

## Creating and retargetting motion by the musculoskeletal human body model

Taku Komura<sup>1</sup>,  
Yoshihisa Shinagawa<sup>1</sup>,  
Tosiyasu L. Kunii<sup>2</sup>

<sup>1</sup>Department of Information Science, Graduate School of Science, University of Tokyo, 7-3-1 Hongo, Bunkyo-Ku, Tokyo, Japan 113-0033

<sup>2</sup>Hosei University, Computing Science Research Center, 3-7-2 Kajino-chou Koganei City, Tokyo, Japan 184-8584  
e-mail: kohmura@is.s.u-tokyo.ac.jp,  
sinagawa@is.s.u-tokyo.ac.jp, kunii@k.hosei.ac.jp

Recently, optimization has been used in various ways to interpolate or retarget human body motions obtained by motion-capturing systems. However, in such cases, the inner structure of a human body has rarely been taken into account, and hence there have been difficulties in simulating physiological effects such as fatigue or injuries. In this paper, we propose a method to create/retarget human body motions using a musculoskeletal human body model. Using our method, it is possible to create dynamically and physiologically feasible motions. Since a muscle model based on Hill's model is included in our system, it is also possible to retarget the original motion by changing muscular parameters. For example, using the muscle fatigue model, a motion where a human body gradually gets tired can be simulated. By increasing the maximal force exertable by the muscles, or decreasing it to zero, training or displacement effects of muscles can also be simulated. Our method can be used for biomechanically correct inverse kinematics, interpolation of motions, and physiological retargetting of the human body motion.

**Key words:** Human animation – Muscle-based model – Motion retargetting – Motion synthesis

*Correspondence to:* T. Komura

## 1 Introduction

In fields such as computer graphics and virtual reality, researchers have been looking for methods to create human motion data. Such data are used to control, for example, virtual actors in video games, movies, and cyber space.

The most practical method these days is to use a motion-capturing system. Optimization techniques have been developed to interpolate (Rose et al. 1996) or retarget (Gleicher 1998; Gleicher and Litwinowicz 1998; Lee and Shin 1999; Popvic and Witkin 1999) the captured motion data. Since such techniques were developed for the purpose of animation, the human body models have been simple multibody systems without any physiological structure. However, the anatomical structure of the body is very important to characterize postures and motion. Even at relaxed postures, people can acknowledge that the inner elements of the body (such as the bones, muscles, and ligaments) greatly affect the relationship among various joints. It is clear that the configuration space of each joint angle is not independent of the others. For example, curling a single finger as opposed to curling that finger in unison with other fingers is very difficult.

The torque at each joint is created by the muscles linking the bones. The configuration space and the maximum power of the body are mostly determined by the geometry and physical structure of each muscle. For further automatic and correct generation of human body motions, it is necessary to construct a human body model that contains the inner physiological structure.

In this paper, a musculotendon model of the legs based on Delp (1990) and Delp et al. (1990) is used for the human body model, and then, with this model, an algorithm to create physiologically feasible human body postures and motion is proposed.

The method is based on spacetime constraints. The method can be used not only for keyframe animation, but also for retargetting motion provided by the user. A dynamic system of the whole human body including the muscles of the entire lower extremities is presented. The physical differences of individuals, such as muscle strength and size, can be handled easily. Such a model is useful to analyze and simulate actual human motions.

Motions are calculated in the following steps:

1. A number of either keyframe postures or trajectories of the joint angles are specified by the user to determine the initial motion. If keyframe postures

are given, they are first interpolated by B-spline curves.

2. A function that estimates the amount of infeasibility of the motion at each moment is defined through the motion. This function is based on muscle dynamics.
3. We convert the motion to a physiologically feasible one by reducing the value of each infeasibility function in the motion to zero.
4. For further tuning of the motion, starting from the feasible motion calculated at step 3, the motion is optimized for an objective function specified by the user.

Our method can be used for inverse kinematics, interpolation of motions, and physiological retargetting of the human body motions. The algorithm to make trajectories feasible can be utilized for inverse kinematics. Since the structure of the muscles are taken into account, the results obtained are more natural than postures created by existing inverse kinematics systems that handle a human body like a puppet.

The fatigue and recovery model of a muscle proposed by Giat et al. (1993, 1996) is combined with our muscle models. Their model limits the maximal force exertable by each muscle according to the history of the muscle force. It is possible to simulate a motion where a human gradually gets tired by using our motion creation algorithm. It can also convert motions provided by the user to a tired motion.

Other motions, such as those by injured bodies, can also be generated by displacement of particular muscles.

We refer to the conversion of a motion to a tired one, or to an injured one, as *physiological retargetting*. Physiological retargetting is a useful tool for creating human body animation. Such a conversion can only be achieved with an anatomy-based human body model. Several examples of the retargetting are shown in Sect. 9.

## 2 Previous work

The techniques to create human animation can be divided into three groups: motion capturing, methods based on kinematics, and methods based on dynamics.

Since motion capturing requires special and expensive devices, people generally use the kinematic trajectories already prepared in databases. If a desired

datum is not found, the animation creator has to modify the kinematic data to meet the requirements.

Techniques introduced by Unuma et al. (1995) and Bruderlin and Williams (1995) decompose trajectory data in the frequency domain and amplify some features of the motion, such as depressive feelings or anger, by changing the coefficients of the basis functions.

Their methods successfully exaggerate psychological features in motion. However, since their methods are completely based on kinematics, such algorithms cannot be used for simulating anatomical or physiological phenomena in the human body.

Dynamics-based human animation has been created by many researchers such as Armstrong and Green (1985) and Wilhelms (1987). The benefit of using the dynamics-based method is the reality of the resulting animation.

The new problem that arises when controlling a human body model in a physical environment is that the animator must describe the changes in the torque and force applied to the model instead of the kinematic trajectories. It is a difficult task because the effect of changing each dynamic parameter is not obvious.

The difficulty of controlling a human model is caused by the redundancy of the body's dynamics. The redundancy is caused by the great number of joints. Each joint has one, two, or three degrees of freedom (DOFs). The total number of DOFs of a human body is more than a hundred. Even though these DOFs enable the body to avoid obstacles, singularities, and structural limitations (e.g., angle limits of the rotational joints), and to move skillfully (e.g., reaching behind an object, crawling into caves, etc), more complicated algorithms are required for controlling the system. For these reasons, a controller that calculates each torque at every joint to let the model move as desired is indispensable. There are two contemporary methods to specify the changes of the joint torques. The first way is to use a control method known as proportional derivative (PD) control in the control theory. The animator gives the system a number of postural keyframes and the joint parameters for each keyframe. Torque is applied to each joint according to the following equation:

$$\tau_i = k_p(\phi_d - \phi) - k_v(\dot{\phi}_d - \dot{\phi}),$$

where  $\phi_d$  and  $\dot{\phi}_d$  are the desired joint angle and angular velocity of the next keyframe, respectively;  $\phi$  is the current joint angle, and  $k_p$  and  $k_v$  are the gain constants that define the strength of the joint. This

approach has been used to simulate gaits (Laszlo et al. 1996; van de Panne 1996) and athletic movements (Hodgins et al. 1995; Wooten and Hodgins 1996). The gain constants are either fixed (Laszlo et al. 1996; Wooten and Hodgins 1996) or optimized (van de Panne 1996). To create human animation using this method, a number of keyframe postures and a set of gain constants for the movements between the keyframes are necessary. The problem in this method is the difficulty of choosing the appropriate gain parameters for the motion. These values depend on the motion and the size or mass of the body model. Hodgins and Pollard (1997) have developed a method to automatically tune up the gain parameters prepared for a motion to be reused by another character with different body size.

However, it is known in biomechanical areas that animals actually do not control their body by feedback control alone (Gomi and Kawato 1996). Therefore, the use of PD control does not accurately simulate the processes that the human body uses to produce motion.

The other approach is to use optimization. In biomechanical studies, various human motions have been calculated by optimization techniques such as the conventional gradient method (Pandy et al. 1990) or dynamic programming (Yamaguchi and Zajac 1990).

In computer graphics, techniques based on space and time constraints (Witkin and Kass 1988), which pose motion synthesis problems such as constrained optimization, have been developed by many researchers to create keyframe animation (Cohen 1992; Liu et al. 1994), to interpolate motions (Rose et al. 1996), and to retarget motions (Gleicher 1998; Gleicher and Litwinowicz 1998). Human animations have been created in some of these researches (Gleicher 1998; Gleicher and Litwinowicz 1998; Rose et al. 1996). However, since no anatomical structure of a real human body has been included in their models, it has been difficult to simulate phenomena that derive from the inner structure, such as fatigue or injuries.

Even though several researchers use muscle models to obtain realistic rendering of human (Scheepers et al. 1997) or animal bodies (Wilhelms and van Gelder 1997), musculoskeletal models have rarely been utilized to yield human motion data with a few exceptions of our work (Komura et al. 1997, 1999). Komura et al. (1997) propose a method to interpolate keyframe postures using a musculoskeletal model of the body. In this paper, we have extended our pre-

vious work by enabling conversion of an infeasible motion to a physiologically feasible one and by enabling simulation of fatigue and injury effects by changing muscle parameters. Komura et al. (1999) use the musculoskeletal model of the body to calculate and visualize the maximum dynamical abilities of the body.

Chen and Zeltzer (1992) create a very precise muscle model using the finite element model (FEM). It has not, however, been used for the control of human body models in a dynamic environment.

### 3 Musculoskeletal model

This paper proposes a method to generate human motion based on the anatomy and physiology of the real human body. For this purpose, a musculoskeletal human body model is necessary. The data published by Delp (1990) and Delp et al. (1990) have been used in this paper<sup>1</sup>. The data include the attachment sites of 43 muscles on each leg, physiological parameters such as the length of tendons, range of joint angles, etc. In Delp's research, several basic properties, such as the passive joint torque and maximum isometric joint torque were calculated and compared with biomechanically measured data to evaluate the validity of the model. Other necessary data, such as the weight and inertia of the body segments, were obtained from (Yamaguchi and Zajac 1990).

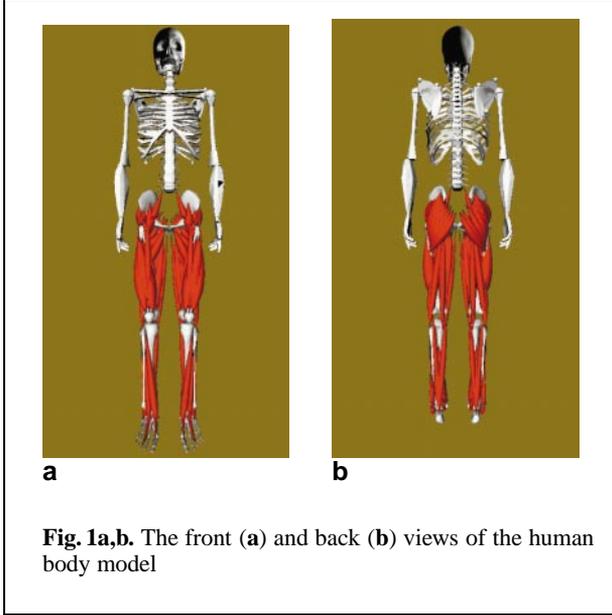
The upper half of the body is composed of the chest, head, upper arms, lower arms, and hands.

The lower half has the pelvis at the top, and each leg consists of the femur, tibia, patella, talus, calcaneus, and toes. In this research, the talus, calcaneus and toes are stuck together to form a rigid foot. Therefore, the human body model is composed of 17 segments in total.

The joints of the legs are assumed here to be either a 3-DOF gimbal joint (hip joint) or a 1-DOF joint (knee and ankle joint). No muscles are attached to the upper part of the body because the motion created here is mainly in the legs.

The front and back views of the body model with muscles are shown in Fig. 1. The muscles are rendered with generalized cylinders. The radii of the generalized cylinders are calculated with the length of the muscle and the physiological cross-sectional

<sup>1</sup> The data can be downloaded from <http://isb.ri.ccf.org/isb/data/delp>.



area (PCSA). The PCSA has been calculated with the peak isometric force data included in Delp (1990) and the scale factor of  $25 \text{ N cm}^{-2}$  from Friedrich and Brand (1990).

#### 4 Preliminaries of Hill-based muscle model

For each musculotendon, a model based on Hill's three-component model (Fig. 2) is used. There are a lot of muscle models that are derived from Hill's model (Winters 1990). The model used here is that from Delp (1990) and Delp et al. (1990). It is composed of three elements: the contractile element (CE), or muscle fibers, series elastic element (SEE), or muscle tendon, and the parallel elastic element (PEE), or connective tissue around the fibers and fiber bundles. There is an inclination between the muscle part (CE and PEE) and the tendon part (SEE). The angle between them is called the pennation angle of the muscle, and it is denoted by  $\alpha$  here. The force  $f^{ce}$  made by the CE is a function of its length  $l^{ce}$ , contraction velocity  $v^{ce}$ , and the muscle activation level  $a$ , which is controlled by the central nervous system (CNS). The muscle-tendon length ( $l^{mt}$ ) is the sum of the muscle fiber length and the tendon length:  $l^{mt} = l^m \cos \alpha + l^t$ , where  $l^m = l^{ce} = l^{pe}$ .  $l^{mt}$  can be calculated by the joint angles of each leg.

The force of the SEE ( $f^t$ ) and PEE ( $f^{pe}$ ) are functions only of their lengths ( $l^t$ ,  $l^{pe}$ ). The relationship between the force at each element is

$$f^t = (f^{ce} + f^{pe}) \cos \alpha. \quad (1)$$

The tendon is a passive element that exerts an elastic force  $f^t$  only when its length is longer than the slack length  $l_s^t$ . The relationship between the tendon strain and the normalized elastic force  $f_o^t((l^t - l_s^t)/l_s^t)$  is shown in Fig. 3.

The CE can generate the maximum force  $f_o^M$  when its length is set to the natural length  $l_o^m$ , and the contraction velocity  $v^{ce}$ , to zero. The normalized curve of  $f^{ce}/f_o^M(l^{ce}/l_o^m)$ , where  $a = 1$  and  $v^{ce} = 0$ , is shown in Fig. 4 together with the normalized curve of  $f^{pe}/f_o^M(l^{pe}/l_o^m)$ .

The force exertable by the CE decreases as the contraction velocity increase. The curve of  $f^{ce}/f_o^M$  and  $v^{ce}$ , where  $a = 1$  and  $l^{ce} = l_o^m$ , is shown in Fig. 5.  $v^o$  is the maximum contraction velocity of the CE, and it is assumed to be  $10l_o^m/s$  (Pandy et al. 1990). This curve is defined here as  $g^{ce}(v^{ce}/v^o)$ . Using  $f_o^{ce}$ ,  $f_o^{pe}$ ,  $f_o^t$  and  $g^{ce}$ , we obtain

$$f^{ce}(l^m, v^m, a) = f_o^M \cdot f_o^{ce}(l^m/l_o^m) \cdot g^{ce}(v^m/v^o) \cdot a \quad (2)$$

$$f^t(l^t) = f_o^M \cdot f^t((l^t - l_s^t)/l_s^t) \quad (3)$$

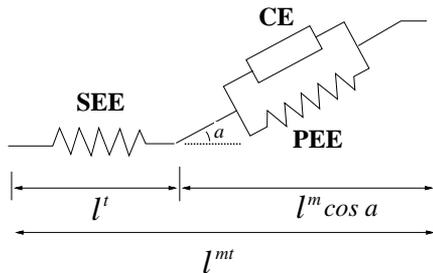
$$f^{pe}(l^m) = f_o^M \cdot f_o^{pe}(l^m/l_o^m). \quad (4)$$

Let us now consider the problem of calculating the maximal and minimal amounts of force that can be exerted by the Hill-based musculotendon when the length  $l^{mt}$  is known. In order to solve this problem, first  $v^m (= \frac{dl^m}{dt})$  must be known. If the body is not moving, the length of each element remains the same, and hence  $v^m = 0$ . However, if the legs are moving,  $v^m$  should be approximated by a finite difference:

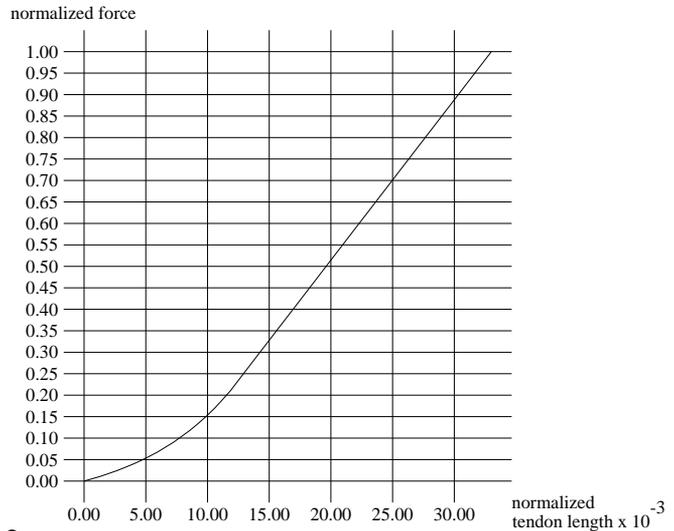
$$v^m = \frac{l^m - l_{prev}^m}{\Delta t}, \quad (5)$$

where  $l_{prev}^m$  is the length of  $l^m$  at the previous time step, and  $\Delta t$  is the time step. Suppose  $l_{prev}^m$  is known. From these equations we can derive:

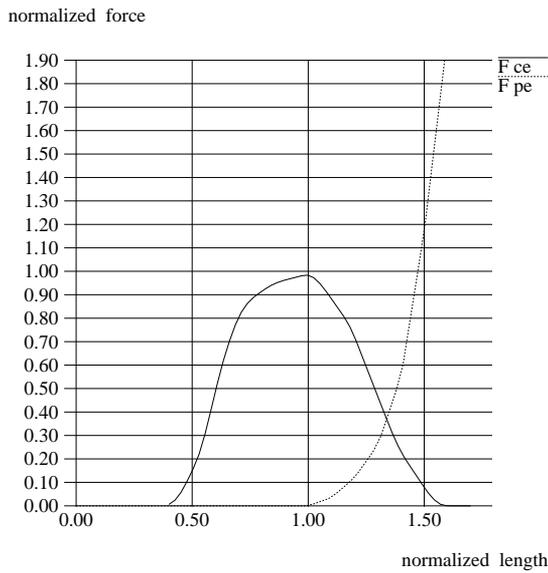
$$\begin{aligned} f^t((l^{mt} - l^m \cos \alpha - l_s^t)/l_s^t) \\ - \{f^{ce}(l^m/l_o^m) \cdot g^{ce}((l^m - l_{prev}^m)/\Delta t)\} \cdot a \\ + f^{pe}(l^m/l_o^m) \} \cos \alpha = 0. \end{aligned} \quad (6)$$



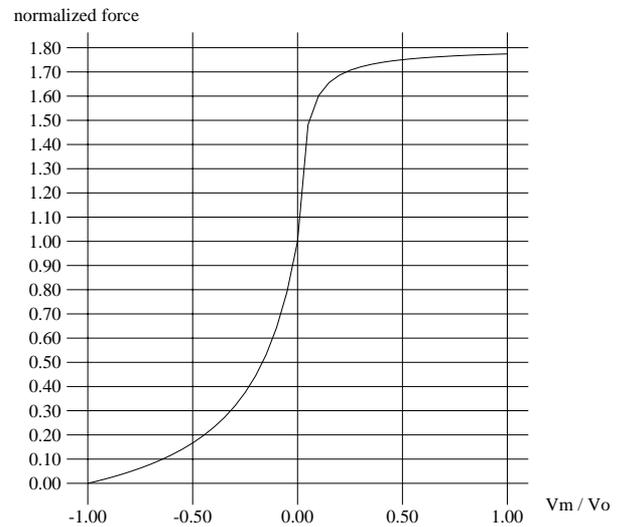
2



3



4



5

**Fig. 2.** The Hill-based muscle model used in this research

**Fig. 3.** The relationship between the tendon strain  $(l^t - l_s^t)/l_s^t$  and the normalized force  $f^t/f_o^M$ . The plotted data are taken from Delp (1990)

**Fig. 4.** The force-length curve of the active and passive muscle element  $f^{ce}$  and its length  $l^{ce}$ ,  $f^{pe}$  and  $l^{pe}$ . The plotted data are taken from Delp (1990)

**Fig. 5.** The velocity-force curve of the contractile element when  $l_m = l_o^m$  and  $a = 1$ .  $v^o$  is the maximal contraction velocity, which is assumed to be  $10l_o^m/s$  (Pandy et al. 1990). The plotted data are taken from Delp (1990)

When the muscle exerts the maximum force, the activation level  $a$  is 1, and when the muscle exerts the minimum force,  $a$  is 0.

As the activation level  $a$  is specified, the only unknown variable in (6) is  $l^m$ .  $l^m$  can be solved numerically from this equation. As  $l^m$  at  $a = 1$  and  $a = 0$  is known, the maximum and minimum musculotendon force  $f^{max}$  and  $f^{min}$  can be computed. The musculotendon force  $f^t$  at each moment is limited by the following equation:

$$f^{min}(a = 0) \leq f^t \leq f^{max}(a = 1). \tag{7}$$

## 5 Calculation of muscle force

In this section, the procedure to obtain the muscle force during the motion is explained. First, joint torque is calculated with inverse dynamics from the motion. Joint torque is then decomposed to muscle force.

### 5.1 Inverse dynamics

The force and torque made at each joint can be calculated with the generalized coordinate parameters of the body. This operation is called inverse dynamics. We have used a commercial software package SD/FAST<sup>2</sup>, which uses Kane’s method for the calculation.

### 5.2 Solution of the closed loop problem

In the double support phase where a human stands on two feet, the torque made by the legs are redundant, and hence cannot be calculated with inverse dynamics. For this reason, we distributed the force and torque to both legs proportionally to the inverse of the distance between the feet and the position where the body’s center of mass is projected to the ground (Ko and Badler 1996; Komura et al. 1997; Vukobratovi 1990).

### 5.3 Prediction of the muscle force

The method to predict the muscle force from the joint torque and the profile of the motion is explained here.

<sup>2</sup> See <http://www.syndym.com> for more information.

The torque  $\tau_i$  made at joint  $i$ , is generated by the muscles crossing the joint:

$$\tau_i = \sum_j r_j \times f_j, \tag{8}$$

where  $r_j$  and  $f_j$  are the moment arm and the force exerted by muscle  $j$ , respectively, and  $\times$  represents the outer product.

The joint torque can be calculated with inverse dynamics if the joint angles  $\theta$ , angular velocities  $\dot{\theta}$ , and angular acceleration  $\ddot{\theta}$  are specified.

However, since the number of muscles crossing joint  $i$  is always greater than the DOF of the joint, solving  $f_j$  in (8) is a redundant problem. We apply an optimization method to determine the muscle force (Crowninshield and Brand 1981; Komura et al. 1997). A criterion

$$u = \sum_{i=1}^{n_m} \left( \frac{f_i^t}{A_i} \right)^2, \tag{9}$$

where  $f_i^t = |f_i|$ ,  $A_i$  is the PCSA of muscle  $i$ , and  $n_m$  is the number of muscles is optimized.

We use (8) as equality constraints and (7) as inequality constraints, so that  $u$  can be minimized by quadratic programming. The muscle force  $f^t$  at this moment is obtained at the same time. However, we do not use the upper limit  $f^t \leq f^{max}$  here. This constraint is used later in the following conversion and optimization stage to obtain feasible motions.

We use  $f^t$  and  $l^{mt}$  to calculate  $l^m$ .  $l^m$  is used as  $l_{prev}^m$  in the next stage to calculate  $v^m$  by finite differentiation, and then  $f^{min}$  and  $f^{max}$  can be calculated again at the next stage. By forward repetition of this calculation, it is possible to calculate  $f^{min}$  and  $f^{max}$  at any stage during the motion.

### 5.4 Balance

To keep the balance of the human body model, it is necessary to define a function that evaluates the stability of the posture. The zero moment point (ZMP) can be used to define such a function of stability. When a human stands on one foot or on both feet, a point exists where the torque applied to the body from the ground becomes zero. When the body is supported by a single leg, this point is at the sole of the support foot, while, when the body is supported by both legs, it stays in the area surrounded by the feet (Vukobratovi et al. 1990). Since there is no joint

between each foot and the ground, the torque that can be generated between the sole and the ground is limited. If the torque exceeds the limit, the body will fall down to the ground. One way to judge whether the motion is valid is to calculate the ZMP and check if it is within the supporting area.

Suppose  $\boldsymbol{\tau}_g$  and  $\boldsymbol{f}_g$  are the torque and force applied by the ground to the body, and  $\boldsymbol{r}_g$  is the vector from the center of mass of the body to ZMP. Then, the relationship between these vectors can be listed by

$$\boldsymbol{\tau}_g = \boldsymbol{r}_g \times \boldsymbol{f}_g. \quad (10)$$

As the elements of these vectors are listed as  $\boldsymbol{\tau}_g = (\tau_x, \tau_y, \tau_z)$ ,  $\boldsymbol{r}_g = (r_x, r_y, r_z)$ , and  $\boldsymbol{f}_g = (f_{gx}, f_{gy}, f_{gz})$ , we have

$$\tau_x = r_y f_{gz} - r_z f_{gy}, \quad (11)$$

$$\tau_y = r_z f_{gx} - r_x f_{gz}, \quad \text{and} \quad (12)$$

$$\tau_z = r_x f_{gy} - r_y f_{gx}. \quad (13)$$

The  $y$  axis is parallel to the vertical direction here. It is assumed here that the static frictional constant between the force and the ground is infinite. This means only the moment around the  $x$  and  $z$  axes of the floor must be checked to keep the body stable.

$r_y$  is already known from the posture (the height of the center of mass of the body). Using the first two equations, we can calculate  $r_x$  and  $r_z$ :

$$r_x = \frac{\tau_z + r_y f_{gz}}{f_{gy}} \quad (14)$$

$$r_z = \frac{-\tau_x + r_y f_{gx}}{f_{gy}}. \quad (15)$$

If the ZMP is in the support area, the posture is stable.

If the ZMP is outside the area, however, some additional torque must be added to the support foot to prevent the body from falling. The value of the additional torque is used in this paper to evaluate the postural stability:

$$s(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \begin{cases} |\tau_+| & (\text{the ZMP is out of the support area}) \\ 0 & (\text{the ZMP is in the support area}), \end{cases} \quad (16)$$

where  $\tau_+$  is the minimum external torque that must be added to the support foot to realize the specified motion.

## 6 The fatigue model of the muscles

When a muscle exerts a large amount of force, the fast glycolytic (FG) fibers in the muscle are recruited. This causes the intracellular pH level inside the muscle to decline, and then the maximum amount of force exertable by the contractile element decreases. This is called the *fatigue phase*. When the muscle is not used, the pH level increases, and the exertable force increases, which is called the *recovery phase*.

Giat et al. (1993) observe the intracellular pH level inside the electronically stimulated quadriceps muscle using  $^{31}\text{P}$  nuclear magnetic resonance spectroscopy and obtain the relationship between the intracellular pH and force exerted by the muscle.

The decay of the pH level during the fatigue phase with time  $t$  is calculated by

$$pH^F(t) = c_1 - c_2 \tanh[c_3(t - c_4)] \quad (17)$$

with constant parameters  $c_1$ ,  $c_2$ ,  $c_3$ , and  $c_4$ .

The pH during the recovery phase is similarly calculated by

$$pH^R(t) = d_1 + d_2 \tanh[d_3(t - d_4)] \quad (18)$$

with constant parameters  $d_1$ ,  $d_2$ ,  $d_3$ , and  $d_4$ . The force output is fitted by the following function:

$$f_{pH}(pH) = d_5[1 - e^{d_6(pH - d_7)}], \quad (19)$$

where  $d_5$ ,  $d_6$  and  $d_7$  are constant values. Equation 19 is normalized by the force obtained at the beginning of the experiment:

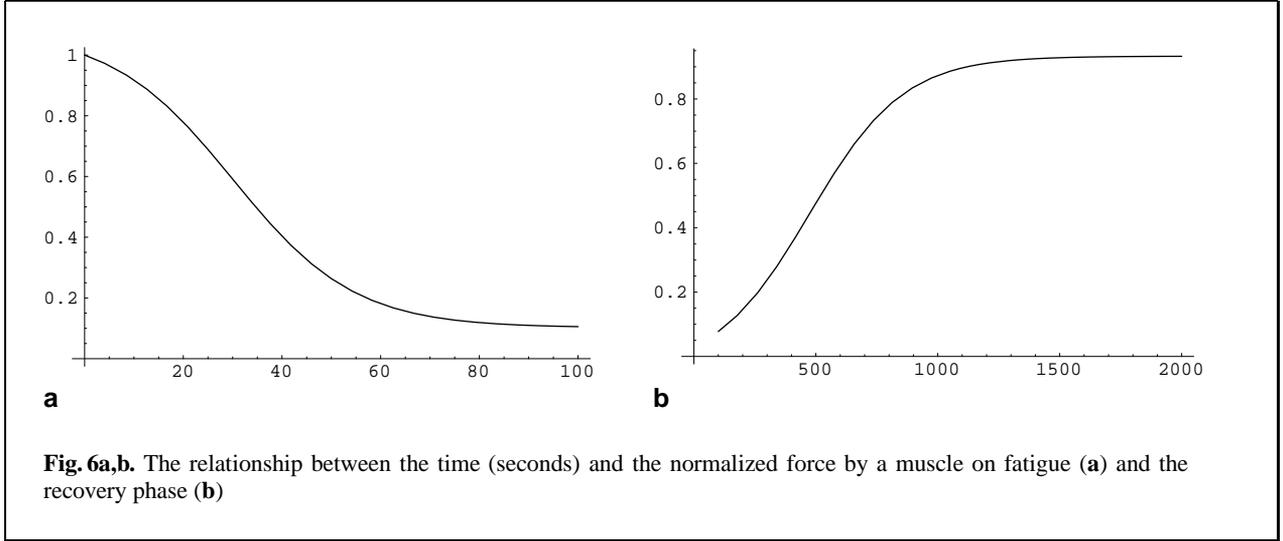
$$f_{pH}^N(pH) = \frac{f_{pH}(pH(t))}{f_{pH}(pH(t_0))}, \quad (20)$$

where  $0 < f_{pH}^N < 1$ . The values of  $c_1, \dots, c_4, d_1, \dots, d_7$  are described in Giat et al. (1993,1996).

The normalized force-pH function  $f_{pH}^N(pH)$  is combined with (2), to compose a new dynamic equation of the CE component:

$$f^{ce} = f^{ce}(l^{ce}, v^{ce}, a) \cdot f_{pH}^N. \quad (21)$$

The decay and increase of the normalized force during the fatigue and recovery phase is shown in Fig. 6. The phase of each muscle is switched between fatigue and recovery according to its activation level (Komura et al. 1999). The threshold is set to 0.5. We have used this fatigue model to simulate motions that gradually tire a human body.



## 7 Conversion of user-specified motion to a physiologically feasible motion

The motions specified by the user are not necessarily biomechanically correct and feasible for the musculoskeletal model. In this section, an algorithm to convert a motion to a physiologically feasible one is explained. Each trajectory of the body is assumed to be expressed by the sum of basis functions multiplied by the coefficients:

$$\theta_i = \sum_j c_i^j B^j, \quad (22)$$

where  $\theta_i$  is the kinematic trajectory of the  $i$ th DOF,  $B^j$  are the basis functions (such as a B-splines, trigonometric functions, etc.), and  $c_i^j$  are the scalar coefficients.

The trajectories must be at least  $C^2$  continuous, because the joint angles  $\theta$ , angular velocities  $\dot{\theta}$ , and angular accelerations  $\ddot{\theta}$  are used in the operation.

From the kinematic values  $\theta$ ,  $\dot{\theta}$ ,  $\ddot{\theta}$ , the joint torque is calculated by inverse dynamics:

$$\tau = f_1(\theta, \dot{\theta}, \ddot{\theta}). \quad (23)$$

The joint torque is then decomposed to the musculotendon force by the method explained in Sect. 5.3:

$$f^t = f_2(\theta, \tau). \quad (24)$$

However, the muscle force calculated in this way might violate the maximum force  $f^{\max}$  that is exertable by the muscle. We check for every muscle during the entire motion whether the muscle force is within the limit.

Let us define the state of the trajectory by the coefficients of the basis function and the time each motion terminates:

$$x = (c_0^0, \dots, c_m^0, \dots, c_0^{n_c}, \dots, c_m^{n_c}, t_f), \quad (25)$$

where  $m$  is the number of control points and  $n_c$  is the number of DOFs of the system.

It is necessary to evaluate at each moment whether each muscle can afford the motion. The following quadratic program is solved here for the evaluation:

$$\min_{f, \tau_{ext}} |\tau_{ext}|^2 \quad (26)$$

$$\tau = A f + \tau_{ext}, \quad (27)$$

where  $f = (f_1^t, \dots, f_{n_m}^t)^T$ ,  $A$  is the matrix that converts the muscle force to joint torque,  $\tau$  is the joint torque calculated by inverse dynamics in (23), and  $\tau_{ext}$  is the supplementary torque that is applied when the motion cannot be realized by the muscle force alone.

Another condition that the human body model must satisfy is the balance constraint defined in Sect. 5.4. For the motion to be feasible,  $|\tau_{ext}|^2 = 0$  and  $|s(q, \dot{q}, \ddot{q})|^2 = 0$  must be satisfied all through the motion. Therefore, the first step is to arrange the

kinematic trajectories in order to let the motion satisfy the following condition:

$$K(x, t) = |\tau_{ext}(x, t)|^2 + |s(q, \dot{q}, \ddot{q})|^2 \quad (28)$$

$$= 0, \quad (29)$$

where  $t_0 \leq t \leq t_f$ . Since  $s(q, \dot{q}, \ddot{q})$  is the amount of moment that must be applied to the human body model for keeping its balance,  $K$  is the square of the total external torque. In this paper, the motion is sampled  $n_s$  times to check whether (29) is satisfied. This means that each motion has  $n_s$  inequality constraints:

$$K_{i_s} = 0 (i_s = 0, \dots, n_s - 1). \quad (30)$$

The problem in the form

$$\min_{i_s} \{\max K_{i_s}\} \quad (31)$$

$$\text{s.t. } x_{\min} \leq x \leq x_{\max}$$

is solved to obtain a motion that satisfies all the constraints, while  $x_{\min}$  and  $x_{\max}$  consist of the minimum and maximum values of the joint angles and the termination time. The inequality constraint of (7) limits the range of each joint angle at every frame. The optimization is done by a program called CF-SQP (Lawrence et al. 1997), in C code, for solving constrained nonlinear optimization problems.<sup>3</sup> The problem is solved by the following process. Given  $x$ , let

$$\Phi(x) = \max_{i_s} K_{i_s}(x). \quad (32)$$

Further, given the descent direction  $d$  of  $x$ , let

$$\Phi'(x, d) = \max_{i_m, i_s} \{K_{i_s}(x + d) + \langle \nabla K_{i_s}(x), d \rangle\} - \Phi(x), \quad (33)$$

which is a first-order approximation to  $\Phi(x + d) - \Phi(x)$ , while  $\langle \cdot, \cdot \rangle$  represents the inner product.  $\Phi(x)$  can be minimized by SQP, which iteratively solves a quadratic program to obtain the descent direction  $d$  for the update of the state vector  $x$ . The quadratic program at iteration  $k$  can be written in the following form:

$$\min_d \frac{1}{2} d^T H_k d + \Phi'(x_k, d), \quad (34)$$

$$\text{s.t. } x_{\min} \leq x_k \leq x_{\max}$$

while  $H_k$  is the Hessian matrix of  $\Phi(x)$ .  $H_0$  is initially set to the identity matrix, and  $H_k$  is updated by the (BFGS) formula with Powell's modification (Powell 1978).

As  $\Phi(x)$  becomes zero, the operation terminates. At this time, all  $K_{i_s}$ s are zero. The resulting motion is a physiologically feasible one.

## 8 Additional space-time constraint optimization

Users can impose further constraints besides feasibility to realize the motion they want.

The operation proposed here is based on optimization. A criterion is specified by the user, and it is optimized while satisfying the constraints in (30).

The user can characterize the motion by defining the criteria of the motion. For example, if the user wants a motion with the least effort of the muscles,

$$J = \int_{t_0}^{t_f} \sum_i \left( \frac{f^i}{f_{max}^i} \right)^2 dt \quad (35)$$

is a good objective function (Komura et al. 1997).

If the user wants to create a motion similar to the specified one, the objective function can be defined as that of Gleicher (1998) and Gleicher and Litwinowicz (1998):

$$J = \int_{t_0}^{t_f} \Delta\theta^T M \Delta\theta dt \quad (36)$$

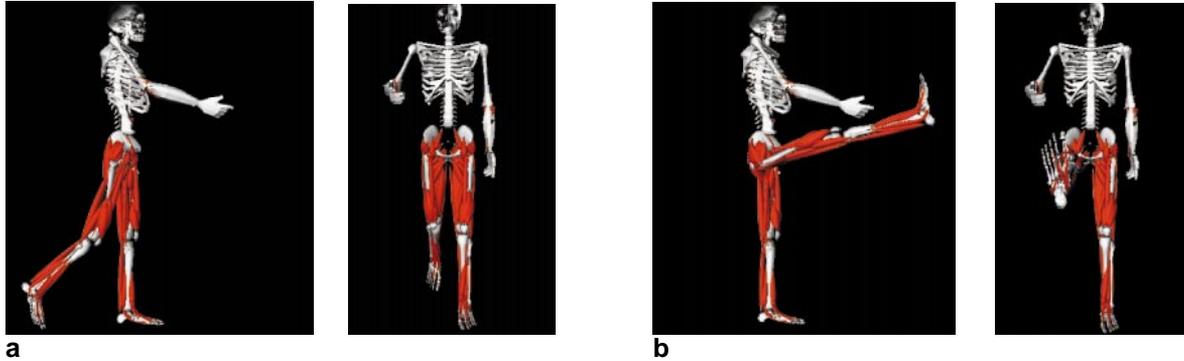
$$\Delta\theta = \theta(t) - \theta_u(t), \quad (37)$$

where  $\theta_u$  is the trajectory provided by the user,  $\theta$  is the controlled trajectory by the basis functions, and  $M$  is a weight matrix that determines the importance of each DOF.

The inequality constraints of (7) and (30) are also taken into account to keep the motion physiologically feasible and to satisfy kinematic requirements. The operation here can be summarized in the following form:

$$\begin{aligned} & \min_x J \\ \text{s.t. } & K_{i_s} = 0 (i_s = 0, \dots, n_s - 1) \\ & x_{\min} \leq x \leq x_{\max} \end{aligned}$$

<sup>3</sup> See <http://www.isr.umd.edu/Labs/CACSE/FSQP/fsqp.html> for detail.



**Fig. 7a,b.** The initial frame (a) and the final frame (b) of the kicking motion, which are used to determine the kicking motion

The iteration starts from the feasible state calculated by the operation explained in the previous section. This problem is an optimization problem with inequality constraints, which can be solved by SQP. Again, CFSQP has been used for the calculation. At each iteration  $k$ , we calculate the descent direction vector  $d$  by solving the following quadratic program:

$$\min_d \frac{1}{2} d^T H_k d + \langle \nabla J, d \rangle \quad (38)$$

$$\text{s.t. } x_{\min} \leq x_k \leq x_{\max}$$

$$K_{i_s} + \langle \nabla K_{i_s}, d \rangle = 0, (i_s = 0, \dots, n_s - 1),$$

where  $H_k$  is the Hessian matrix of  $J$ .  $H_k$  is updated as explained previously.

## 9 Experimental results

In this section, example motions created by our method are shown. First, a simple kicking motion is created by physiological retargetting. The kicking motion is repeated to show the effects of the fatigue model. Next, an example of a standing-up motion is shown. The effect of displacement of some muscles, which can be considered a motion by an injured person, is shown. Finally, examples of gait motion are shown.

### 9.1 Kicking motion

Two keyframes are specified to create a kicking motion (Fig. 7). The posture in Fig. 7b is not feasible for

most people because the right knee is completely extended while the right hip is greatly flexed. In such a posture, the hamstring muscle is stretched excessively. The balance of body in both of the postures is not stable either.

First, these two postures were simply interpolated by cubic B-spline curves. Six control points were prepared for every DOF, and they were arranged so that the human body model had the same posture as Fig. 7a at the initial time  $t_0$ , and as Fig. 7b at final time  $t_f$ .  $t_0$  is 0 and  $t_f$  is 1 at the beginning.

Needless to say, the motion initially created is not feasible. The motion has to be modified by the method explained in Sect. 7. The motion is then optimized to let the leg reach the highest point during the motion. The objective function is written in the following form:

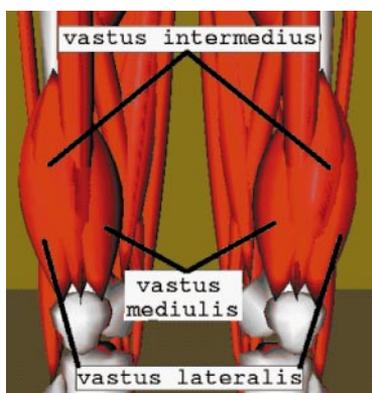
$$J = - \max_{t_0 \leq t \leq t_f} y_{\text{ankle}}, \quad (39)$$

while  $y_{\text{ankle}}$  is the height of the right ankle during the motion.

The results of the retargetting operation are shown in Fig. 8 (left). In this motion, observe that the human body model first pulls back the leg, then starts the kicking motion, kicks up the right foot to the highest position, and finally relaxes the leg. By comparing the raised leg with the motion interpolated by B-splines, the thigh flexion angle is decreased and the knee flexion angle is increased at the moment the kicking leg reaches the highest point. The  $t_f$  of the retargetted motion is 0.78. This means that the motion should be quick enough (finishing in 0.78 s) to



**Fig. 8.** The physiologically retargetted kick at the beginning (*left*) and after 100 attempts at the motion (*right*)



**Fig. 9.** The vastus medialis, vastus intermedius, and vastus lateralis muscles

raise the leg to that height. It is also important to note that the position of the right foot in the last frame is lower than the position in the previous frame. This is because a human cannot hold his leg at the maximum height. Such facts cannot be derived without a musculoskeletal model. This is one of the contributions in this paper. The whole body is controlled properly to keep the ZMP above the supporting left foot.

Next, the kicking motion was repeatedly simulated, and the pH level of each muscle was calculated. The initial motion was again retargetted, since the leg can no longer achieve the previous kick because of the decreased pH level of the muscles. This operation was repeated 100 times. The retargetted kicking motion after these repetitions is shown in Fig. 8 (right). Observe the effect of fatigue from the lower foot position of the kicking leg at the final posture. The whole motion looks unstable compared to the initial motion because of the tired supporting leg.

### 9.2 Standing-up motion

In this subsection, standing-up motions under two conditions were created. The first example is a standing-up motion under a normal condition that was created from two keyframes. Another example shows the standing-up motion for a human body model when the force exertable by the vastus medialis, vastus intermedius, and vastus lateralis muscles (Fig. 9) of the right leg were reduced to zero. These muscles are used to extend the knee. Equation 35 was used as the objective function this time.

The optimized standing-up motion calculated from two keyframes is shown in Fig. 10a. Next, the three

muscles were removed and the new standing-up motion was calculated. The new motion is shown in Fig. 10b. The initial keyframe posture is automatically retargetted. The human body model with powerless vastus muscles has shifted the pelvis above the left foot to reduce the ground force applying to the right foot. The human body model gradually moves the pelvis toward the middle and finally stands up.

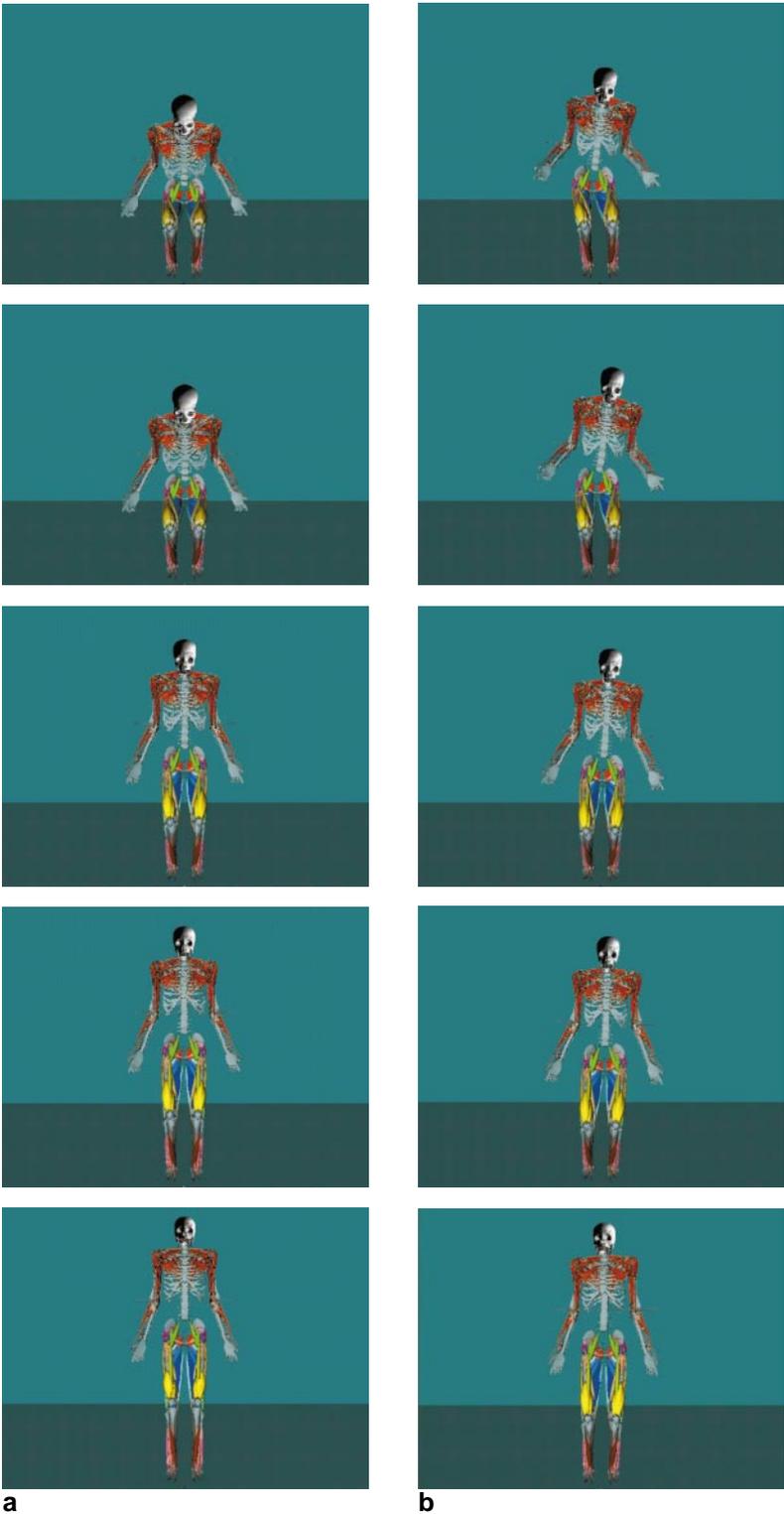
### 9.3 Gait motion

A gait motion was created from four keyframes shown in Fig. 11. The keyframes were again interpolated with cubic B-splines first. The interpolation yields a motion that is infeasible for a human body model. The physiological retargetting was applied to convert the motion to a feasible one. Then, optimization was applied to the motion to minimize the effort by the muscles (35). The optimal gait motion is shown in Fig. 12.

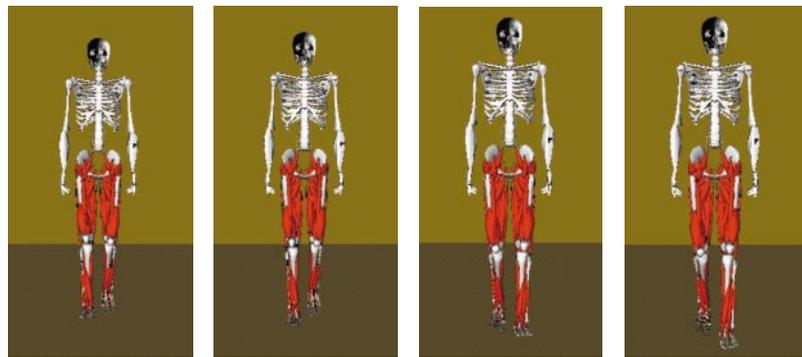
Next, the gluteus medius muscles (Fig. 13), mainly used during gait and particularly when the leg is in the support phase, were removed from the leg. This can be considered as a simulation of the gait by an injured person. The gait motion was again generated by the same steps. The final gait motion is shown in Fig. 14. We can observe the effects of displacement of the muscles by the motion. The gait in the disabled model is much more unstable than the gait in the normal human body model.

## 10 Discussion

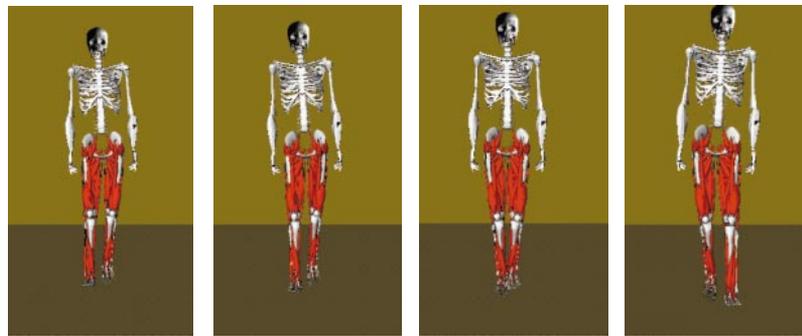
In this section, the validity of the conversion algorithm proposed in this paper is discussed by comparing the kicking motion calculated in the previous section with a real human kick. There are a number of features that can be found in both motions. Several frames of the kicking motion are shown in Fig. 15. Comparing the front view of the motions, both figures begin the motion by raising the right foot (top frame), tilt the upper half of the body to bring the center of mass above the support foot (second frame), and extend the leg to the front together with abduction of the hip joint (third frame). The leg then reaches the highest point (fourth frame), and finally is lowered (fifth frame). Observing the side view of the motions, the correspondence of the leg and chest trajectories are clearer. The knee joint of the kicking leg is flexed first, and is extended as the leg is swung



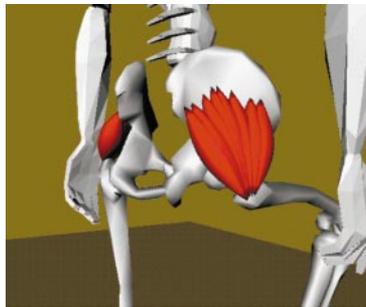
**Fig. 10a,b.** The feasible standing-up motion (a) and the standing-up motion (b) after displacement of the vastus medialis, vastus intermedius, and vastus lateralis muscles from the right leg



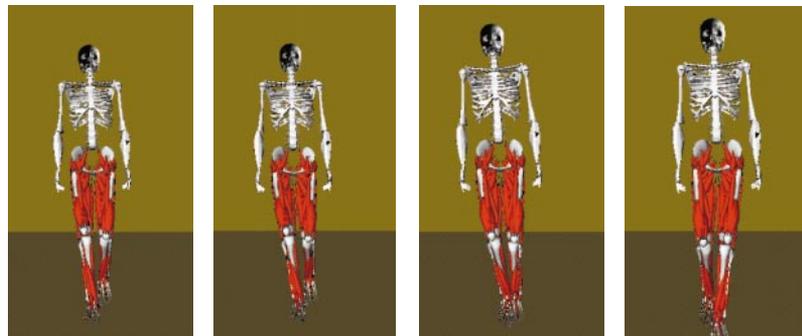
11



12



13



14

**Fig. 11.** The four keyframes used to create the half cycle of a gait motion

**Fig. 12.** The optimal gait motion under normal condition

**Fig. 13.** The gluteus medius (side-back view)

**Fig. 14.** The optimal gait motion when the gluteus medius muscles are disabled

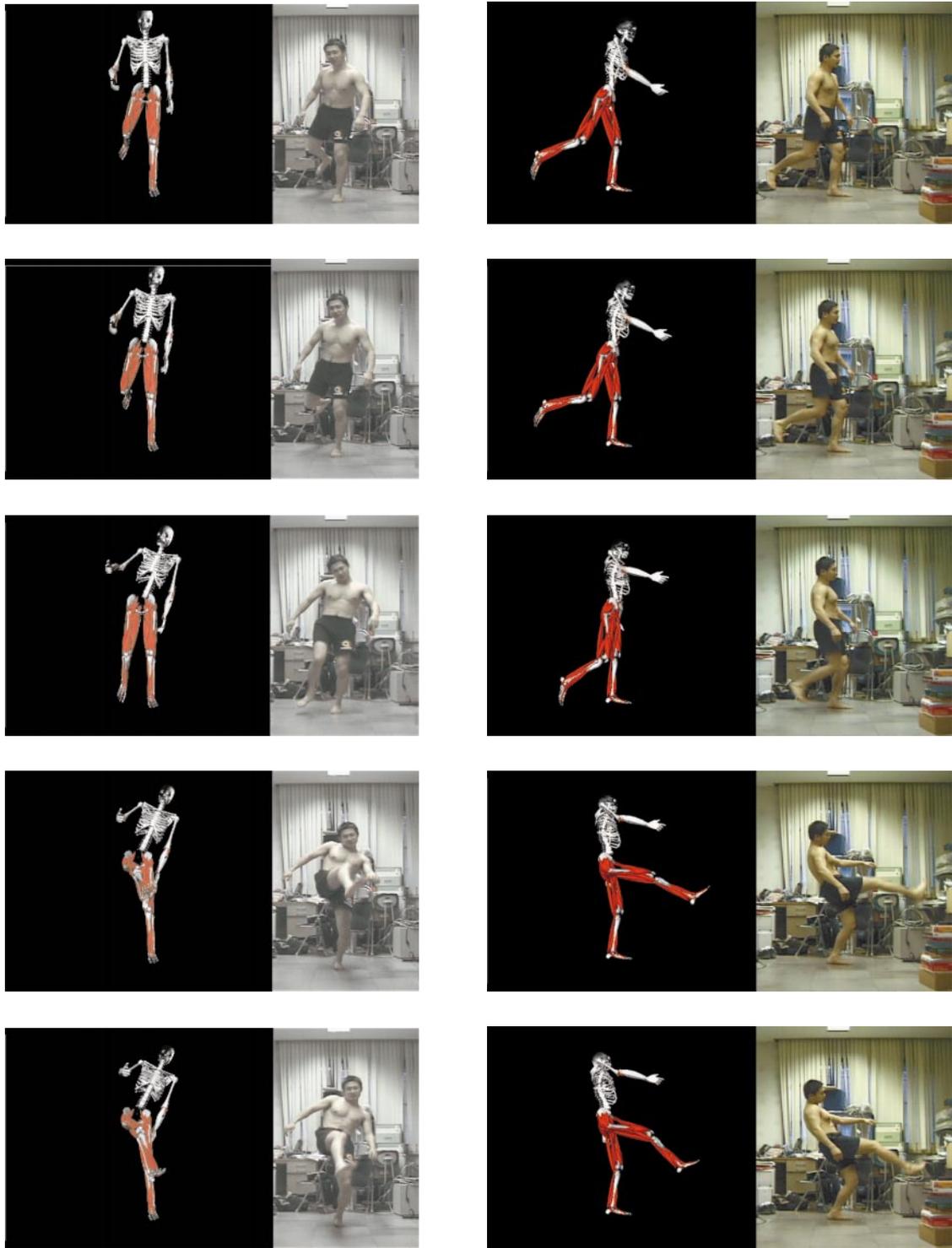


Fig. 15. Comparison of the kicking motions

to the front, and after the leg is kicked out, the knee joint is flexed again. The chest is bent backward as the kicking motion proceeds.

The great difference between the motions is that the rotation of the chest joint by the musculoskeletal human body model is greater than that by the real human body. This great rotation can be considered the result of modeling the chest joint by a single rigid segment. Actually, the human spine is composed of a number of bones, and the rotation of the chest is much more flexible and smooth. This problem can be avoided by modeling the spine more precisely. However, it can be seen that the calculated trajectory of the kicking leg and the upper half of the body is similar to that of the real human body.

## 11 Conclusions and future work

In this paper, we have presented a method to create and retarget human body motions using the musculoskeletal model. The musculoskeletal model is essential for creating dynamically and physiologically feasible motions. Our method makes it possible to simulate physiological effects such as fatigue and injuries. Such simulation has been difficult for contemporary systems that have not taken into account the inner structure of the body.

The application of the conversion algorithm to a single posture results in a feasible posture that can be achieved by the musculoskeletal model, which has an application in inverse kinematics systems.

Simulation of rehabilitation is a future application. Construction of a musculoskeletal model of the upper half of the body, which will enable the generation and retargetting of a great number of motions, and the quantitative evaluation of the simulated motions, is in progress.

## References

1. Armstrong WM, Green MW (1985) The dynamics of articulated rigid bodies for purposes of animation. *Visual Comput* 1:231–240
2. Bruderlin A, Williams L (1995) Motion signal processing. (Proceedings of SIGGRAPH '95) *Comput Graph* 29:97–104
3. Chen DT, Zeltzer D (1992) Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method. (Proceedings of SIGGRAPH '92) *Comput Graph* 26:89–98
4. Cohen MF (1992) Interactive spacetime control for animation. (Proceedings of SIGGRAPH '92) *Comput Graph* 26:293–302
5. Crowninshield RD, Brand RA (1981) A physiologically based criterion of muscle force prediction in locomotion. *J Biomech* 14:793–800
6. Delp S (1990) Surgery simulation: a computer graphics system to analyze and design musculoskeletal reconstructions of the lower limb. PhD Thesis, The Department of Mechanical Engineering, Stanford University, Stanford, Calif.
7. Delp S, Loan P, Hoy M, Zajac FE, Fisher S, Rosen J (1990) An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Trans Biomed Eng* 37:757–767
8. Frierich JA, Brand RA (1990) Muscle fiber architecture in the human lower limb. *J Biomech* 23:91–95
9. Giat Y, Mizrahi J, Levy M (1993) A musculotendon model of the fatigue profiles of paralyzed quadriceps muscle under fes. *IEEE Trans Biomed Eng* 40:664–674
10. Giat Y, Mizrahi J, Levy M (1996) A model of fatigue and recovery in paraplegic's quadriceps muscle subjected to intermittent fes. *Trans ASME: J Biomech Eng* 118:357–366
11. Gleicher M (1998) Retargetting motion to new characters. (Proceedings of SIGGRAPH '98) *Comput Graph*, pp 33–42
12. Gleicher M, Litwinowicz P (1998) Constraint-based motion adaptation. *J Visualization Comput Anim* 9:65–94
13. Gomi H, Kawato M (1996) Equilibrium-point control hypothesis examined by measured arm stiffness during multi-joint movement. *Science* 272:117–120
14. Hodgins J, Pollard NS (1997) Adapting simulated behaviors for new characters. (Proceedings of SIGGRAPH '97) *Comput Graph*, pp 153–162
15. Hodgins JK, Wooten WL, Brogan DC, O'Brien JF (1995) Animation of human athletics. (Proceedings of SIGGRAPH '95) *Comput Graph*, pp 71–78
16. Ko H, Badler NI (1996) Animating human locomotion with inverse dynamics. *IEEE Comput Graph Appl* 16:50–59
17. Komura T, Shinagawa Y, Kunii TL (1997) Muscle-based feed-forward controller of the human body. *Comput Graph Forum* 16:C165–C176
18. Komura T, Shinagawa Y, Kunii TL (1999) The calculation and visualization of dynamic ability of the human body. *J Visualization Comput Anim* 10:57–78
19. Laszlo J, Panne M van de, Fiume E (1996) Limit cycle control and its application to the animation of balancing and walking. (Proceedings of SIGGRAPH '96) *Comput Graph* 30:155–162
20. Lawrence CT, Zhou JL, Tits AL (1997) User's guide for CFSQP version 2.5: C code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality constraints. Technical Report TR-94-16r1, Institute for Systems Research, University of Maryland, College Park, MD 20742
21. Lee J, Shin SY (1999) A hierarchical approach to interactive motion editing for humanlike figures. (Proceedings of SIGGRAPH '99) *Comput Graph*, pp 39–48
22. Liu Z, Gortler SJ, Cohen MF (1994) Hierarchical spacetime control. (Proceedings of SIGGRAPH '94) *Comput Graph* 28:35–42

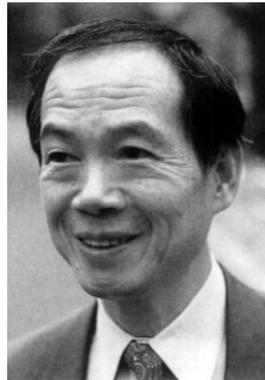
23. Panne M van de (1996) Parameterized gait synthesis. *IEEE Comput Graph Appl* 16:40–49
24. Pandy MG, Zajac FE, Sim E, Levine WS (1990) An optimal control model for maximum-height human jumping. *J Biomech* 23:1185–1198
25. Popovic Z, Witkin A (1999) Physically based motion transformation. (Proceedings of SIGGRAPH '99) *Comput Graph*, pp 11–20
26. Powell MJD (1978) A fast algorithm for nonlinearly constrained optimization calculations. *Numerical Analysis, Dundee, 1977, Lecture Notes in Mathematics 630*, pp 144–157
27. Rose C, Guenter B, Bodenheimer B, Cohen MF (1996) Efficient generation of motion transitions using spacetime constraints. (Proceedings of SIGGRAPH '96) *Comput Graph* 30:147–152
28. Scheepers F, Parent RE, Carlson WE, May SF (1997) Anatomy-based modeling of the human musculature. (Proceedings of SIGGRAPH '97) *Comput Graph*, pp 163–172
29. Unuma M, Anjyo K, Takeuchi R (1995) Fourier principles for emotion-based human figure animation. (Proceedings of SIGGRAPH '95) *Comput Graph* 29:91–96
30. Vukobratovi M, Borovac B, Surla D, Stoki D (1990) *Biped Locomotion*. Springer, Berlin, Heidelberg, New York
31. Wilhelms J (1987) Using dynamic analysis for realistic animation of articulated bodies. *IEEE Comput Graph Appl* 7:13–27
32. Wilhelms J, Gelder AV (1997) Anatomically based modeling. (Proceedings of SIGGRAPH '97) *Comput Graph*, pp 172–180
33. Winters JM (1990) Hill-based muscle models: a systems engineering perspective. In: Winters JM, Woo SL-Y (eds) *Multiple muscle systems: biomechanics and movement organization*. Chapter 5, Springer, New York, pp 69–93
34. Witkin A, Kass M (1988) Spacetime constraints. (Proceedings of SIGGRAPH '88) *Comput Graph* 22:159–168
35. Wooten WL, Hodgins JK (1996) Animation of human diving. *Comput Graph Forum* 15:3–13
36. Yamaguchi DG, Zajac FE (1990) Restoring unassisted natural gait to paraplegics via functional neuromuscular stimulation: a computer simulation study. *IEEE Trans Biomed Eng* 37:886–902



**TAKU KOMURA** is currently a doctoral student of the Department of Information Science at the University of Tokyo. He received his BSc (1995) and MSc (1997) degrees in Information Science from the University of Tokyo. His research interests include computer graphics and biomechanics. He is a member of the IEEE Computer Society and ACM.



**YOSHIHISA SHINAGAWA** is currently an Assistant Professor of the Department of Information Science at the University of Tokyo. He received his BSc (1987), MSc (1990), and DSc (1992) degrees in Information Science from the University of Tokyo. His research interests include computer graphics, vision, and its applications. He has published more than 55 refereed academic/technical papers in computer science. He is on the Editorial Board of *The International Journal of Shape Modeling*. He is a member of the IEEE Computer Society, ACM, IPSJ and IEICE.



**TOSIYASU L. KUNII** is Professor Emeritus of the University of Tokyo and Professor of Hosei University. He was the Founding President and Professor of the University of Aizu. He received his BSc in 1962, MSc in 1964, and DSc in 1967, all from the University of Tokyo. Dr. Kunii is the Founder of the Computer Graphics Society, Editor-in-Chief of *The Visual Computer*, and Associate Editor-in-Chief of *The Journal of Visualization and Computer Animation*. He is on the Editorial Boards of *Information Systems Journal*, *Information Sciences Journal*, and *IEEE Computer Graphics and Applications*.