EUROGRAPHICS 2009 / P. Dutré and M. Stamminger
(Guest Editors)

Volume 28 (*2009*), *Number 2*

Physics-driven Multi Dimensional Keyframe Animation for Artist-directable Interactive Character

Hironori Mitake[1], Kazuyuki Asano[1], Takafumi Aoki[1], Salvati Marc[1], Makoto Sato[1] and Shoichi Hasegawa[2]

[1]Tokyo Institute of Technology [2]University of Electro Communications

Abstract

Various forms of art and entertainment involve many different characters, and advances in human interfaces have necessitated physical interactions in order to develop an improved sense of reality. In this paper we propose a method for generating the motions of characters using multidimensional keyframe animation in parallel with real-time physical simulation. The method generates characters capable of physical interaction, and also allows animators to use traditional methods for designing character motion. We have implemented the system and confirmed its effectiveness experimentally.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.6]: Methodology and Techniques—Interaction Techniques; Computer Graphics [I.3.7]: Three-dimensional Graphics and Realism—Virtual Reality; Animation

1. Introduction

In this paper, we propose a motion generation method combining real-time physical simulation and prepared animation using a multidimensional keyframe animation, to realize empathetic characters with realistic physical interactions, while reducing the amount of preparation needed.

With the recent development of human interface and display technology, virtual worlds used for works such as games, entertainment and media arts are becoming more immersive and empathetic. Flow of information between users and virtual worlds is significantly increased using technologies such as haptic devices, motion sensors and 3D high resolution images.

These interfaces also allow for users to have “physical interaction”, more direct interaction using their own bodies, just as they would in the real world. Therefore, making the virtual world itself more appealing is becoming more and more important. To achieve this, we particularly focus on characters in virtual worlds because they attract, interact with and evoke empathy from users. As such, there is also a growing necessity for empathetic characters with realistic interactivity.

Due to the nature of physical interaction, realism is important, and characters must react based on dynamics. When a virtual world is ruled with real world dynamics, users can use real world knowledge to understand and predict the behavior of the virtual world. For example, when a character is pushing an object, users can predict the mass of the object by the speed or exertion of the character.

On the other hand, to evoke empathy from users, characters must tell the users their emotion or intention. Although in games, this information is often expressed using thought balloons or symbols, using character motions like stage or movie actors allows more varied expressions of emotion or intention and helps users to empathize more with the characters. To create such characters, handmade motions are important. Animators make character motions from scratch or modify captured motions for various individualities and situations in scenario. In addition, a particular pose or motion for the character sometimes becomes a key to progress the scenario.

To realize empathetic characters with realistic physical interactivity, characters’ reactions must be varied. However, preparing all of those reaction motions as motion data (e.g. keyframe animations) can create enormous production costs. For practical use, automatic generation of various motions from smaller amounts of motion data is necessary. Those

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

motion generation systems must allow animators to design output motion as easily as creating handmade motions, and without having to acquire new skills, for the purpose of generating motion to express content.

We have set the goal of our motion generation method to realize characters with animator-designed motions and realistic physical interactivity, while reducing the amount of preparation needed. We first describe related works in section 2, then explain the proposed method in section 3, 4 and 5, and lastly evaluate the effectiveness of the method experimentally in section 6.

2. Related Works

There are various studies on characters for physical interactions using, for example, motion databases, physical simulation and combination of both motion database and physical simulation.

2.1. Motion Database and Motion Editing

In methods with motion databases such as [KGP02, PB02, KG04], reaction motion of characters is generated by replaying prepared motion data selected from a motion database corresponding to situations and users' interactions. These methods are still popular for game characters. Since motion databases have to contain every motion for expected interactions, increased variation of character actions or use of input devices which allow more variety of input cause a rapid increase in the motion data required.

As a solution, methods which expand variety of basic input motions and generate motion automatically to meet to the situation are desired. With methods as Motion retargeting [Gle98], motion blending [BW95] and interpolation [WH97,RCB98,CHP07], characters react in various situations only with the preparation of basic motions, rather than having every motion prepared for an expected situation. However, these post-processes sometimes defers the laws of physics. [IAF07] proposed the method to evaluate naturalness of blend results, but it only works with limited situations.

2.2. Physical Simulation

Another solution for generating character motion automatically is to use physical simulation. [HWBO95, FvdPT01, HTH05] applied methods using articulated body simulation. Using these methods, characters can react dynamically in any situation because physical law is simulated. However, articulated body simulation has some difficulties in practical use. To attain a desired motion, creators have to build articulated body models and controllers for the characters. An articulated body model itself has many parameters that must be adjusted to attain the desired motion. Moreover, the controller for such a model often becomes complex and also has many parameters. Since adjusting motion with these parameters is very different from traditional motion design methods, animators are forced to learn additional skills to use these methods.

Physical law is also combined with motion data in several ways. [RGBC96, KSK00] used optimization to make motions ruled by physical law. However, optimizations creates large computational costs and are difficult to implement in real time. To achieve real time motions ruled by physical law and with lower computational cost, [ALP04] used transition patterns of whole body momentum and generated various physically realistic motions from single motion capture data. However, it is not being used for generating reaction motion from force input. [AFO05] used a pre-trained evaluator to discriminate desired modifications, but it still requires ample motion data for a motion database.

[ZH99, OM01, ZH02, WJM06] used articulated body model simulation controlled to fit motion data. [SPF03] switched dynamic motion and motion data according to the situation, and [ZMCF05] used simulation to bridge captured motions. These methods still have the problem of adjusting parameters of the articulated body model, or the need to prepare enough motion data for a motion database.

Simulated characters must be driven by physical controllers. Walking and balancing are basic behaviors for human-like characters, and physical controllers are researched such as using limit cycle control [LvdPF96], inverted pendulum control [KLKK05] or finite state machine [YLvdP07]. On the other hand, [YPvdP05] generates balancing behaviors with data-driven method, but it needs vast amount of prepared motion data. Our method employs inverted pendulum control to maintain balance during gait.

2.3. Simple Physics with Motion Data

A solution for difficulties in articulated models and controllers is combining of simple physical simulations and motion data. Decreasing dimension [SHP04] or simplifying the model [PW99] is used to decrease computational cost of optimization. Some computer games approximate a character as a mass point with collision detection, and achieve output by replaying motion data at the position of a simulated model. Our method employs a single rigid body model for a character. A rigid body is different from a mass point in ability of simulating rotational motion, which is important for expressing balancing or falling down.

A recent game employs multiple pendulum models to generate motion of hanging characters [USS*]. They apply the angle of pendulums to the angles of bones of the animated character. To create realistic motion, they modify the angles of bones by adding values which are proportional to the simulated values. Motion designers adjust the coefficients of added values to create realistic motion. However, the adjustments are not intuitive and take considerable time.

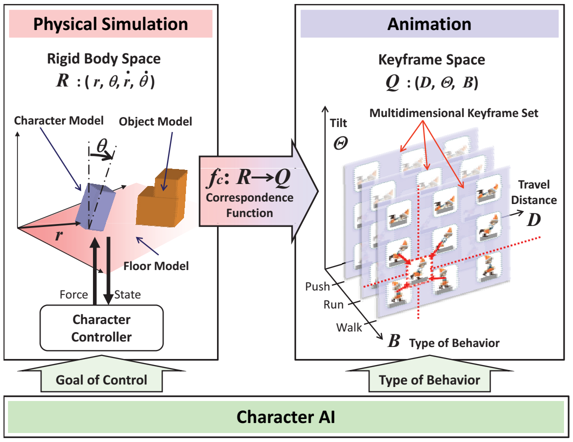


Figure 1: *Overview of the proposed method*

In our method, connection of simulation and motion data which is described with a multidimensional keyframe animation provides intuitive adjustments of output motions.

[dSAP08] employed 3-link model which represents supporting/idling leg and upper body. Optimization to keep balance is solved and combined with walking motion data. The method generates various locomotion for uneven and dynamic floor from single input motion data in real time. However, the method generates reactions within only one step. In our method, inverted pendulum controller for single rigid body model can generate long term reaction motion for user interaction.

3. Overview of Proposed Method

Our motion generation method uses keyframe animation and links it to real-time dynamics simulation. The dynamics simulation calculates physical effects applied to the character by the environment, and the keyframe animation describes the motions by intention and emotion. In this way we obtain characters with empathetic animator-designed motions and realistic physical interactivity. Figure 1 shows an overview of our method.

In our method, both physical simulation and keyframe animation are driven by the command from the character AI. Character AI determines the behavior of each character. AI determines goal of attitude and locomotion for the character controller, which in turn, makes the character model perform the desired behavior. AI also determines a type of behavior for the animation.

We use a real-time rigid body dynamics simulator and a single rigid body model as the physical model of a character. We can then simulate dynamics of the entire body and generate various motions for physical interaction. A character controller controls the character model to attain translational and rotational behavior determined by AI. We used an inverted pendulum-based method, known as one of the simplest approximations of creatures standing with legs [SK91]. Controlling a single rigid body character model does not require advanced knowledge of control, and has few parameters to adjust.

In addition, various keyframe animations prepared by animators are linked to the result of a rigid body simulation to achieve appealing and emotive motions. Character motion is easily designed by changing or adjusting keyframe animations to be linked with a rigid body. For example, preparing different keyframe animation sets for each character results in individual motions for each character. If a keyframe set describing a specific pose is added, then the character will become able to take this pose.

Since various motions are generated automatically with the combination of a rigid body simulation and keyframe animation, creators are required fewer prepared keyframe animations. In addition, keyframe animations are a popular way to describe character motion, and animators can use their existing skills to create.

Details of the proposed method are described in the following sections. Section 4 describes physical simulation and character control. Section 5 describes the multidimensional keyframe set.

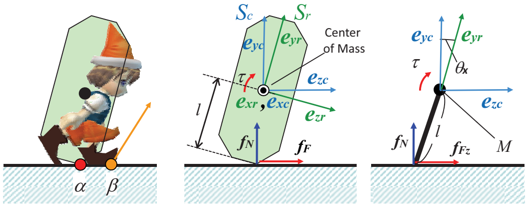
4. Physical Simulation

In this method, we used a real-time physical simulator, which simulates translation and rotation of a rigid body by calculating forces between multiple rigid bodies, such as collisions and friction. Every character and the objects they may come in contact with are represented as rigid body models. The simulation gives characters varied and physically realistic motions to realize physical interaction. Each character model is controlled by the character controller to attain a behavior of the character. We describe the rigid body model and control of the character in following sections.

4.1. Character Model and the Rigid Body Space

The character's body is considered as a mass system. We consider the mass of parts which move differently from the body trunk while walking, such as arms and legs, to be vanishingly small. With this approximation, a walking body is modeled as a single rigid body fixed to the body trunk. Compared to an articulated body, a single rigid body model has much fewer parameters, requiring no complex control and is easy to adjust to obtain a desired motion.

A single rigid body can also be used to approximate conservation of linear and angular momentum of the whole character body. Conservation of momentum is important because it strongly reflects the physical effects of the environment which are intentionally unavoidable for the character. Changes in total linear and angular momentum are only



(a) Targeting situation (b) Rigid Body Model (c) Inverted Pendulum Model

Figure 2: *Physics Model of Character*

caused by external forces applied to the character. The external forces are quite limited according to character conditions (i.e., contact forces are only applied in each contact area with limited strength and direction, and gravity is constantly applied on the center of mass of the character). Therefore, total momentum of the character is determined depending on the environmental forces, such as gravity and collisions.

A rigid body model has parameters for mass, center of mass, inertia tensor and shape. Those parameters are determined from settings of a character in the work and should be adjusted to achieve the desired motion. In particular, stability while standing can be adjusted by changing the bottom shape of the model.

To represent rotation of the rigid body model, we define two coordinate systems S_r and S_c (Figure 2(b)). S_r is the local frame of the rigid body model with a basis of three unit vectors $(\boldsymbol{e}_{xr}, \boldsymbol{e}_{yr}, \boldsymbol{e}_{zr})$. S_c is a coordinate system with a basis of three unit vectors $(\boldsymbol{e}_{xc}, \boldsymbol{e}_{yc}, \boldsymbol{e}_{zc})$, such that $\boldsymbol{e}_{xc}$ and $\boldsymbol{e}_{zc}$ are in the floor surface. Then we define rotation of the rigid body model $\boldsymbol{\theta}(\theta_x, \theta_y, \theta_z)$ as $\sin\theta_x = \boldsymbol{e}_{yr} \cdot \boldsymbol{e}_{xc}$, $\sin\theta_z = \boldsymbol{e}_{yr} \cdot \boldsymbol{e}_{zc}$ and $\theta_y = \arccos(\boldsymbol{e}_{zc} \cdot \boldsymbol{e_z})\,\mathrm{sgn}(\boldsymbol{e}_{zc} \times \boldsymbol{e_z})$, where $\boldsymbol{e_z}$ is the Z-basis of the world coordinate system.

We define $\boldsymbol{r}$ as the center of mass of the rigid body model in the world coordinate system. Finally we define the rigid body space $\boldsymbol{R}$ as $\{\boldsymbol{r}, \boldsymbol{\theta}, \dot{\boldsymbol{r}}, \dot{\boldsymbol{\theta}}\}$.

4.2. Character Control

A character controller controls the rigid body model to attain an intended position and orientation which are determined by the character AI.

We simplified the rigid body model into a physical model similar to an inverted pendulum (Figure 2(c)). The inverted pendulum consists of a massless rod and the point mass at the end of the rod. The mass of the point mass M is same as the mass of the rigid body model. The rod has a length l, which is same as the distance between the base and the center of mass of the rigid body model.

Figure 2(a) shows a character and the rigid body model of the character. When we consider a walking character, the point of application of floor reaction force is a certain point on the neighboring floor to the rigid body, because characters move their legs when walking. Therefore, it is possible for the point to be such as Figure 2(a)-β, not limited to the contact point between the rigid body and the floor(Figure 2(a)-α).

A human walks keeping balance by controlling floor reaction force and center of pressure of the force. We apply the same control to the character. The floor reaction force consists of normal force and friction force. When the rigid body contacts to the floor, the rigid body is subjected to the normal force from the floor $\boldsymbol{f_N}(f_{Nx}, f_{Ny}, f_{Nz})$. The controller applies friction force $\boldsymbol{f_F}(f_{Fx}, f_{Fy}, f_{Fz})$ between the foot of the character and the floor which depends on the motion of the foot, and torque τ corresponds to the difference between supposed center of pressure and contact point of the rigid body and the floor. Where Z-coordinate value of point α and β in Figure 2 is given as α_z, β_z, then τ is calculated as $\tau = (\beta_z - \alpha_z) f_N$.

In our method, however, a constraint for β_z is given instead of an actual value of β_z. That is, β_z is in the neighborhood of the rigid body. Corresponding to this constraint of β_z, τ is also constrained as $\tau_{min} < \tau < \tau_{max}$. This constraint corresponds to a movable range for the center of pressure, that is, a reachable range for the legs. Also $|\boldsymbol{f_F}|$ is constrained as $f_{Fmin} < |\boldsymbol{f_F}| < f_{Fmax}$, corresponding to friction coefficient of the floor and maximum horizontal force which can be generated by the legs of the character. Those constraints are determined from individual character settings and environment settings (e.g. friction coefficient of the floor).

Then we derive equations of motion for the model with the rigid body space defined in section 4.1. We consider rigid body motion projected into the plane defined by $\boldsymbol{e}_{xc}$, $\boldsymbol{e}_{yc}$, and the plane defined by $\boldsymbol{e}_{yc}$, $\boldsymbol{e}_{zc}$. We discuss only the motion projected into the plane defined by $\boldsymbol{e}_{yc}$, $\boldsymbol{e}_{zc}$, because these two projected motions are equivalent. With the moment of inertia I_x of the rigid body about the axis $\boldsymbol{e}_{xc}$ the equations of motion become respectively :

$$M\ddot{r}_z = f_{Fz} \tag{1}$$

$$M\ddot{r}_y = f_{Ny} - Mg \tag{2}$$

$$I_x\ddot{\theta}_x = l\begin{pmatrix}\sin\theta_x \\ -\cos\theta_x\end{pmatrix} \times \begin{pmatrix} f_{Fz} \\ f_{Ny}\end{pmatrix} + \tau \tag{3}$$

After linearization with $\theta_x \approx 0$ and simplification, the equations of motion become :

$$M\ddot{r}_z = f_{Fz} \tag{4}$$

$$I_x\ddot{\theta}_x = Mgl\theta_x + Ml\ddot{r}_z + \tau \tag{5}$$

To reach character position, we control character velocity $\dot{r}_z$. If we define the target orientation θ_{x0} and velocity $\dot{r}_{z0}$, we can reach this target by controlling f_{Fz} and τ. Such control is possible with a state feedback controller if $\theta_x \approx 0$. When the character falls down (i.e. $\theta_x \gg 0$), we use PD control of f_{Fz}

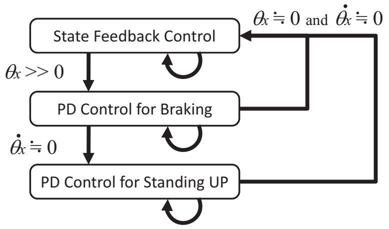


Figure 3: *State Machine to Select Control Policy*

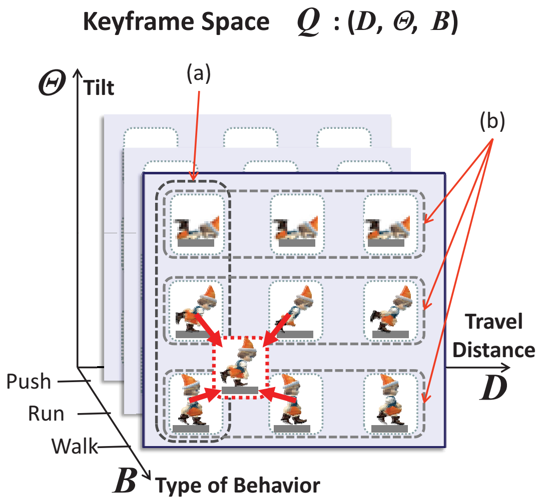


Figure 4: *Multidimensional Keyframe Set*

and τ to put a brake on the character and to stand up after the character stopping. We defined the state machine in Figure 3 to select the control policy.

5. Animation with a Multi Dimensional Key-frame Set

In our method, motions caused by internal forces in the mass system of a character are generated with keyframe animations.Concretely, from the perspective of dynamics, each joint of a character must generate torque to move its own arms or legs and to change its pose. However, the forces generating joint torques are internal forces in the mass system of the character. Those internal forces cannot be simulated with a single rigid body model simulation. Motions caused by those internal forces are generated with keyframe animations corresponding to the state of rigid body model.

We propose a multidimensional keyframe set: a set of keyframes in multidimensional space, whereas a usual keyframe animation is a set of keyframes on a 1-dimensional time axis. Such multidimensional space is also used in the spatial keyframing [IMH05]. In our method, this allows correspondence of a determined character pose not only in time space but also in the rigid body space.

We describe an example of the multidimensional keyframe space and a keyframe set, correspondence function, and typical design procedure in the following sections.

5.1. Keyframe Space and Multidimensional Keyframe Set Definition

The keyframe space is defined based on the dimensions of the rigid body space, or independent of the rigid body space and the correspondence between the two spaces is defined by a transformation function. In correspondence to the rigid body space, we define a multidimensional space with dimensions representing tilt of the character, (Θ_x, Θ_z), total travelled distance of the character, (D_x, D_z). In correspondence to the character AI, the space also has a dimension representing type of behaviors of the character, B. Finally, we define the keyframe space Q as $\{\Theta_x, \Theta_z, D_x, D_z, B\}$.

We define a multidimensional keyframe set as the set of keyframes in the keyframe space. Determination of a rigid body state by the physical simulator defines a point in the rigid body space. Then the point is mapped into the keyframe space, and the character pose is achieved by interpolating nearby keyframes to the point in keyframe space. Finally, the character motion is achieved by displaying the interpolated pose in the local frame of the rigid body model, S_r.

For example, the keyframe set given in Figure 4 represents walking motion and falling down motion. When the rigid body model rotates vertically while changing its position horizontally, the resulting motion of the character becomes a combination of both walking and falling down motions.

5.2. The Correspondence Function

We must define a function which transforms the rigid body space into the keyframe space. The simplest way is to define identical functions for each corresponding dimension. We define a correspondence between (Θ_x, Θ_z) and (θ_x, θ_z) as identical. For cyclic motions like walking, we use a periodic function which corresponds total travel distance of a character into the dimension of the keyframe space representing the phase of walking such as $\{(D_x, D_z) | D_x, D_z \in (0,1)\}$. Total travel distance of the character in front-back direction and in sideways direction is defined as equation 6 and 7. The $\boldsymbol{e}_{xc}(t)$, $\boldsymbol{e}_{zc}(t)$ are basis vectors at time t mentioned in section 4.2, $\dot{\boldsymbol{r}}_{xz}(t)$ is a velocity of the character at time t projected into the plane defined with $\boldsymbol{e}_{xc}(t)$,$\boldsymbol{e}_{zc}(t)$. t_{now} is current time.

$$D_{front} = \int_{t=0}^{t_{now}} \boldsymbol{e}_{zc}(t) \cdot \dot{\boldsymbol{r}}_{xz}(t) dt \tag{6}$$

$$D_{side} = \int_{t=0}^{t_{now}} \boldsymbol{e}_{xc}(t) \cdot \dot{\boldsymbol{r}}_{xz}(t) dt \tag{7}$$

Then we define a periodic function as equation 8, and we achieve transformation into the keyframe space as equation

9.

$$p(x) := x - \lfloor x \rfloor \quad (8)$$

$$(D_x, D_z) = (p(D_{side}), p(D_{front})) \quad (9)$$

As for the dimension B, each value in this dimension corresponds to a type of behavior, determined by the character AI. This represent changes in the behavior caused by an intention of the character. This is similar to how animators usually prepare several animations, each describing different behaviors. All values in this dimension are considered as adjacent during interpolation.

5.3. Typical Design Procedure of Multidimensional Keyframe Sets

A multidimensional keyframe set is achieved with a combination of several 1-dimensional keyframe animations. This animations have to be built with each keyframe to its correct position in the keyframe space. Furthermore, because the final animation is displayed in the coordinate system S_r, each keyframe has to be made in this coordinate system.

Just as in traditional keyframe animation, checking output motions and adjustment of keyframe sets by animators is necessary to achieve natural motion of the characters. After preparing roughly made keyframe sets, the motion generation system is activated and characters are interacted with as expected. Animators can add or adjust keyframes to refine the motion. Doing so, animators can use their traditional knowledge to attain desired motions.

6. Evaluation

We evaluated the effectiveness of the proposed method. We implemented and configured a motion generation system with the proposed method. Using the system, we generated character motion for repeated random impact forces expected to be seen during interaction with users.

Figure 5 shows some examples of generated motion. The system successfully generated approximately 30 totally different patterns of motions. In addition even two motion of the same pattern are quite different. It is important to note that these motions are generated with only 6 one-dimensional keyframe animations. To confirm the effectiveness of the proposed method, we showed these 30 patterns of motions to some animators. They say, it will take about 15 hours to create these patterns with conventional keyframe animation from scratch, while it takes 1.5 hours to create the multidimensional keyframe for the proposed method using the same tools to create keyframe animations.

Figure 5: *Examples of Generated Motion*

For whole body physical reaction motion, the character walks quickly in the same direction as the impact force to recover attitude. When the point of application of the impact force is slanted from the center of the body trunk, the character pivots on a single foot horizontally. In addition, independently of the timing, the system generated smooth motion transition. This is important for physical interaction because it is possible to apply various and unpredictable forces.

These generated motions also reflect specified features of motion in a prepared keyframe set. For example, a character folds the body trunk and finally sits down when he falls backward. The character also flutters his hands when he falls laterally. On the other hand, our method cannot generate motions in which only the upper body moves after the impact force or motions which change bending angle according to the strength of the force.

To validate physical reality of generated motion, we also conducted a subjective experiment. We describe the experimental configuration and the experiment itself in the following sections.

6.1. Experimental Configuration

The system was run on a Laptop PC with an Intel(R) Core2Duo T7300 Processor, Windows XP. Microsoft Visual C++ 7.1 for development, and DirectX 9.0 for graphics. We employed Springhead [HS04], which is a rigid body dynamics simulator with a penalty method based collision engine, working in real time on an average PC.

A character rigid body model is shown in Figure 6. The model is $1.5[m]$ tall, weighs $40.0[kg]$ and has an inertia tensor of $10.0\boldsymbol{I}[m^2 \cdot kg]$ where $\boldsymbol{I}$ is a unit matrix. We used a narrow shape for the bottom, allowing the character to fall down more easily in the front-back direction. We also made the side and top of the model convex to ensure the character lands face up or face down when it falls, instead of taking a transverse or inverted pose. This contributed to a reduction in standing up motion after falling down.

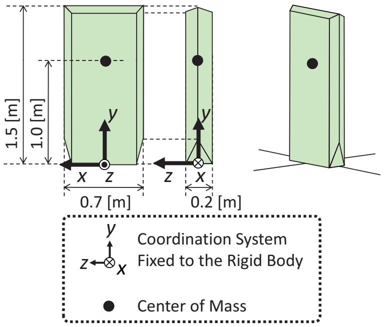


Figure 6: *Rigid Body Model of Character*

Figure 7: *Graphic Model of Character*

We used a polygon model shown in Figure 7 for the animation. We defined the keyframe space as equation 10.

$$
\begin{aligned}
Q = \{&(D_x, D_z, \Theta_x, \Theta_z, B) | D_x, D_z \in [0,1],\\
&\Theta_x \in [-\pi, \pi], \Theta_z \in [-\frac{\pi}{2}, \frac{\pi}{2}]\}\\
&B \in \{\text{"}\textit{Walking and Falling down}\text{"}, \text{"}\textit{Standing up}\text{"}\}
\end{aligned} \tag{10}
$$

Then we prepared a keyframe set shown in Figure 8 consisting of 12 1-dimensional keyframe animations, totaling 57 keyframes. For the $B =$ "Walking and Falling down" subspace (Figure 8(a)), all keyframes are put in the ($\Theta_z = 0, D_x = 0$) plane or ($\Theta_x = 0, D_z = 0$) plane. For the $B =$ "Standing Up" subspace (Figure 8(b)), all keyframes are put on ($\Theta_z = 0, D_x = 0, D_z = 0$) line because the character AI decides to stop standing up if the character falls down in a sideways direction (i.e. $\Theta_z \neq 0$) or if the character translates horizontally (i.e. $D_x, D_z \neq 0$).

6.2. Experiment

We evaluated dynamically correctness for generated motions of the system. When we watch some motions of creatures reacting to certain causal forces, we can make rough estimates of the forces. We conducted subjective experimentation to verify if the subjects can make rough estimates of the causal forces by only watching the reaction motions generated by the system. First, we showed the subject a generated motion reacting to a randomly selected impact force. The subject could watch the motion repeatedly. Then, the subject was asked to choose the impact force which caused the motion.

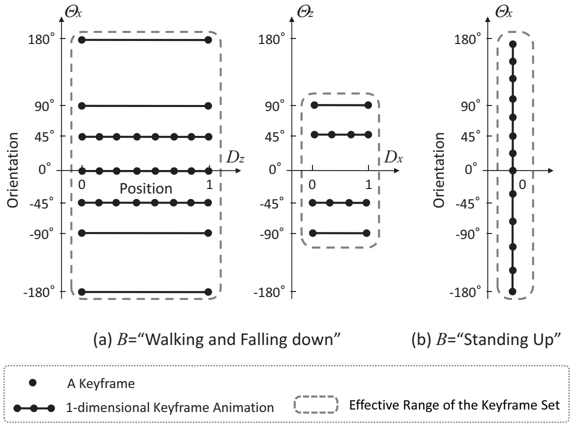


Figure 8: *Keyframe Sets used for Evaluation*

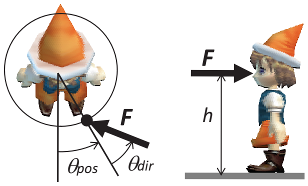


Figure 9: *Applied Forces in the Experimentation*

We applied the force $\boldsymbol{F}$ as shown in Figure 9. The θ_{pos} and θ_{dir} range $-45°$, $0°$ and $45°$, h ranges 0.0, 0.5 and $1.0[m]$. Strength of the force $|\boldsymbol{F}|$ has two different patterns, "strong" and "weak". The force was randomly selected from combinations of these parameters, and we showed the subject generated motion after the force was applied once. The subjects were asked to choose the parameters of impact force which caused the motion. 6 men in their twenties participated in the experiment, and each subject tried 20 times.

6.3. Result

First, we calculated distance between the chosen answer and the correct answer. Then we calculated the accuracy rate, or the rate of answers which had a distance of zero, for each parameter. We used absolute angle values for the direction of the force, $\theta_{dir} + \theta_{pos}$, instead of θ_{dir} itself. After the normalization with minimum interval of choices for each parameter, we also calculated average μ and standard deviation σ of the distances.

Table 1: *Result of the Impact Force Estimation Task*

	θ_{pos}	$\theta_{dir} + \theta_{pos}$	h	$\lvert\boldsymbol{F}\rvert$
Accuracy Rate	0.73	0.78	0.54	0.88
μ	0.017	-0.099	-0.008	0.041
σ	0.545	0.488	0.766	0.350

Table 1 shows the results. For parameters except the height *h* of the position where the force applied, the accuracy rate is over 0.7. The error was distributed equally and narrowly according to the value of μ and σ, i.e. most errors occurred in neighboring choices. Despite the accuracy rate of the parameter *h* being relatively low, most parameters were accurately guessed by viewing the generated reaction motion of the character. Therefore, we consider the characters motion generated with the proposed method to look physically correct.

7. Conclusion

The proposed method is able to generate various and dynamically correct reaction motions for various forces repeatedly applied to the character. The method is especially effective at generating physical reaction motion with the entire body of the character. For example, if it is used for a character walking around freely in a 3D field, the system will generate physically realistic motions such as locomotion with entire body (walking, running, jumping etc.) or generating and reacting to force with the entire body (crashing, pushing etc.).

Motions generated with the method can be designed with prepared keyframe sets, which can be created using traditional animator skills. Drastic reduction in preparation of keyframe animation is achieved because various motions are automatically generated from a few prepared keyframe sets. A single rigid body model used as a simulation model of a character has few parameters to be adjusted and is easy to control without advanced knowledge of control systems. Since a single rigid body character model needs low computational cost, the method is also effective at applying physics on a great number of characters, such as background mobs. This will contribute to raising the level of reality for the entire virtual world.

On the other hand, the proposed method is not applicable if the effect on body parts, aside from the body trunk, cannot be neglected. Using articulated body model is a possible solution. We must separate bare essentials of the rigid body which are considered as non-negligible for the desired interaction. For example, if the arms of the character need to be moved freely, we must use an articulated body with a body trunk and arms. More varied interactions with characters will be achieved with a relaxation in the number of rigid body parts. Nevertheless increasing the number of body parts may increase computational cost.

One of the possible extensions of our method is to enable more than one behavior dimension of the keyframe space to the character AI. Currently, the AI uses only one dimension of the keyframe space for behavior, while physical simulation uses 4 dimensions of the keyframe space. If we assign more than one dimension of the keyframe space to the AI, such as strength of a certain emotion of the character, various motions reflecting the character AI will be generated automatically. In this way, a multidimensional keyframe set will be useful to describe detailed motion in combination with rough simulation of characters physically and even emotionally.

Acknowledgements

We would like to thank Charles DaSalla for helpful comments and English review, Takashi Toyama for providing the graphics model of the character, and the anonymous reviewers for valuable comments.

References

[AFO05] ARIKAN O., FORSYTH D. A., O'BRIEN J. F.: Pushing people around. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), ACM, pp. 59–66.

[ALP04] ABE Y., LIU C. K., POPOVIĆ Z.: Momentum-based parameterization of dynamic character motion. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2004), Eurographics Association, pp. 173–182.

[BW95] BRUDERLIN A., WILLIAMS L.: Motion signal processing. In *SIGGRAPH '95: Proc. of the 22nd annual conf. on Computer graphics and interactive techniques* (New York, NY, USA, 1995), ACM Press, pp. 97–104.

[CHP07] COOPER S., HERTZMANN A., POPOVIĆ Z.: Active learning for real-time motion controllers. *ACM Trans. Graph. 26*, 3 (2007), 5.

[dSAP08] DA SILVA M., ABE Y., POPOVIĆ J.: Interactive simulation of stylized human locomotion. *ACM Trans. Graph. 27*, 3 (2008), 1–10.

[FvdPT01] FALOUTSOS P., VAN DE PANNE M., TERZOPOULOS D.: Composable controllers for physics-based character animation. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM, pp. 251–260.

[Gle98] GLEICHER M.: Retargetting motion to new characters. In *SIGGRAPH '98: Proc. of the 25th annual conf. on Computer graphics and interactive techniques* (New York, NY, USA, 1998), ACM Press, pp. 33–42.

[HS04] HASEGAWA S., SATO M.: Real-time Rigid Body Simulation for Haptic Interactions Based on Contact Volume of Polygonal Objects. *Computer Graphics Forum 23*, 3 (2004), 529–538. http://springhead.info/.

[HTH05] HASEGAWA S., TOSHIAKI I., HASHIMOTO N.: Human scale haptic interaction with a reactive virtual human in a realtime physics simulator. In *ACE '05: Proc. of the 2005 ACM SIGCHI Intl. Conf. on Advances in computer entertainment technology* (New York, NY, USA, 2005), ACM Press, pp. 149–155.

[HWBO95] HODGINS J. K., WOOTEN W. L., BROGAN D. C., O'BRIEN J. F.: Animating human athletics. In *SIGGRAPH '95: Proc. of the 22nd annual conf. on Computer graphics and interactive techniques* (New York, NY, USA, 1995), ACM Press, pp. 71–78.

[IAF07] IKEMOTO L., ARIKAN O., FORSYTH D.: Quick transitions with cached multi-way blends. In *I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2007), ACM, pp. 145–151.

[IMH05] IGARASHI T., MOSCOVICH T., HUGHES J. F.: Spatial keyframing for performance-driven animation. In *SCA '05: Proc. of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), ACM Press, pp. 107–115.

[KG04] KOVAR L., GLEICHER M.: Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph. 23*, 3 (2004), 559–568.

[KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion graphs. In *SIGGRAPH '02: Proc. of the 29th annual conf. on Computer graphics and interactive techniques* (New York, NY, USA, 2002), ACM Press, pp. 473–482.

[KLKK05] KOMURA T., LEUNG H., KUDOH S., KUFFNER J.: A feedback controller for biped humanoids that can counteract large perturbations during gait. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (April 2005), pp. 1989–1995.

[KSK00] KOMURA T., SHINAGAWA Y., KUNII T.: Creating and retargetting motion by the musculoskeletal human body model. *The Visual Computer 16*, 5 (2000), 254–270.

[LvdPF96] LASZLO J., VAN DE PANNE M., FIUME E.: Limit cycle control and its application to the animation of balancing and walking. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1996), ACM, pp. 155–162.

[OM01] OSHITA M., MAKINOUCHI A.: A Dynamic Motion Control Technique for Human-like Articulated Figures. *Computer Graphics Forum 20*, 3 (2001), 192–203.

[PB02] PULLEN K., BREGLER C.: Motion capture assisted animation: texturing and synthesis. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2002), ACM, pp. 501–508.

[PW99] POPOVIĆ Z., WITKIN A.: Physically based motion transformation. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 11–20.

[RCB98] ROSE C., COHEN M. F., BODENHEIMER B.: Verbs and adverbs: Multidimensional motion interpolation. *IEEE Comput. Graph. Appl. 18*, 5 (1998), 32–40.

[RGBC96] ROSE C., GUENTER B., BODENHEIMER B., COHEN M. F.: Efficient generation of motion transitions using spacetime constraints. In *SIGGRAPH '96: Proc. of the 23rd annual conf. on Computer graphics and interactive techniques* (New York, NY, USA, 1996), ACM Press, pp. 147–154.

[SHP04] SAFONOVA A., HODGINS J. K., POLLARD N. S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph. 23*, 3 (2004), 514–521.

[SK91] SHUUJI KAJITA K. T.: Study of Dynamic Biped Locomotion on Rugged Terrain -Derivation and Application of the Linear Inverted Pendulum Model. In *Proceedings of the IEEE International Conference on Robotics and Automation* (1991), pp. 1405–1411.

[SPF03] SHAPIRO A., PIGHIN F., FALOUTSOS P.: Hybrid control for interactive character animation. In *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 2003), IEEE Computer Society, p. 455.

[USS*] UEDA F., SUGIYAMA H., SEKI T., TANAKA M., NISHIKAWA Z.: The making of "shadow of the colossus". http://www.team-ico.net/sotc/index.html, http://www.watch.impress.co.jp/game/docs/20051207 /3dwa.htm.

[WH97] WILEY D. J., HAHN J. K.: Interpolation synthesis of articulated figure motion. *IEEE Comput. Graph. Appl. 17*, 6 (1997), 39–45.

[WJM06] WROTEK P., JENKINS O. C., MCGUIRE M.: Dynamo: dynamic, data-driven character control with adjustable balance. In *Sandbox '06: Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames* (New York, NY, USA, 2006), ACM, pp. 61–70.

[YLvdP07] YIN K., LOKEN K., VAN DE PANNE M.: Simbicon: simple biped locomotion control. *ACM Trans. Graph. 26*, 3 (2007), 105.

[YPvdP05] YIN K., PAI D. K., VAN DE PANNE M.: Data-driven interactive balancing behaviors. In *PG '05: Proceedings of the 13th Pacific Conference on Computer Graphics and Applications* (2005), pp. 118–121.

[ZH99] ZORDAN V. B., HODGINS J. K.: Tracking and modifying upper-body human motion data with dynamic simulation. In *Proc. of Eurographics Workshop on Animation and Simulation '99* (1999).

[ZH02] ZORDAN V. B., HODGINS J. K.: Motion capture-driven simulations that hit and react. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2002), ACM, pp. 89–96.

[ZMCF05] ZORDAN V. B., MAJKOWSKA A., CHIU B., FAST M.: Dynamic response for motion capture animation. *ACM Trans. Graph. 24*, 3 (2005), 697–701.