Optimization Control for Biped Motion Trajectory

Yumeng Wang
Interactive Digital Media Technology Research Center,
Institute of Automation, Chinese Academy of Sciences
Beijing, China
Email: yumeng.wang@ia.ac.cn

Zheng Wang
Interactive Digital Media Technology Research Center,
Institute of Automation, Chinese Academy of Sciences
Beijing, China
Email: zheng.wang@ia.ac.cn

Guanbo Bao
Interactive Digital Media Technology Research Center,
Institute of Automation, Chinese Academy of Sciences
Beijing, China
Email: guanbo.bao@gmail.com

Bo Xu
Interactive Digital Media Technology Research Center,
Institute of Automation, Chinese Academy of Sciences
Beijing, China
Email: xubo@ia.ac.cn

***Abstract*—Character animation is important in many applications, such as video games, movie production, and somatosensory interaction. However, it is difficult to control the locomotion of a virtual character in a dynamic environment, because bipeds are under-actuated and unstable. In this paper, we present a novel optimization strategy to improve the realism and stability of virtual character animation. Our trajectory optimization method produces lifelike motions by using the motion capture data. The method describes systematic computation of controllers that can optimize locomotion. We use center of mass (COM) control and foot control to enhance the sense of reality in physical movement, and use tracking control to maintain similarity with the reference motion. Experiment results demonstrate our approach is applicable by simulating walking motion and the optimized motion is both natural and recognizable.**

***Keywords*- character animation, trajectory optimization, biped locomotion control**

I. Introduction

The motion capture technology has been widely used in computer animation, game character set and movie post-production. The motion capture system is based on computer vision and sensor techniques, which measuring, tracking and recording a moving trajectory of the object in the 3D space, it captures the positions and rotations of the points.

In virtual character animation, animators increasingly use motion capture data as input to actuate the animation of virtual characters. Compared with the traditional key-frame animation methods, it is an efficient way for animators to create character animation. Nowadays, the motion capture data can be easily obtained, and does not even need the markers, it is much more convenient to be used.

However, the motion capture data is physically inaccurate, using it directly may inspire a feeling of unreality, such as body swing, foot sliding, body collision, uncoordinated limb and some other unreal body movements.

In this paper, we want to solve these problems. Firstly, we extract the skeleton data from the color images and the depth images captured by the Kinect, the skeleton data contains the positions and rotations of the skeletal joints. Then, we apply a novel optimization method using both center of mass (COM) control and foot control to enhance the sense of reality in physical movement, and using tracking control to maintain similarity with the reference motion to adjust and optimize the motion capture data. In this case, the character animation not only looks more similar to the original motion data, but also has a sense of reality. By using our method, these motion capture data can be applied to animation scene directly, avoiding the trivial post-editing.

II. Related Work

The motion capture tracking controller adaption is an important research problem. Some researchers are using motion capture data to control the motion of virtual character [1][2][3], which can reduce new motions' authoring and editing. There are some discrepancies between original motion data and simulated character, the motion needs to be correct.

The pre-recorded motions are desirable, some methods have been proposed to adapt record data to new situations, while maintaining physical properties [4]. To make optimization easier, [5] learns from low dimensional motion data of stylistic motion. In order to adapt input motions, [6][7] use randomized search. Such strategies have produced many perfect lifelike results. However, their output is limited to remain near provided input data.

Trajectory optimization problem has proved to be difficult, on account of its high degrees of freedom and nonlinear solution. [8] optimizes for motion with complex contact. A quadratic programming problem solves both reference and balance objectives simultaneous [9]. Defining proper objectives is also a difficult problem, even for low-energy motions, such as walking [10].

Several methods focus on locomotion problem. [11] gives a robust controller to build a reference trajectory and a step-based feedback loop to follow the reference trajectory, by employing a series of key poses. [12] shows a method to trace path from a solved problem to the target unsolved problem Continuously. [1] employs Linear Quadratic Regulator (LQR) balance control, and [3] employs a more sophisticated model, Nonlinear Quadratic Regulator (NQR), to track the full degrees of freedom (DOFs) of a human body model. [2] develops walking controllers that appear to be improved robust and stable, their controller employed short-horizon tracking and quadratic programming to maintain biped balance. [13] uses the null-space projection, which is to combine all tasks into a single quadratic program, this can achieve highly-constrained balancing.

To simplify the optimization problem, and to make the problem steerable, some methods use cyclic behaviors. [14] simplifies the foot of character as a point, and optimize gait parameter for morphology. [15] uses a cyclic tractor to generate a variety of running behaviors. [16] simplifies computation, using dimension reduction techniques. These methods usually simplify physical models.

Some methods are used to tune settings for pre-defined controllers, but it may cause more expensive optimization, while adding new controllers. [17] uses action specific controllers, with low dimensional parameter. There are also some research based on basic locomotion controllers [18][19][20]. Optimization tunes control strategies [7], without necessarily resorting to complex analytical models of the dynamics, or careful manual tuning of parameters. In order to allow more human-like gaits, [21] optimizes controllers by using biomechanically motivated objective functions.

As motion capture data is physically inaccurate, it needs to be rectified to make it physically realism. We use some of the control strategies mentioned in [13] and [20] for reference, then synthesize and optimize these methods. Our work shows how virtual character perform locomotion with motion capture data by trajectory optimization, the motion will be both realism as natural and similarity to original motion.

III. Hierarchical Structure

A. Skeleton Model

The Fig. 1 shows the topology of human skeleton. The skeleton totally contains one root and 17 joints. As it shows in the figure, the root node is hip. Since our research is mainly on gait style and trajectory correction, we simplify the model by leaving out the finger joints of the hand , so that it can be more intuitive.

For each joint, there are three rotation parameters to describe the motion information, with which respect to Axis X, Y, Z rotation angles. The hip, which is the root node of the entire body, also contains three-dimensional position parameters, besides the rotation parameters. The skeleton has 57 channels in total.

B. Animation Format

We use the Kinect from Microsoft to capture the motion data, these data record in BVH format files. the BVH format file includes two parts [22], the first part represents hierarchy of the skeleton and its initial posture, the second part contains the animation data.

In the first part of the file, it defines a joint tree, includes the name of each joints, channel number, the relative position between two different neighbor joint nodes. That means it defines the length of bones of every parts of the body.

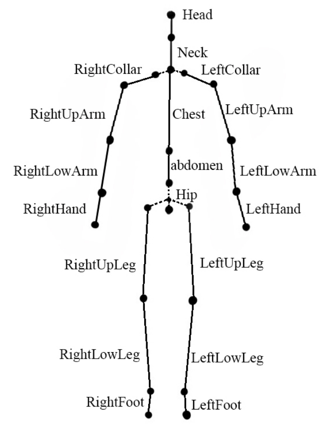

Fig. 1. Skeletal hierarchical structure

The second part of the file records motion data. It defines the duration of data, and time interval between each frame. According to the order of joints defines by the first part, the second part provide the data of every frame, which record skeletal global location and rotation information of each joint.

The world space has been defined as a right handed coordinate system. The Axis Y defines as the world up vector, and the XOZ plane ($Y = 0$) is the ground plane. In our motion data, character walks along the direction of Axis Z.

This rotation representation is called Euler angle representation, we construct the rotation matrix from three separate Eula angles, multiple the separate rotation matrices from three channels. In the right-handed Cartesian coordinate system, the rotation matrix can be expressed as Eq. 1.

$$R\left(\alpha,\beta,\gamma\right) = R_z\left(\alpha\right) R_x\left(\beta\right) R_z(\gamma) \tag{1}$$

The root node is the only one which has the translation data every frame, so that each bone has a base offset that needs to be added to the local matrix stack. The translation matrix T, represents the summation of the bones base position and frame translation data.

With the rotation matrix R and translation matrix T be calculated, R is a 3×3 matrix, T is a 3×1 matrix, we can get the transform matrix M.

$$M = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \tag{2}$$

If we need to calculate the global position of a given bone, this bone's local transform should multiple the parent bone's global transform, which is derived from grandparent's global transform and so on. As Eq. 3 shows, n is the current bone, his parent is $n-1$, the root bone is $n = 0$.

$$M_{global_n} = M_{global_0} \prod_{i=1}^{n} M_{local_i} \tag{3}$$

IV. Optimization

The reference motion queues are provided by the sequence of patterns from the motion data. The data driven controller can reproduce the reference trajectory. The controller maintains the balance between the actual simulation and the reference trajectory.

With regard to the reference motion sequence, we consider $Q(t)$ as a fragment of motion.

$$Q(t) = (p_0(t), q_0(t), ..., q_n(t)) \tag{4}$$

Where t is the current frame during the motion, $p_0 \in T^3$ is the position of the root node, $q_0 \in R^3$ is the orientation of the root node, $q_k \in R^3$ for $k > 0$ is other joint's orientation to its parents. $Q(t)$ represents the time varying pose of the character during each frame.

The motion interactive with ground is much more complex than free motion, but the motion from captured data is a kind of free motion, so we need to adjust it to the environment.

We choose a set of objectives to optimize character's motion. Trajectory optimization is hard to directly apply to full body poses, so we apply it to joint torques. Our method use the energy objective function, i represents each objective, objective E_i are combined with weight α_i , subject to dynamics constrains.

$$\begin{aligned} E(Q(t)) &= \sum_i \alpha_i E_i(Q(t)) \\ &= \alpha_c E_{COM} + \alpha_f E_{foot} + \alpha_t E_{track} \end{aligned} \tag{5}$$

In our method, as show in Eq. 5, objective function is the summation of three parts. We calculate energy function of center of mass (COM), foot, and tracking separately, and then integrate them together to get the final trajectory optimization. The function is constrained to avoid the immersion between different body parts, and the collision between body and system.

The virtual character's movement is determined by forward dynamics, and we adjust its motion by inverse dynamic. We apply the joint torques to the virtual character, these joint torques is based on joint accelerations and the contact with ground. The joint torques drive the motion of character, which satisfy some objectives, each objectives has a different function.

We divide the optimization problem into some parts of phases, because it is difficult to optimize the trajectory during a long period of time. Each phase is a fixed duration of the motion, each period of time can be regarded as a finite state machine cycle.

A. State Machine

The control of objectives depends on a finite state machine, as shown in Fig. 2. Every state has its own posture, as the two legs are symmetrical, the gaits should be symmetry during a circle. In the first state, the right leg is supported by the ground with the left foot lift-up, in the second state, the character puts down its left foot, with the heel touch down. Symmetry to the first two states, the third state with the left leg support, and the fourth state with the right heel touch down.

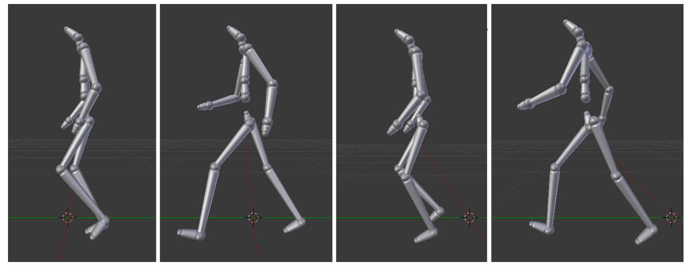

Fig. 2. Finite state machine of walking

The controller uses motion capture data, other than manual design, as a result, it is hard to distinguish one state from another. In order to simplify the optimization problem, we need to firstly take it as a periodic motion, and then separate the motion into each cycle. Since the motion is from the captured data which is different from key-frame animation, each cycle of the captured data may be has different length, we can separate them by using relative displacement of some specific points, in this way, the problem can be controllable.

During the period from toe-off to heel-down, we call it the swing phase, and for the rest part, from heel-down to the foot support by the ground and then toe-off, this duration is called the stance phase. As shown in Fig. 3.

B. COM Control

The center of mass (COM) controller is responsible to regulate linear momentum. Along the direction of the character forward, COM is controlled as a sinusoid path. We regulate it to the center of the base support. In order to balance the body, COM should be placed over the foot which is supporting by the ground. When the swing foot enter into the stance phase and touch the ground, the trajectory of COM should be recomputed.

In our method, the motion perpendicular to the ground in Axis Y direction will not be constrained, avoiding the possible conflict with the dynamic system.

We need to keep the projection of COM close to the center of support, the COM controlled through desired accelerations.

$$\ddot{c}_{des} = k_l(c_r - c) + k_d(\dot{c}_r - \dot{c}) \tag{6}$$

Where c is the position of the COM, c_r is the reference position of COM, which can be chosen as the center of support, $\dot{c}$ is the velocity of c, and $\dot{c}_r$ is the reference velocity of c. k_l is proportional gains and k_d is the derivative gains. We usually set the parameters as $k_d = 2\sqrt{k_l}$ to control the acceleration.

For the optimization, the COM objective E_{COM} can be stated as Eq. 7, $\ddot{c}$ is the acceleration of COM.

$$E_{COM} = \|\ddot{c}_{des} - \ddot{c}\|^2 \tag{7}$$

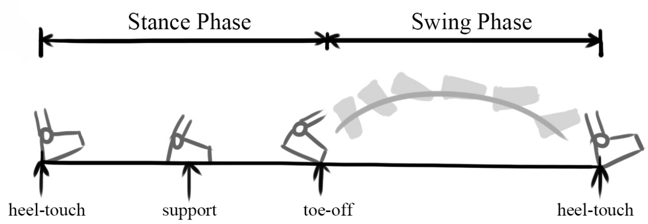

Fig. 3. The motion of one foot in stance phase and swing phase

C. Foot Control

The problem of foot control can be divided into two parts, as we can see in Fig. 3, we set two objectives to control the feet, $E_{contact}$ and E_{swing} . When a foot is in the stance phase, the contact condition enforces the objective $E_{contact}$. When a foot is in the swing phase, the objective E_{swing} controls its trajectory.

In the stance phase, during the time from heel-down to support, the position of heel should not be changed, during the time from support to toe-off, the position of toe should not be changed. Based on this, we can calculate the objective function of the stance phase as Eq. 8.

$$\begin{aligned} E_{contact} &= \phi E_{heel} + (1-\phi)E_{toe} \\ &= \phi\left(y_{heel}^2 - k y_t oe\right) + (1-\phi)\left(y_{toe}^2 - k y_{heel}\right) \end{aligned} \tag{8}$$

Where $\phi \in \{0,1\}$, when the time from heel-down to support, $\phi = 1$, the objective function only add constraint on the heel; when the time from support to toe-off, $\phi = 0$, the objective function only add constraint on the toe. When the foot contact to the ground plane, there should be $y = 0$, therefore, when the heel touch the ground, $y_{heel} = 0$ and $y_{toe} > 0$, when the toe touch the ground, $y_{toe} = 0$ and $y_{heel} > 0$. If $y \neq 0$, the foot may go through the ground, or cannot touch the ground. In our method, we can recover this unnatural thing.

In the swing phase, the foot rises up firstly, then reaches a certain height, and then lowers it down in the end. At the same time, the foot moves forward.

$$y_{swing} = \sqrt{(k_h l_{hip}^2 + l_{step}^2)} \tag{9}$$

The Eq. 9 shows the target location for the foot. Hip may swing up and down while walking, l_{hip} is the relative value of default hip value in the Axis Y direction, l_{step} the length of step. Therefore the objective function shows as follow, and y is the reference location for the foot.

$$E_{swing} = \|y_{swing} - y\|^2 \tag{10}$$

As the symmetry of gait, when one feet is in the stance phase, the other one should in the swing phase. The foot objective is the sum of these two objectives.

$$E_{foot} = E_{contact} + E_{swing} \tag{11}$$

D. Tracking Control

The center of mass control and the foot control which are discussed in the previous chapters, can make the motion produced by the motion capture data more natural and smooth. Except for that we need the motion keep highly similarity to original motion, in case of changing to an unexpected motion. For this reason, we need tracking control.

We apply torques to each joint, which is computed by proportional derivative, so that we can drive each joint to its target angle.

The purpose of tracking control is to follow a trajectory as closely as possible. The desired acceleration is computed as follow.

$$\ddot{\theta}_{des} = k_t\left(\theta_r - \theta\right) + k_v\left(\dot{\theta}_r - \dot{\theta}\right) + \ddot{\theta}_r \tag{12}$$

Where θ_r is the motion coordinate position, $\dot{\theta}_r$ is the motion coordinate velocity, $\ddot{\theta}_r$ is the motion acceleration. All these can extract from motion data. Generally, the parameters are set as $k_v = 2\sqrt{k_t}$ to control the acceleration.

The tracking objective function is computed as follow, and $\ddot{\theta}$ is the reference motion acceleration.

$$E_{track} = \left\|\ddot{\theta}_{des} - \ddot{\theta}\right\|^2 \tag{13}$$

This can be used for any joints of the skeleton. This objective determines which bone of the skeleton is more useful during tracking and balancing, by regulating the weights of each joint. Therefore, the tracking objective is as follow.

$$E_{track} = \sum_i \left\|\omega_i\left(\ddot{\theta}_{des_i} - \ddot{\theta}_i\right)\right\|^2 \tag{14}$$

Each joint has a different influence to the naturalness of the whole character action, in order to distinguish the different significance of each joints, we apply different weights to each joint. If we want to improve the utilization of one joint, the user can lower the weights of it.

V. Results

The motion capture data we use in the experiment is captured by Kinect, which is made by Microsoft, these data are used as original data in our experiment, We capture the data of a man walking forwards, which cannot be directly used onto the character animation, for it may has some unclear and unstable problems.

There are several previous methods to simulate animation is based on physics. These motion controllers in our paper are used as reference. When the motion capture data use onto a character, the character motion needs to be more natural and real. In addition, we don't want to change too much of the style of the original motion, it is improper that the character animation may be quite the other way of the original motion.

In our experiment, a circle of state machine is about 160 frames. The circle in our simulation starts from the left foot stand on the ground with the right foot swing to the front. The world space follows the right handed coordinate system,

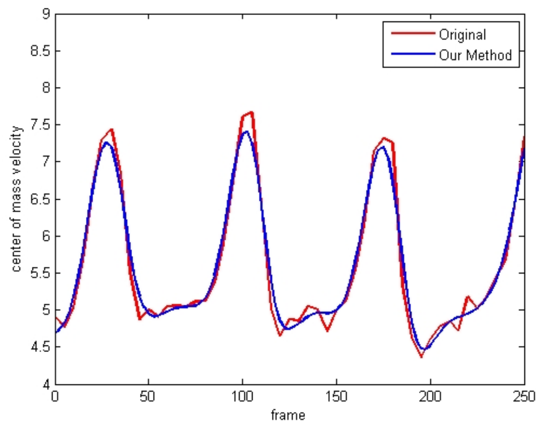

Fig. 4. The velocity of COM in z direction

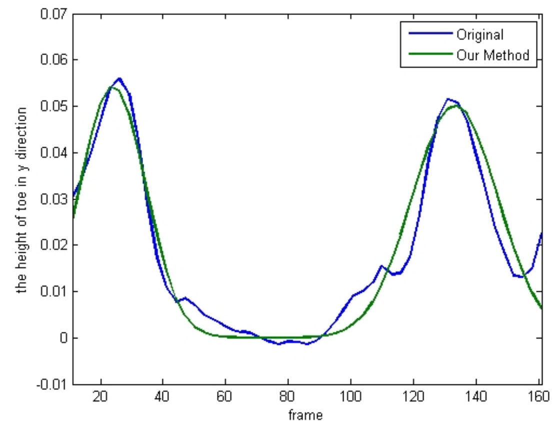

Fig. 5. The optimization of the toe motion

with the Axis Y is the world up vector, and XOZ plane is the ground, Axis Z is the walking direction of the character.

We can use the midpoint of hip, as the position of COM for model simplification. The COM should be placed over the foot which is supported by the ground, this is a balance posture for biped character. When character changes its stance foot, the hip may swing to the other side in x direction.

While the hip swing left and right, the natural walk of human is actually a non-uniform motion, when we keep under observation of the character's gait. The period we swing one foot from back to forward, the value of velocity is much higher than the period of changing the stance foot from one to the other. As Fig. 4 shows, the velocity of COM in z direction, the red line is the original motion capture data.

We optimize the trajectory of the COM. As shown in Fig. 4, the blue line is the velocity of COM after optimization, to be a comparison of original data. The character should not slide back, when it walks forward.

In order to make the motion more natural, we not only need to control the COM, but also need to control the feet. We represent the movement of toe and heel of one foot in a circle of state machine.

With the method of foot control, we optimize both heel and toe. This is different from some other method which simplifies the foot to single point, in order to simplify the optimization problem. The optimization of the toe shows in Fig. 5, the optimization of the heel shows in Fig. 6. In the figure, the blue line represents the original motion data, and the green line represents the final motion after optimization.

During the period of touching the ground, the tiny vibration should be smoothing, the non-uniform trembling is not supposed to be present. This unstable situation generates from the motion capture. In the period from heel down to the support, the height of heel should be $y = 0$, in the period from support to toe off, the height of toe should be $y = 0$, and it should be a smooth transition between the stance phase and the swing phase.

In the state machine circle while walking, the heel touch

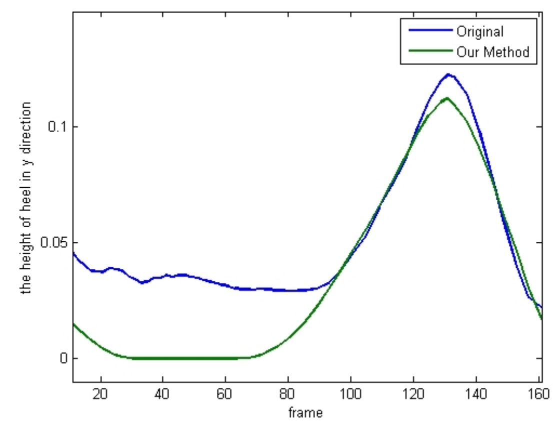

Fig. 6. The optimization of the heel motion

the ground first, and then the toe. While both of heel and toe touch down, the foot is supported by the ground, this is the stance phase. The heel lift up first to leave the ground, and then the toe, while the foot swing forward, both the toe and the heel should not touch the ground any more, this is the swing phase.

The Fig. 7 shows the height of the heel and the toe in y direction, during a circle of motion. Axis Y shows the height of the toe or the heel at a certain moment. The blue line shows the height of toe, and the green line shows the height of heel.

As out settings, the character walks along Axis Z. When the foot touches the ground, it should be motionless, in case of foot sliding, which is the biggest problem of unnatural, as shown in Fig. 8, the green line is toe, and the blue line is heel. When the foot is on the ground, both toe and heel should not have any move in z direction, so that the velocity in z direction equals zero.

VI. Conclusion

In this paper, we present a novel method to improve the realism and stability of virtual character animation by using

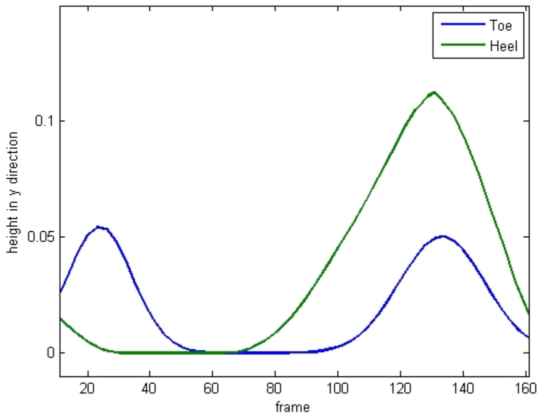

Fig. 7. The height of foot during a cycle of walking

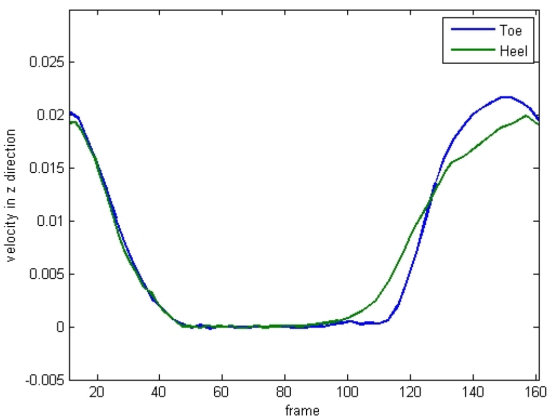

Fig. 8. The velocity of foot in z direction

COM and foot control. It also improves the similarity to original motion by tracking control. We synthesize these terms with different weights to optimize motion trajectory. The experiments show better results for trajectory optimization. This method can greatly enhance the efficiency of virtual character animation production.

There are also some limitations in our work. For the moment, our optimization method has only been used on the walking style, some other motion styles have not been tested. The parameters of the objective function are chosen manually. If we use this method on other kinds of motion style, the gain value should make some adjustment. In addition, the character model we use in our method is rigid body, and it has not been skinned, thus we should tune some set of parameters in the skinning animation. In the future, we will study method of several kinds of motion style, a unified motion objective function would be needed, and the gain constants should be self-regulation. And then we will apply the skeletal animation to skinned character, which need to add an objective item about geometry deformation, to avoid the collision between contact parts.

ACKNOWLEDGMENT

The authors thank the 863 Plan (No. 2013AA013803) and the National Science Foundation of China (No. 61103153/F020503) for financially supporting this study.

REFERENCES

[1] da Silva, Marco, Yeuhi Abe, and Jovan Popovic. *Interactive simulation of stylized human locomotion*, ACM Transactions on Graphics (TOG). Vol. 27. No. 3. ACM, 2008.
[2] Da Silva, Marco, Yeuhi Abe, and J. Popovic. *Simulation of Human Motion Data using Short Horizon Model Predictive Control*, Computer Graphics Forum. Vol. 27. No. 2. Blackwell Publishing Ltd, 2008.
[3] Muico, Uldarico, et al. *Contact-aware nonlinear control of dynamic characters*, ACM Transactions on Graphics (TOG). Vol. 28. No. 3. ACM, 2009.
[4] Gleicher, Michael. *Retargetting motion to new characters*, Proceedings of the 25th annual conference on Computer graphics and interactive techniques. ACM, 1998.
[5] Safonova, Alla, Jessica K. Hodgins, and Nancy S. Pollard. *Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces*, ACM Transactions on Graphics (TOG) 23.3 (2004): 514-521.
[6] Liu, Libin, et al. *Sampling-based contact-rich motion control*, ACM Transactions on Graphics (TOG) 29.4 (2010): 128.
[7] Sok, Kwang Won, Manmyung Kim, and Jehee Lee. *Simulating biped behaviors from human motion data*, ACM Transactions on Graphics (TOG). Vol. 26. No. 3. ACM, 2007.
[8] Mordatch, Igor, Emanuel Todorov, and Zoran Popovic. *Discovery of complex behaviors through contact-invariant optimization*, ACM Transactions on Graphics (TOG) 31.4 (2012): 43.
[9] Abe, Yeuhi, Marco da Silva, and Jovan Popovic. *Multiobjective control with frictional contacts*, Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, 2007.
[10] Al Borno, Mazen, Martin de Lasa, and Aaron Hertzmann. *Trajectory optimization for full-body movements with complex contacts*, Visualization and Computer Graphics, IEEE Transactions on 19.8 (2013): 1405-1414.
[11] Yin, KangKang, Kevin Loken, and Michiel van de Panne. *Simbicon: Simple biped locomotion control*, ACM Transactions on Graphics (TOG). Vol. 26. No. 3. ACM, 2007.
[12] Yin, KangKang, et al. *Continuation methods for adapting simulated skills*, ACM Transactions on Graphics (TOG) 27.3 (2008): 81.
[13] Macchietto, Adriano, Victor Zordan, and Christian R. Shelton. *Momentum control for balance*, ACM Transactions on Graphics (TOG). Vol. 28. No. 3. ACM, 2009.
[14] Wampler, Kevin, and Zoran Popovic. *Optimal gait and form for animal locomotion*, ACM Transactions on Graphics (TOG). Vol. 28. No. 3. ACM, 2009.
[15] Nunes, Rubens F., et al. *Using natural vibrations to guide control for locomotion*, Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. ACM, 2012.
[16] Chai, Jinxiang, and Jessica K. Hodgins. *Constraint-based motion optimization using a statistical dynamic model*, ACM Transactions on Graphics (TOG) 26.3 (2007): 8.
[17] Coros, Stelian, et al. *Locomotion skills for simulated quadrupeds*, ACM Transactions on Graphics (TOG) 30.4 (2011): 59.
[18] Coros, Stelian, Philippe Beaudoin, and Michiel van de Panne. *Generalized biped walking control*, ACM Transactions on Graphics (TOG) 29.4 (2010): 130
[19] Lee, Yoonsang, Sungeun Kim, and Jehee Lee. *Data-driven biped control*, ACM Transactions on Graphics (TOG) 29.4 (2010): 129.
[20] de Lasa, Martin, Igor Mordatch, and Aaron Hertzmann. *Feature-based locomotion controllers*, ACM Transactions on Graphics (TOG) 29.4 (2010): 131.
[21] Wang, Jack M., David J. Fleet, and Aaron Hertzmann. *Optimizing walking controllers*, ACM Transactions on Graphics (TOG). Vol. 28. No. 5. ACM, 2009.
[22] Meredith M, Maddock S. *Motion capture file formats explained*, Department of Computer Science, University of Sheffield, 2001, 211.