fig. 5. Without controlling the defender (in purple), the attacker (in yellow) can tangle with the defender easily by changing the topology coordinates

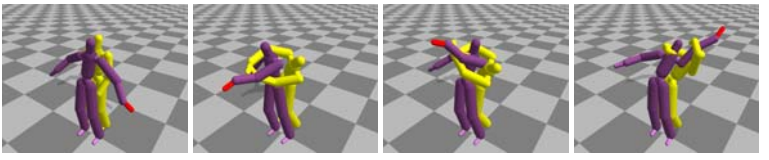


Fig. 6. The defender (in purple) cannot escape from the attack if the player does not control it quick enough

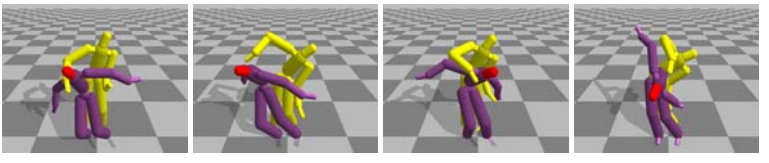


Fig. 7. The defender (in purple) escapes from the attack

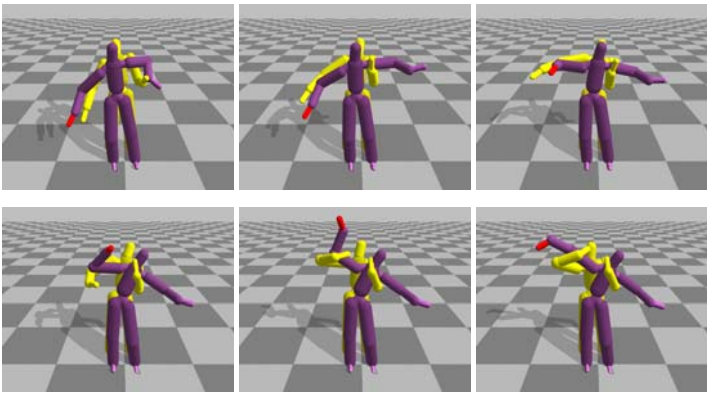


Fig. 8. The attacker (in yellow) switched the attack in the middle in order to lock the defender (in purple)

In the second example, the defender tries to escape from the attacks shown in Figure 5. In our implementation, the player can specify kinematic constraints on the defender by dragging the body segments using a pointing device. In Figure 6, the player dragged the left hand (colored in red) of the defender. However, the defender fails to escape from the attack as it was not controlled quick enough. In Figure 7, the defender escapes from the attack successfully by moving the torso quickly and vigorously.

The attacker can also switch to another wrestling move in the middle of the attack. In the third example, the defender escapes from the attack by blocking the right arm of the attacker in the early stage of the interaction. Then the attacker switches to another wrestling move by tangling its right arm with the right arm of the defender and its left arm with the torso and neck of the defender. Finally the attacker successfully locks with the defender. The screenshots of this interaction are shown in Figure 8.

5 Discussions and Future Work

In this paper, we have introduced a method to make use of topology coordinates for real-time interactions of two virtual wrestlers each of which controlled by game players. By computing the movements of the two characters by two independent quadratic programming problems, we can increase the reality of the interactions and the essence of the game play. Our experimental results show that the application of topology coordinates greatly increases the attractiveness of wrestling games from the following view points:

1. The players, especially the defender has great access to the kinematics of the virtual wrestlers which increases the variety of movements.
2. Although the kinematics are controlled interactively, the topology coordinates automatically guide the characters to avoid collisions and penetrations.

In the future, we would like to conduct a user study to examine our user interface and seek for better ways to control the characters.

References

1. WWE: Smackdown vs Raw, `http://www.smackdownvsraw.com`
2. ILOG Inc, CPLEX, `http://www.ilog.com`
3. Hecker, C., Raabe, B., Enslow, R.W., DeWeese, J., Maynard, J., van Prooijen, K.: Real-time motion retargeting to highly varied user-created morphologies. ACM Trans. Graph. 27(3), 1–11 (2008)
4. Ho, E.S.L., Komura, T.: Wrestle alone: Creating tangled motions of multiple avatars from individually captured motions. In: Proceedings of Pacific Graphics 2007, pp. 427–430 (2007)
5. Ho, E.S.L., Komura, T.: Character motion synthesis by topology coordinates. Computer Graphics Forum (Special issue of Eurographics 2009) 28(2) (2009)
6. Ho, E.S.L., Komura, T.: Indexing and retrieving motions of characters in close contact. IEEE Transactions on Visualization and Computer Graphics 15(3), 481–492 (2009)
7. Jakobsen, T.: Advanced character physics. In: Game Developers Conference Proceedings, pp. 383–401 (2001)

8. Kagami, S., Kanehiro, F., Tamiya, Y., Inaba, M., Inoue, H.: Autobalancer: An online dynamic balance compensation scheme for humanoid robots. In: DonaldK, B.R., Lynch, M., Rus, D. (eds.) Robotics: The Algorithmic Perspective Workshop on Algorithmic Foundations of Robotics, pp. 329–340 (2001)
9. Kulpa, R., Multon, F., Arnaldi, B.: Morphology-independent representation of motions for interactive human-like animation. Computer Graphics Forum 24(3), 343–351 (2005)
10. Lee, J., Lee, K.H.: Precomputing avatar behavior from human motion data. In: Proceedings of 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 79–87 (2004)
11. Lee, J., Shin, S.Y.: A hierarchical approach to interactive motion editing for human-like figures. In: Proceedings of SIGGRAPH 1999, pp. 39–48 (1999)
12. Liu, C.K., Hertzmann, A., Popović, Z.: Composition of complex optimal multi-character motions. In: ACM SIGGRAPH / Eurographics Symposium on Computer Animation, pp. 215–222 (2006)
13. Nakamura, Y., Hanafusa, H.: Inverse kinematics solutions with singularity robustness for robot manipulator control. Journal of Dynamic Systems, Measurement, and Control 108, 163–171 (1986)
14. Phillips, C.B., Zhao, J., Badler, N.I.: Interactive real-time articulated figure manipulation using multiple kinematic constraints. Computer Graphics 24(2), 245–250 (1990)
15. Pohl, W.F.: The self-linking number of a closed space curve. Journal of Mathematics and Mechanics 17, 975–985 (1968)
16. Shin, H., Kovar, L., Gleicher, M.: Physical touch-up of human motions (2003)
17. Shum, H.P.H., Komura, T., Yamazaki, S.: Simulating competitive interactions using singly captured motions. In: Proceedings of ACM Virtual Reality Software Technology 2007, pp. 65–72 (2007)
18. Shum, H.P.H., Komura, T., Yamazaki, S.: Simulating interactions of avatars in high dimensional state space. In: ACM SIGGRAPH Symposium on Interactive 3D Graphics (i3D) 2008, pp. 131–138 (2008)
19. Treuille, A., Lee, Y., Popovic', Z.: Near-optimal character animation with continuous control. ACM Transactions on Graphics 26(3), 7:1–7:7 (2007)
20. Whitney, D.E.: Resolved motion rate control of manipulators and human prostheses. IEEE Transactions on Man-Machine Systems 10, 47–53 (1969)
21. Yamane, K., Nakamura, Y.: Natural motion animation through constraining and deconstraining at will. IEEE Transactions on Visualization and Computer Graphics 9(3), 352–360 (2003)
22. Yamane, K., Nakamura, Y.: Dynamics filter - concept and implementation of on-line motion generator for human figures. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 688–694 (2000)
23. Zhao, J., Badler, N.I.: Inverse kinematics positioning using nonlinear programming for highly articulated figures. ACM Transactions on Graphics 13(4), 313–336 (1994)