THE JOURNAL OF VISUALIZATION AND COMPUTER ANIMATION
J. Visual. Comput. Animat. 2000; **11:** 223–235

Online motion retargetting

*By Kwang-Jin Choi and Hyeong-Seok Ko**

This paper presents a method to retarget the motion of a character to another in real time. The technique is based on inverse rate control, which computes the changes in joint angles corresponding to the changes in end-effector position. While tracking the multiple end-effector trajectories of the original subject or character, our online motion retargetting also minimizes the joint angle differences by exploiting the kinematic redundancies of the animated model. This method can apply a captured motion to another anthropometry so that it can perform slightly different motion, while preserving the original motion characteristics. Because the above is done online, a real-time performance can be mapped to other characters. Moreover, if the method is used interactively during motion capture session, the feedback of retargetted motion on the screen provides more chances to get satisfactory results. As a by-product, our algorithm can be used to reduce measurement errors in restoring captured motion. The data enhancement improves the accuracy in both joint angles and end-effector positions. Experimental results show that our retargetting algorithm preserves high-frequency details of the original motion quite accurately.

Received: January 2000; Revised: 10 June 2000

KEY WORDS: character animation; motion retargetting; motion editing; inverse kinematics

Introduction

The dream of animating complex living creatures with pure computation (such as inverse kinematics, or dynamic control) has proven impractical. Even though creatures are not free from physics, their motion is not a direct consequence of physics. Dynamic control can provide solutions based on simplified assumptions about human motion. However, the result tends to look quite mechanical.

If a high-quality character animation has to be produced during a short period of time, motion capture is probably the most reasonable choice these days. The captured data itself is for a specific person in performing a specific motion. Whenever the data needs to be reused, it has to be retargetted to account for the differences in the anthropometry and motion. Therefore motion retargetting is emerging as an important technique in recent character animation.

If the original motion characteristics are severely lost during motion retargetting, the technique loses its merit over pure computational approaches. The problem we try to solve in this paper can be summarized as: (1) finding in real time the motion retargetted to a new character that has different anthropometric proportions, and (2) at the same time, preserving the features of the original motion during the retargetting. Our solution with the above two properties has an interesting by-product (see 'Motion Capture Data Enhancement', below). The algorithm can be used to enhance motion capture data by reducing the errors in joint angles and end-effector positions.

Online motion retargetting presented in this paper is based on *inverse rate control*[1] (or resolved motion rate control), which is a way to implement inverse kinematics based on Jacobian. It computes the changes in joint angles corresponding to the changes in end-effector position. While tracking the multiple end-effector trajectories of the original subject or character, our online motion retargetting imitates the joint motion of the original character by exploiting the kinematic redundancies of the animated model. Moreover, jerky motion is prevented since the next configuration is dependent on the previous configuration in inverse rate control. As will be shown in the experimental results, our algorithm can preserve high-frequency

*Correspondence to: Hyeong-Seok Ko, SNU Human Animation Center, School of Electrical Engineering, Seoul National University, San 56-1, Shinlim-dong, Kwanak-ku, Seoul 151-742, Korea.

details of the original motion, which carries important characteristics of the motion.

Figure 1 shows the online retargetting process schematically. The input is a stream of joint angle vectors θ^{src} of the measured subject in the source motion and another stream of the reference (or desired) end-effector positions x_1 of the *animated character* at discrete time ticks. The output is a stream of joint angle vectors θ^{des} of the animated character during the destination motion at corresponding time ticks. The filter in the figure is causal, i.e., the output is calculated based on the current and immediately previous input values, but does not depend on the future input. This explains why it is called *online*.

If the retargetting can be done online, real-time performance can be mapped to another character, or the feedback of the retargetted animation can facilitate a motion capture session so that satisfactory results can be obtained with fewer trials. Since the memory required for online retargetting does not increase with time, our algorithm can handle an infinitely long sequence of motion.

The primary goal of our online motion retargetting (OMR) is to track the given reference end-effector trajectory $x_1(t)$, and the secondary goal is to imitate the pattern of joint angle trajectory $\theta^{src}(t)$ as much as possible. Therefore $\theta^{src}(t)$ carries the content to be retargetted, and $x_1(t)$ carries the variations needed during the retargetting. For example, when there is a bat-swing motion, we can obtain different swing motions aiming at different hit positions by specifying $x_1(t)$ appropriately.

As a by-product, our OMR algorithm can reduce measurement errors in restoring captured motion. If the positioning of the pose is done from joint angles alone, the errors can accumulate as the forward kinematic positioning propagates toward the end-effector. The end-effector position data, if they are measured, can be utilized to limit the above error accumulation within a certain range. For the data enhancement, we use the end-effector position data for $x_1(t)$. The details are further explained under 'Motion Capture Data Enhancement', below.

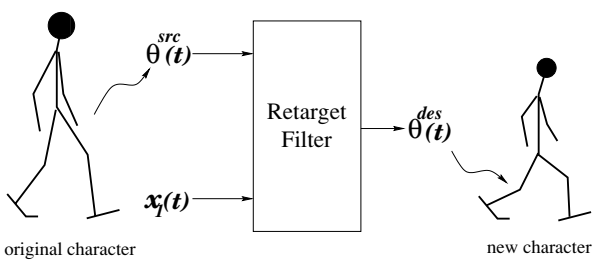

Figure 1. Online motion retargetting filter.

The rest of this paper is organized as follows. The next section reviews recent work related to motion retargetting. The third section discusses inverse rate control and its implementation, and the fourth section presents the problem formulation of motion retargetting with inverse rate control. The fifth section discusses how our OMR can be used to reduce measurement errors in restoring captured motion. The sixth section shows experimental results of our technique. Finally, the seventh section concludes the paper.

Related Work

Several techniques have been proposed for reusing or altering existing motions. Witkin and Popović's motion warping[2] and Bruderlin and Williams' motion displacement mapping[3] discussed motion editing technique based on direct manipulation of data curves. Bruderlin and Williams[3] and Unuma and Takeuchi[4] utilized signal processing techniques for motion editing. Wiley and Hahn[5] proposed an interpolation synthesis algorithm that chooses and combines the most relevant motions from the database to produce animation with specific positional goals.

Although some of those techniques can be used (with user's extra efforts) for retargetting motions, they do not specifically address the motion retargetting problem. Boulic and Thalmann[6] presented a motion editing technique that combines direct and inverse kinematic control. The concept called *coach–trainee metaphor* is very similar to the formulation of the motion retargetting problem. The basic idea is to consider the joint motion of *coach* as a reference input to *trainee* motion, and exploit the null space of the Jacobian when solving inverse kinematics so that the trainee's motion follows the coach's. The inverse kinematic constraint is given by half-spaces such as planes, cylinders, or spheres. Although the approach shares the technique of utilizing the redundancy in inverse kinematic control with ours, it cannot be considered as a motion retargetting technique, but rather as a motion correction technique; the end-effector constraints specified by half-spaces are not sufficiently general so as to solve the motion retargetting problem.

Gleicher proposed a method devoted to the motion

retargetting problem.[7] He used the space-time constraint method, which minimizes an objective function $g(x)$ subject to the constrains of the form $f(x)=c$. The constraints can represent the ranges of parameters, or various kinds of spatial–temporal relationship among the body segments and the environment. The objective function is the time integral of the signal displacement between the source and destination motions, i.e.

$$g(x)=\int\left(m^{src}-m^{des}\right)^2 \mathrm{d}t \qquad (1)$$

Since the whole interval has to be integrated to find the optimal solution, the method is intrinsically an offline process.

It is interesting to note that the local methods such as our OMR technique do not necessarily trade off the quality with the computational speed. The global method of Gleicher[7] can correlate frames back and forth within the whole duration and thus generally produces smoother results than local methods. But there are several reasons that the result of our OMR also has a reasonable level of global and local motion coherence:

- OMR takes continuous input: The look-ahead property of the global method is effective only when the constraints are imposed at sparse key frames. On the other hand, our OMR can take continuous trajectories of constraints as input, which can contribute to produce a globally coherent motion in spite of the local computation.
- OMR exploits kinematic redundancy to resemble the original motion: The global coherence is also achieved by the effort to exploit the kinematic redundancy of the system in resembling the original motion.
- OMR is based on inverse rate control: The local coherence of the motion results from the fact that the adjacent frames are interrelated by the inverse rate control.

Note that the global and local coherence is not the only measure for judging the retargetting quality. The results produced by local methods can be better than the ones produced by global methods in some aspects. For example, local methods are superior in preserving high-frequency details of a motion. While global methods obtain a motion by optimizing a single scalar value throughout the whole motion, local methods perform (a kind of) optimization at each frame. While global methods unnecessarily correlate frames far apart along the time axis, the frames are not correlated in local methods except for the adjacent ones. Therefore the *fidelity of motion retargetting* (preservation of the local parameter fluctuation) is better in local methods than in global methods.

Bindiganavale and Badler[8] presented a method to abstract and edit motion capture data. Their algorithm detects significant events, abstracts constraints from the motion, and imposes those constraints on other characters. The constraints abstracted from the motion are solved by inverse kinematics at significant frames; then those frames are interpolated. Although the constraint abstraction is an improvement over other techniques, the interpolation technique may fail to preserve high-frequency details when keyframes are spaced sparsely.

Inverse Rate Control

In an articulated figure, the joint configuration can be related to the position and orientation of the end-effector by a kinematic mapping f: $\Theta \rightarrow X$, which maps the joint space Θ to Cartesian space X. The mapping is usually a non-linear equation given by

$$x_1=f_1(\theta) \qquad (2)$$

where x_1 is an m-dimensional vector and θ is an n-dimensional vector. We use $m=3$ if we consider only position, or $m=6$ if we are interested in both position and orientation of the end-effector. The value of m can be 12 or 18 when we impose multiple end-effector constraints.

If we differentiate the above equation, we obtain

$$\dot{x}_1=J_1\dot{\theta} \qquad (3)$$

where $\dot{x}_1$ and $\dot{\theta}$ denote the end-effector positional velocity and joint angle velocity, respectively. J_1 is called Jacobian and is an $m \times n$ matrix that linearly relates the end-effector velocity and joint angle velocity at the moment.

Given the end-effector velocity, we can obtain joint angle velocity by inverting the Jacobian. However, most articulated figures have kinematic redundancy (i.e., $m<n$) and thus the inverse of Jacobian is not unique. Therefore there are an infinite number of possible solutions that satisfy equation (3). Some criteria can be specified for the selection of the solution that best fits our purpose. One of the popular criteria is the *minimal norm solution*

$$\dot{\theta} = J_1^+ \dot{x}_1 \tag{4}$$

where $J_1^+ = J_1^T (J_1 J_1^T)^{-1}$ is the pseudo inverse* of J_1.

Equation (4) gives a particular solution, and can be generalized to include all possible solutions by adding a term from the null space of J_1 as in

$$\dot{\theta} = J_1^+ \dot{x}_1 + (I - J_1^+ J_1) y \tag{5}$$

where y is an arbitrary n-dimensional vector. Note that $(I - J_1^+ J_1)$ projects y onto the null space of J_1. This null space term corresponds to the redundant degrees of freedom, and can be utilized to perform secondary priority tasks.[9–12] For example, consider the following task set:

$$\text{primary task: } x_1 = f_1(\theta), \text{ thus } \dot{x}_1 = J_1 \dot{\theta},$$

$$\text{secondary task: } x_2 = f_2(\theta), \text{ thus } \dot{x}_2 = J_2 \dot{\theta}$$

If the equation

$$\dot{\theta} = J_1^+ \dot{x}_1 + (I - J_1^+ J_1) J_2^+ \dot{x}_2 \tag{6}$$

is used for equation (5), then the primary goal is accurately achieved in the case of continuous domain† and the secondary goal is also achieved in an optimal sense.

Another way of utilizing the redundancy of the system is to set y to the gradient $-\alpha \Delta g$ of a criterion function $g(\theta)$ in equation (5). Then integration of equation (5) tries to reduce the value of $g(\theta)$ while the end-effector is made to track the given trajectory.[13]

Closed-Loop Inverse Rate Control and its Discrete Implementation

To control an articulated figure so that it follows a given reference end-effector trajectory $x_1(t)$, we need to integrate $J_1^+ \dot{x}_1(t)$ to get the value of θ (t) (equation (4)). But this open-loop fashion of integration cannot eliminate the initial tracking error $e_1(t_0) = x_1(t_0) - x_1^{des}(t_0)$, where $x_1^{des}(t)$ is the resulting end-effector position at time t in the destination motion.

Balestrino and Sciavicco,[14] Tsai and Orin,[15] Chiacchio *et al.*[16] and Sciavicco and Siciliano[17] proposed the *closed-loop inverse kinematics* (CLIK) scheme based on Jacobian pseudo-inverse. CLIK leads to zero steady state error, which means that the error is exponentially convergent to zero for a fixed target position. For CLIK, equation (4) has to be modified to

$$\dot{\theta}(t) = J_1^+ (\dot{x}_1(t) + K_1 e_1(t)) \tag{7}$$

where K_1 is a positive definite matrix we provide. It can be easily shown that, as the smallest eigenvalue of K_1 becomes large, the convergence rate increases accordingly since the error dynamics is governed by the relation $\dot{e}_1 + K_1 e_1 = 0$. In a continuous time formulation such as equation (7), a large value of K_1 is desirable. However, as will be shown below, an arbitrarily large K_1 does not guarantee convergence when we implement the discrete version of equation (7).

The difference equation corresponding to Equation (7) is given by

$$\Delta\theta[n] = J_1^+[n-1](\Delta x_1[n] + K_1 e_1[n]) \tag{8}$$

where

$$\Delta\theta[n] = \theta[n] - \theta[n-1]$$

and

$$\Delta x_1[n] = x_1[n] - x_1[n-1]$$

$$e_1[n] = x_1[n] - x_1^{des}[n] \tag{9}$$

Here $u[i]$ is the value of the function $u(t)$ at the discrete time t_i.

Equation (8) is implicit: to compute $\Delta\theta[n]$, we need to know the value of $e_1[n]$. But computing $e_1[n]$ by equation (9) in turn requires the value of $x_1^{des}[n]$, which is not available until $\Delta\theta[n]$ is known. Therefore, $e_1[n]$ should be estimated. Below we show that any estimation based on the old values (at n-1, n-2, …) requires K_1 to be I for the best tracking performance.

Suppose that we estimate $e_1[n]$ simply with $e_1[n\text{-}1]$. Then equation (8) becomes

$$\Delta\theta[n] = J_1^+[n-1](\Delta x_1[n] + K_1 e_1[n-1]) \tag{10}$$

To obtain the error equation, multiply both sides of equation (10) by $J_n[n-1]$ and we obtain

$$J_1[n-1]\Delta\theta[n] = \Delta x_1[n] + K_1 e_1[n-1] \tag{11}$$

Assuming that the step size is small enough, we can rewrite the above equation as

$$\Delta x_1^{des}[n] = \Delta x_1[n] + K_1 e_1[n-1] \tag{12}$$

*We used the damped least squares solution, $\dot{\theta} = J_1^+ (J_1 J_1^T + \lambda^2 I)^{-1} \dot{x}_1$, to obtain consistent motion near the singularities.[18,19]

†In the discrete domain, since the Jacobian is a linear estimation of a non-linear function, the integration can produce errors, especially when the step size is large.

With the relations $e_1[n]=x_1[n]-x_1^{des}[n]$ and $e_1[n-1]=x_1[n-1]-x_1^{des}[n-1]$, equation (12) can be rewritten as

$$e_1[n]=(I-K_1)e_1[n-1] \tag{13}$$

Equation (13) reveals that the eigenvalues of I-K_1 should be within the interval (-1,1) to prevent the error from growing indefinitely. Even with $K_1=I$, the stability is not guaranteed due to the non-linearity of f_1. (Note that $J_1[n-1]\Delta\theta[n]$ is approximated as $\Delta x_1^{des}[n]$ in equation (12).) But in practice we found that instability rarely occurs at a usual sampling rate ($30\sim60$ Hz) in dealing with human motion.

If we include the secondary task $x_2=f_2(\theta)$, the open-loop control law takes the form

$$\dot{\theta}=J_1^+(\dot{x}_1+K_1e_1)+\left(I-J_1^+J_1\right)J_2^+\dot{x}_2 \tag{14}$$

To prevent possible divergence due to errors, however, a closed-loop version needs to be considered again. The CLIK scheme including the secondary task based on Jacobian transpose is given by

$$\dot{\theta}=J_1^+(\dot{x}_1+K_1e_1)+\left(I-J_1^+J_1\right)J_2^TK_2e_2 \tag{15}$$

where $e_2=x_2-x_2^{des}$. The term x_2^{des} represents the actual result of the secondary task that tries to realize the given goal x_2.

It is proven that e_2 is ultimately bounded within a certain range and the tracking error for the primary task is not affected by the second term of equation (15).[16] But again, arbitrarily large K_2 is not allowed in discrete implementation. With the estimation of $e_2[n]$ based on old values, a reasonable choice for K_2 is I.

The final CLIK scheme in discrete domain with $K_1=K_2=I$ is given by

$$\begin{aligned}\Delta\theta[n]=&J_1^+[n-1](\Delta x_1[n]+\tilde{e}_1[n])\\&+\left(I-J_1^+[n-1]J_1[n-1]\right)J_2^T\tilde{e}_2[n]\end{aligned} \tag{16}$$

where $\tilde{e}_1[n]$ and $\tilde{e}_2[n]$ are the estimations of $e_1[n]$ and $e_2[n]$, respectively. Although any estimation scheme based on the old values cannot completely eliminate the error caused by the non-linearity of f_1, it can be reduced with a higher-order estimation for $e_1[n]$ than $e_1[n\text{-}1]$, and consequently can give better tracking performance. We found that the estimation rule described below gives satisfactory results.

Step 1: $\Delta\tilde{\theta}[n]=J_1^+[n-1]\Delta x_1[n]$

Step 2: $\tilde{x}_1^{des}[n]=f_1\left(\theta[n-1]+\Delta\tilde{\theta}[n]\right)$

$$\tilde{x}_2^{des}[n]=f_2\left(\theta[n-1]+\Delta\tilde{\theta}[n]\right)$$

Step 3: $\tilde{e}_1[n]=x_1[n]-\tilde{x}_1^{des}[n]$

$$\tilde{e}_2[n]=x_2[n]-\tilde{x}_2^{des}[n]$$

The above procedures complete the discrete implementation of the CLIK algorithm with a secondary task.

Inverse Rate Control with Multiple End-Effector Trajectories

In this section, we discuss how to extend inverse rate control so that it can track multiple end-effector trajectories.

The serial chain is not suitable for modelling living creatures since their underlying articulated structures contain branches. An illustrative example is taken from the human upper body, and is shown in Figure 2. The model consists of a spine and two arms. The waist is the root of the kinematic tree structure, and the two arms are branching at the top of the spine. If both hands have their own goals to reach, and the motion of each hand is computed by solving inverse kinematics separately for the two chains (from the waist to the left and right hands, respectively), then the spine angles thus generated will be different in the two solutions.

Zhao and Badler[20] solved this problem by a weighted sum of independently obtained gradients, each of which directs its corresponding end-effector to a goal position. However, the effects of different weights are not easy to predict. Depending on the weight assignment, their algorithm may not find an inverse kinematic solution even if all the constraints

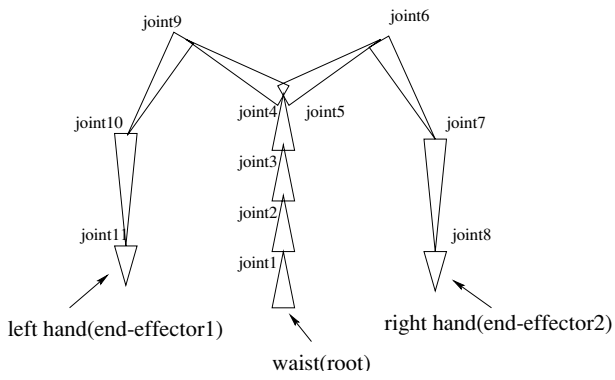

Figure 2. Kinematic structure of human upper body.

can be actually satisfied. Intrinsically, finding the solution of an inverse kinematics problem with multiple constraints does not require any weight or priority assignment: if all the end-effector constraints can be met, then the solution should exist no matter what the values of the weights or priorities are.

Compared with Zhao and Badler's algorithm, Jacobian-based inverse rate control provides a quite simple and intuitive solution to the problem. In order to incorporate multiple end-effector constraints we simply need to concatenate the end-effector vectors and compose the Jacobian appropriately. In the above example, the end-effector vector x_1 should be a 12-dimensional vector and the Jacobian J_1 should be a 12×33 matrix (two end-effector constraints with six DOFs each, and 11 joint constraints with three DOFs each). Of course, the Jacobian will have zeros where the joint angle and the end-effector have no relation (e.g., the left elbow joint and right hand). In inverse rate control, the conflict of the spine angles mentioned above is resolved during the computation of the pseudo inverse of the Jacobian.

Motion Retargetting with Task Priority Strategy

In general, we can formulate the motion retargetting problem with the following task set, and can solve for θ^{des}:

$$\text{primary task: } x_1 = f_1(\theta^{des}) \tag{17}$$

$$\text{secondary task: } x_2 = f_2(\theta^{des}) \tag{18}$$

x_1 in the above is the desired end-effector trajectory. It can be taken from the source character (and then be modified for necessary variations), or it may also be provided by the user.

According to equation (6), the open-loop control law for these tasks is given by

$$\dot{\theta}^{des} = J_1^+ \dot{x}_1 + (I - J_1^+ J_1) J_2^+ \dot{x}_2 \tag{19}$$

The block diagram of its closed-loop version is shown in Figure 3.

Joint angle trajectories contain important characteristics of a motion, and the end-effector movements are already tracked by the primary task. Thus an obvious and useful choice for the secondary task might be to imitate the joint motion of the source character, i.e.

$$\text{secondary task: } \theta^{src} = \theta^{des} \tag{20}$$

This is simply the case when θ^{src} and the identity function are used for x_2 and f_2, respectively, in Equation (18). Then Equation (19) becomes

$$\dot{\theta}^{des} = J_1^+ \dot{x}_1 + (I - J_1^+ J_1) \dot{\theta}^{src} \tag{21}$$

The block diagram of the closed-loop control scheme with the secondary task of joint motion imitation is shown in Figure 4. Reasonable choices for K_1 and K_2 are I's in discrete implementation as stated before. But K_1 and K_2 can be adjusted based on the *dexterity measure* to get consistent motion near the kinematic singularities. A popular dexterity measure is $\sigma_{min}/\sigma_{max}$, where σ_{min} and σ_{max} are the minimum and maximum, respectively, among the singular values of the Jacobian. When the dexterity measure is small, smaller K_1 and K_2 should be used.

The adaptive scheme can be also used for motion transition. Smaller gain (e.g. $K_1 = K_2 = 0.1I$) will produce sluggish tracking, but smooth motion. Therefore, when the animated model switches to another motion and there exists a large discrepancy at the motion boundary, smooth transition can be obtained by adjusting the gain matrix K_1 and K_2 appropriately.

As briefly mentioned before, $K_1 = I$ does not guarantee stability since the non-linear function $\dot{\theta}$ is linearly approximated by $J_1^+ \dot{x}_1$. The system can become unstable when $\dot{x}_1$ gets very large, or the sampling rate is very low. Therefore another provision for enforcing stability might be to clamp the value that goes into the box J_1^+ in Figure 4 whenever it is over a certain threshold. The provision might be effective when there is an excessively large acceleration, or when the model is fully stretched and almost no manipulative

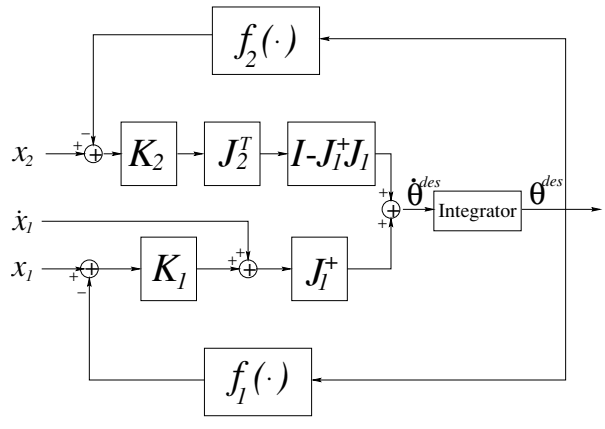

Figure 3. Closed-loop control scheme with a secondary task.

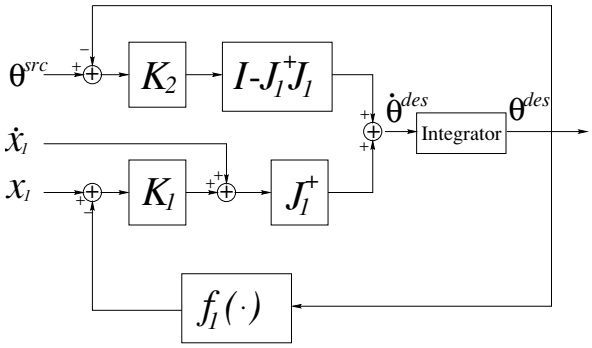

Figure 4. Closed-loop control scheme with the secondary task of joint motion imitation.

redundancy is left in the system. (In dealing with human motion, however, the above provision was almost never needed.)

Motion Capture Data Enhancement

When we capture a motion, we often measure the joint angles and use forward kinematics to reconstruct the motion. But the method can introduce large end-effector position errors since the joint angle error near the base is amplified when it comes to the end-effector, and joint angle errors are accumulated as the forward kinematic positioning propagates toward the end-effector.

Choi *et al.*[21] proposed an interpolation/regression method, which applies inverse kinematics at sparse keyframes and the resulting joint angles are interpolated with cubic spline curves. The interpolation is combined with least squares fitting so that the characteristics of the original joint angle data are preserved in the resulting motion.

The OMR algorithm described in the previous section can be used to reduce measurement errors in restoring the captured motion. The new method is an improvement on the above interpolation/regression method in three respects: (1) inverse kinematics is done at every frame, which promises much closer end-effector tracking, (2) the joint angle imitation is done by exploiting redundant degrees of freedom rather than solely depending on the least squares fit, and (3) the high-frequency component of the original motion is preserved in the new method more effectively.

For the enhancement, we measure both joint angle and end-effector trajectories during the motion capture session. The measured trajectories are supplied to our motion retargetting algorithm: the end-effector trajectories are supplied to x_1, and the joint angle trajectories are supplied to θ^{src}. Of course, the destination character has to be the same as the source character, if pure data enhancement needs to be done. As the retargetting progresses, θ^{src} will be adjusted to θ^{des} so that the end-effector constraint x_1 is met while maintaining the joint angle pattern of θ^{src}.

Compared with the forward kinematic motion reconstruction, our OMR algorithm reduces end-effector errors remarkably. In general, our algorithm also reduces the errors in joint angle measurements. While the joint angle errors can accumulate in forward kinematic reconstruction, once it is processed by our OMR, the total amount of accumulated error is limited by the amount of end-effector position error. Moreover, the joint angle error due to the end-effector position error is distributed among all the joints. Therefore unless the amount of end-effector position error is considerably larger than that of joint angle errors, our OMR produces a more accurate result than the unprocessed data.

It is interesting to note that the retargetting and data enhancement are done simultaneously. Even when a different destination character is used, these two steps are carried out at the same time. The fact is particularly useful when a real-time performance is retargetted.

Experiments

This section describes two experimental results of our OMR algorithm. In the first experiment, we show a retargetting example in which our OMR is applied to retarget a walking motion. The result demonstrates that our OMR based on inverse rate control is not inferior in quality to the retargetting based on space-time constraints. Major error analysis of the algorithm is given in this example. In the second experiment, we show the retargetting of a bat-swing motion. The motion clips mentioned below are available at http://graphics.snu.ac.kr/demo/omr/omr.mov.

Retargetting of Walking Motion

In this experiment, the source motion (refer to the motion clip #1) is a walking motion along a curved path, which was procedurally generated by the

locomotion algorithm of Ko.[22] The walker took 13 steps and produced a total of 390 frames. The kinematic structure of the characters used for the walking motion is shown in Figure 5.

Since lower body motion is far more important than upper body motion in walking, we retargetted only the lower body motion. As shown in Figure 5 the lower body consists of pelvis, upper leg, lower leg, foot, and toes; they are connected at the hip, knee, ankle, and ball joints. The total degrees of freedom of the lower body is $8 \times 3 + 6 = 30$. (All the joints were modelled by 3-DOF joints, and the base has six extra DOFs.)

The destination character was scaled down by about 60% from the source character with non-uniform proportions. The two characters are shown in Figure 6, and the lower body dimensions are compared in Table 1.

In the retargetting, the secondary task was set to $\theta^{\text{src}} = \theta^{\text{des}}$. To specify the primary task, we set the toe-tip of the stance leg as the base and the toe-tip of the swing leg as the end-effector. The source character's toe-tip trajectory was used for x_1 without any modification. Therefore the destination character had to take bigger steps relative to his body size. At the boundaries of steps, the base and end-effector were switched. This implies that we may have discontinuity at the boundary when the tracking error is large. The retargetted motion with the above task set is shown in the motion clip #2. The tracking error of the swing foot was negligible and thus the produced motion was smooth at the step boundaries.

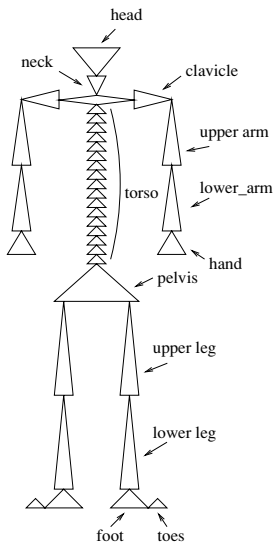

Figure 5. Kinematic model used for walking motion.

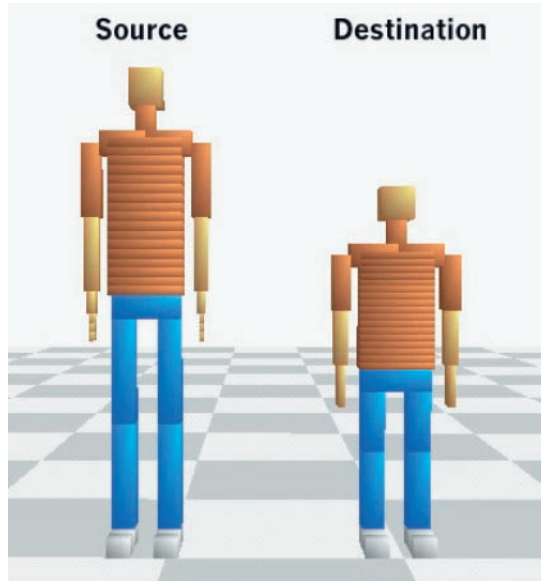

Figure 6. Characters used for walking motion (left: source character, right: destination character).

But the pelvis motion showed non–uniform speed along the direction of progression (anterior–posterior), which was not observable in the source motion. So we constrained the transverse plane motion of the pelvis; that is, the pelvis was designated as another end-effector, and the (x,z) component of the source character's pelvis movement was tracked in the destination motion. (Note that the pelvis motion along the y-axis should be adapted to account for the height difference.) After adding the constraint, we obtained a satisfactory result as shown in the motion clips #3 and #4. Even with the extra constraints, the end-effector trajectories of the source and the destination made an accurate match. The comparison is shown in Figure 7. The dotted curves for the source motion are very difficult to see because they overlap with the solid curves, the end-effector trajectories of the destination motion.

	src character	des character	Ratio
Pelvis (width)	30.0	30.0	1.00
Upper leg	46.0	26.0	0.57
Lower leg	46.0	34.0	0.74
Foot	16.0	16.0	1.00
Toes	8.0	8.0	1.00

Table 1. Comparison of lower body dimensions

To show the tracking error microscopically, the area within a small box near the 150th frame in Figure 7 is magnified in Figure 8. The trajectories in the figure show that the tracking error is kept small where the velocity is nearly constant, but the error increases when the velocity makes sudden changes. The maximum error (1.0464 cm) occurred at the 128th frame where the y-coordinate (height) of the toe-tip reached its peak acceleration and this error was reduced to a negligible level at the 135th frame as the acceleration decreased. The step boundary was taken from low-acceleration points so that the base to end-effector switch makes a smooth transition.

The joint angle trajectories of the left leg during the original and retargetted motion are plotted in Figure 9. Only the angles around the sideways direction (medial–lateral) axes are presented in the graphs. The comparison shows that the amplitude of the hip angle increases in the destination motion to cover the given step length with the relatively smaller body. Other than that, the original joint angle pattern was preserved quite well.*

At the end of each step, the ball joint of the source character showed an abrupt change from a large negative value to zero. It corresponds to the toe-off moment when the toes take off the ground. After the retargetting, the sharp corner of the trajectory was well preserved. In general, our OMR preserves the high-frequency content of the motion quite faithfully, since inverse rate control is directed by Jacobian values. Big mountains or valleys are never missed. However, to recover tiny fluctuations as well, a high sampling rate is needed to avoid aliasing.

If sharp corners are undesirable, they can be prevented by adjusting the gain matrix K_2 or clamping some of the control input as stated earlier ('Motion Retargetting with Task Priority Strategy'). The adjustment of K_2 does not affect the end-effector tracking performance.

Retargetting of Bat-Swing Motion

In this experiment, actual performance of a bat-swing motion was processed by our OMR algorithm to produce the destination motion of three different characters shown in Figure 10. The anthropometry of character B is about the average. Character A has a longer torso but shorter limbs than average, and

*Note that zero error in tracking the joint angle trajectories is unachievable due to the anthropometric difference.

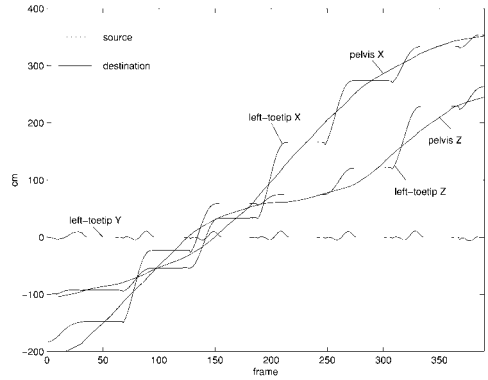

Figure 7. Comparison of end-effector trajectories.

character C has a shorter torso but longer limbs. Their kinematic structures are the same as the one shown in Figure 5 except that the torso is segmented to five parts and the feet are excluded.

To set the primary task, the base and end-effectors should be specified as before. In this experiment, the pelvis was chosen as the base and two hands were chosen as the end-effectors. Three 6-DOF sensors were used to capture the position and orientation of the base and end-effectors. The end-effector motion was directly supplied to $x_1(t)$ without any modification. Therefore character A, for example, had to make a relatively lower swing.

Since all the torso segments cannot be measured due to the limited number of sensors, we measured only the topmost segment. Therefore, the five joints from the waist to the topmost torso segment had to be generated from the orientation difference between the pelvis and the topmost torso segment. For this, the measured orientation of the topmost torso segment was added to the primary task, and zero angles for the

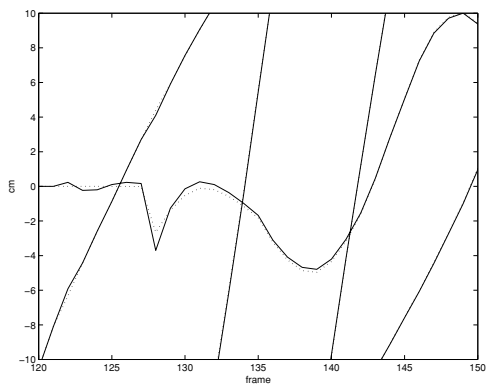

Figure 8. Comparison of end-effector trajectories (a magnified view).

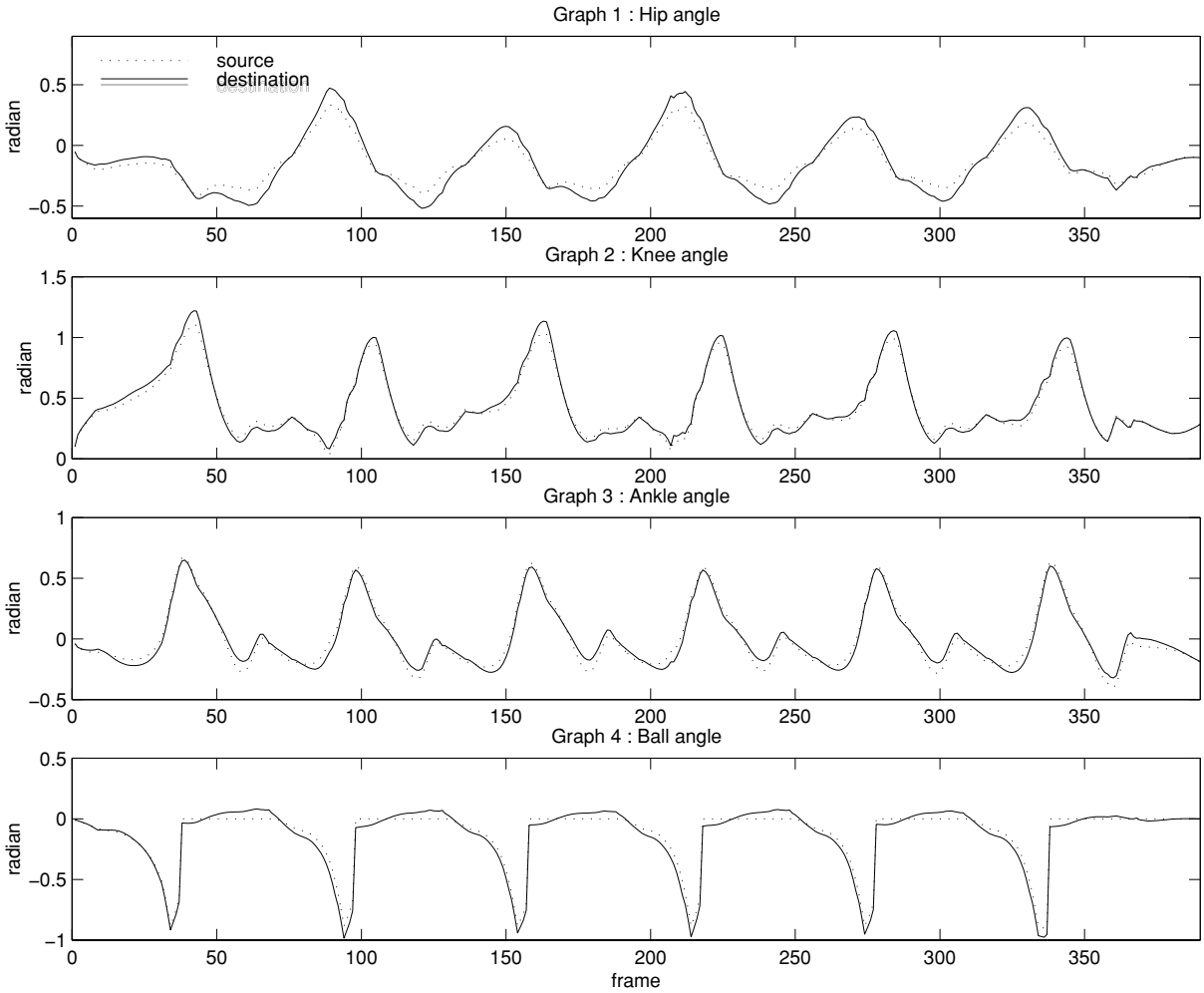

Figure 9. Comparison of joint angle trajectories.

unmeasured joints were added to the secondary task, expecting the primary task can be met with minimal joint angles along the torso.

The other sensors measured the joint angles. (Because we had only 13 sensors available, we gave up capturing the foot motion.) In this experiment only the upper body motion was adapted by OMR. The lower body motion was reconstructed by applying the measured joint angles directly.

The retargetting of the source motion to characters A, B, C is shown in the motion clips #5–6, #7–8, and #9–10, respectively. The small green boxes in the video indicate the position of the end-effectors and base. In the video we can observe that end-effector positions are accurately tracked.

Since the body dimensions of character B and the real performer are similar, the retargetted motion does not contain any noticeable difference from the source motion. In the case of character A, however, we can see the waist is bent to lower the hit position, and the torso is shifted forward to account for the shorter arms. In the case of character C, the torso is bent backward and makes a bigger twist to account for the longer arms and shorter torso. Snapshots taken during the retargetted motions clearly demonstrate the above adaptation for the anthropometric differences and are shown in Figure 11.

Computational Time

Since we were not able to install our program to a platform equipped with a motion capture system, we had to emulate the real-time motion capture. That is, the motion data captured at 30 Hz was fed to the OMR

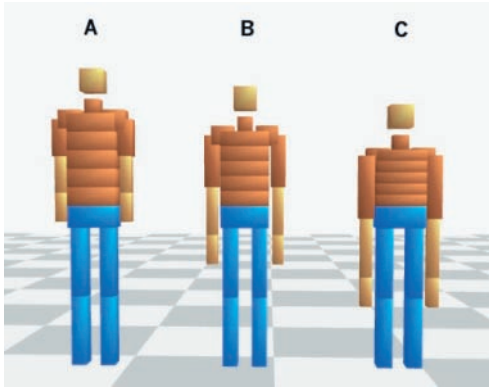

Figure 10. Three characters used for retargetting the bat-swing motion.

system with the same sampling rate using a timer. At this sampling rate, not a single frame was lost even with the visualization included. We used an Intergraph GX1 system (dual P-III 550, Wildcat 4000) for the experiments. Table 2 describes the computational time spent purely for the retargetting procedure in each motion. The code had not been fully optimized and so the performance may be improved further.

The slower rate of the bat swing motion is due to the bigger size of the Jacobian matrix compared to the walking motion (8×30 vs. 9×42). As shown in the table, the OMR is fast enough to process motion capture data collected at a usual sampling rate ($30 \sim 90$ Hz) in real time for models of reasonable complexity.

Conclusion

In this paper, we presented an online motion retargetting technique that is based on inverse rate control. The method is an improvement over the offline retargetting algorithm based on space-time constraints, since real-time performances can be retargetted without degrading the retargetting quality. The OMR algorithm greatly helps to get more satisfactory results in motion capturing with fewer trials by giving the real-time feedback to the performer. Furthermore, the captured data is enhanced in both end-effector positions and joint angles by going through our OMR filter.

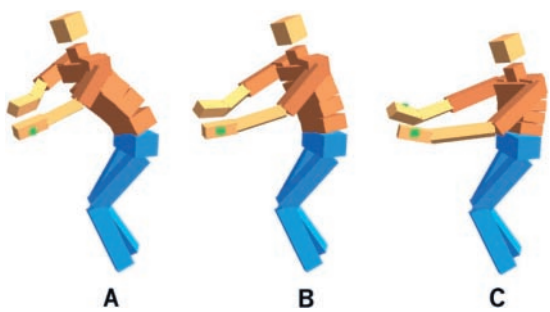

Figure 11. Snapshots taken from the retargetted motion. Observe the different adaptations to compensate for the anthropometric differences.

	Walking	Bat swing
Number of frames	390	136
Total DOFs	30	42
Constraint DOFs	8	9
Elapsed time (sec)	1.219	0.984
Frame rate* (Hz)	300.7	138.2

Table 2. Computational time spent for retargetting (*The frame rate does not include the time for visualization)

One minor unsolved problem is that there is no easy way to guarantee foolproof stability of the system due to the non-linearity. We observed that, at a very low sampling rate, when the model approaches kinematic singularity, thus leaving very little manipulative redundancy, then the system can become unstable. However, experimental results showed that the system never becomes unstable at 30 Hz or higher sampling rate. If the source motion is available only at a low sampling rate, two remedies are recommended: (1) by interpolating the source motion curves, first produce more samples, and then use them as the input to the OMR filter, or (2) scale down the end-effector trajectory to avoid the singular configuration, or use both (1) and (2).

The above remedies are for an extreme situation. Our online motion retargetting produces satisfactory results in retargetting most human or creature motion. This technique can be very useful for character animation and the games industry.

ACKNOWLEDGEMENTS

This work was supported by Creative Research Initiatives of the Korean Ministry of Science and Technology. This work was also partially supported by ASRI (Automation and Systems Research Institution), Seoul National University, and the Brain Korea 21 Project.

References

1. Whitney DE. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man–Machine Systems MMS*. **10**: 1969; 47–53.
2. Witkin A, Popovic Z. Motion warping. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, pp. 105–108, August 1995.
3. Bruderlin A, Williams L. Motion signal processing. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, pp. 97–104, August 1995.
4. Unuma M, Anjyo K, Takeuchi R. Fourier principles for emotion-based human figure animation. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, Cook R (ed.), pp. 91–96, August 1995.
5. Wiley DJ, Hahn JK. Interpolation synthesis of articulated figure motion. *IEEE Computer Graphics and Applications* 1997; November/December: 39–45.
6. Boulic R, Thalmann D. Combined direct and inverse kinematic control for articulated figure motion editing. *Computer Graphics Forum* 1992; **11**(4): 189–202.
7. Gleicher M. Retargeting motion to new characters. In *SIGGRAPH '98 Conference Proceedings*, Annual Conference Series, pp. 33–42. ACM SIGGRAPH, Addison-Wesley: Reading, MA, July 1998.
8. Bindiganavale R, Badler NI. Motion abstraction and mapping with spatial constraints. In *Modelling and Motion Capture Techniques for Virtual Environments, International Workshop, CAPTECH'98, Geneva*, pp. 70–82, November 1998.
9. Hollerbach JM, Suh KC. Redundancy resolution of manipulators through torque optimization. *IEEE Journal of Robotics and Automation* 1987; RA-**3**(4): 308–316.
10. Maciejewski AA, Klein CA. Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *International Journal of Robotics Research* 1985; **4**(3): 109–117.
11. Nakamura HHY, Yoshikawa T. Task-priority based redundancy control of robot manipulators. *International Journal of Robotics Research* 1987; **6**(2): 3–15.
12. Yoshikawa T. Analysis and control of robot manipulators with redundancy. In *Robotics Research: The First International Symposium*, Brady M, Paul ER (eds). MIT Press: Cambridge, MA, 1984; 735–747.
13. Liegeois A. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man and Cybernetics* 1977; SMC-**12**(12): 868–871.
14. Balestrino GDMA, Sciavicco L. Robust control of robotic manipulators. In *Preprints of the 9th IFAC World Congress, Budapest*, Vol. 6, Elsevier Science: Oxford 1984, pp. 80–85.
15. Tsai YT, Orin DE. A strictly convergent real-time solution for inverse kinematics of robot manipulators. *Journal of Robotic Systems* 1987; **4**(4): 477–501.
16. Chiacchio LSP, Chiaverini S, Siciliano B. Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *International Journal of Robotics Research* 1991; **10**(4): 410–425.
17. Sciavicco L, Siciliano B. A dynamic solution to the inverse kinematic problem for redundant manipulators. In *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, 1081–1087. IEEE Computer Society Press: Los Alamitas, CA; 1081–1087.
18. Nakamura Y, Hanafusa H. Inverse kinematic solutions with singularity robustness for robot manipulator control. *ASME Journal of Dynamic Systems Measurement and Control* 1986; **108**(3): 163–171.
19. Wampler CW. Manipulator inverse kinematic solutions based on damped least-squares solutions. *IEEE Transactions on Systems, Man and Cybernetics* 1986; SMC-**16**(1): 93–101.
20. Zhao J, Badler NI. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics* 1994; **13**(4): 313–336.
21. Choi K, Park S, Ko H. Processing motion capture data to achieve positional accuracy. *Graphical Models and Image Processing* 1999; **61**(5): 260–273.
22. Ko H. Kinematic and dynamic techniques for analyzing, predicting, and animating human locomotion. PhD thesis, University of Pennsylvania, Department of Computer and Information Science, Philadelphia, May 1994.

Authors' biographies:

Kwang-Jin Choi is a Ph.D. candidate in the School of Electrical Engineering at Seoul National University. He received his B.A. and M.S. in electrical engineering from Seoul National University in 1996 and 1998 respectively. His research interests include motion editing, cloth animation, and fluid simulation.

Hyeong-Seok Ko is an assistant professor in the School of Electrical Engineering at Seoul National University. He received his B.A. and M.S. in Computer Science from Seoul National University in 1985 and 1987, respectively, and Ph.D. in Computer Science from the University of Pennsylvania in 1994. After he gained his Ph.D., he joined the Computer Science Department at the University of Iowa during the academic years 1994 and 1995 as an assistant professor. Since then, he has been working at Seoul National University. Currently, he is the director of Seoul National University Human Animation Centre. His research interests include gross-body motion editing, facial animation, hairstyle modelling and animation, body deformation, cloth simulation, and virtual reality applications.