Computational Visual Media
DOI 10.1007/s41095-016-0065-1

Research Article

Dynamic skin deformation simulation using musculoskeletal model and soft tissue dynamics

Akihiko Murai[1](✉), **Q. Youn Hong**[2], **Katsu Yamane**[3], **and Jessica K. Hodgins**[3]

Abstract Deformation of skin and muscle is essential for bringing an animated character to life. This deformation is difficult to animate in a realistic fashion using traditional techniques because of the subtlety of the skin deformations that must move appropriately for the character design. In this paper, we present an algorithm that generates natural, dynamic, and detailed skin deformation (movement and jiggle) from joint angle data sequences. The algorithm has two steps: identification of parameters for a quasi-static muscle deformation model, and simulation of skin deformation. In the identification step, we identify the model parameters using a musculoskeletal model and a short sequence of skin deformation data captured via a dense marker set. The simulation step first uses the quasi-static muscle deformation model to obtain the quasi-static muscle shape at each frame of the given motion sequence (slow jump). Dynamic skin deformation is then computed by simulating the passive muscle and soft tissue dynamics modeled as a mass–spring–damper system. Having obtained the model parameters, we can simulate dynamic skin deformations for subjects with similar body types from new motion data. We demonstrate our method by creating skin deformations for muscle co-contraction and external impacts from four different behaviors captured as skeletal motion capture data. Experimental results show that the simulated skin deformations are quantitatively and qualitatively similar to measured actual skin deformations.

Keywords three-dimensional graphics and realism; musculoskeletal model; quasi-static muscle model; dynamic skin deformation

1 Introduction

Skin deformation of animated characters must be natural, dynamic, and detailed if the characters are to appear realistic and lifelike. This level of realism is particularly important in scenes of rich natural environments such as those in *The Jungle Book* and realistic special-effect shots such as those in *Planet of the Apes*. These deformations are essential for creating a sense of life: tension in the muscles and jiggle of the underlying muscle and soft tissue convey the exertion of the character and the dynamics of the motion.

A number of algorithms have been created for generating plausible skin deformation [1–3]. Most of these approaches drive the skin motion from the skeletal structure. Linear blending, spring–damper systems, and volumetric models have been used to create natural skin deformations. These approaches, however, do not accurately represent the complex underlying structure of bones, tendons, muscles, and interstitial tissue that are present in the human body. Recently, more effort has been expended in making anatomically grounded models. For example, Si et al. [4] used an anatomically correct human musculoskeletal model to simulate quasi-static skin deformation for the upper body based on detailed geometry and deformation models of the muscles.

In this paper, we present an algorithm that generates detailed skin deformation (movement and jiggle) from a skeleton animation based on

[1] National Institute of Advanced Industrial Science and Technology, Tokyo, 135-0064, Japan. E-mail: a.murai@aist.go.jp (✉).

[2] Carnegie Mellon University, USA.

[3] Disney Research, USA.

Manuscript received: 2016-08-31; accepted: 2016-10-23

standard motion capture joint angle data and three models: (i) a quasi-static muscle model, (ii) a muscle dynamics model, and (iii) a soft tissue dynamics model. Our approach has two main steps: identification of quasi-static muscle model parameters, followed by simulation of muscle and soft tissue dynamic deformation. In the identification step, which is performed only once for each body type, we compute subject-specific muscle shape parameters using a musculoskeletal model [5] and a short sequence of skin deformation data (slow jump) captured with a dense marker set from Ref. [6]. The quasi-static muscle model relates the quasi-static muscle shape to muscle length and tension, which can be obtained by computing the inverse kinematics and dynamics using a musculoskeletal model and joint angle data. Having obtained a muscle deformation model, we can simulate dynamic muscle deformations using only joint angle data. These can be obtained from skeletal motion capture (with 50–60 markers) or from a physically plausible keyframe animation. The simulation step first uses the quasi-static muscle deformation model identified in the previous step to obtain the quasi-static muscle shape for the given motion sequence. It then computes the dynamic skin deformation by simulating the passive muscle and soft tissue dynamics modeled as a mass–spring–damper system. To demonstrate the efficiency of this approach, we use skeletal motion capture data and electromyography (EMG) data to generate skin deformations that show muscle co-contraction and external impacts. Four motions, co-contraction, jumping, running, and impact, are created in this way using parameters identified from a slow jump.

1.1 Contributions

We realize a detailed skin deformation model based on a human musculoskeletal model and measured skin surface deformation data. The contributions of our work include: (i) a method for identifying muscle deformation model parameters from a short sequence of skin deformation data (using a slow jump) measured by motion capture using a dense marker set (400–450 markers), (ii) a method for applying the muscle and skin deformation model to joint angle data (for the motions co-contraction, jumping, running, and impact) recorded with 50–60 markers to create new sequences with detailed skin and muscle deformation. Our approach applies a parametric model for muscle deformation and a simple spring–damper model for soft tissue simulation for its simplicity and accuracy.

Simulation-based methods tend to only handle relatively static body parts and motions. Data-based methods find it difficult to handle novel motions. Our method combines simulation-based and data-driven approaches: our model has the ability to generalize to different types of motions that are not used for model identification, and the ability to generalize to different body types, e.g., with different fat thickness.

2 Related work

Skin deformation and dynamics are required for a realistic and natural-looking character, and therefore many approaches have been developed to generate this motion. One of the most common approaches is linear-blend skinning in which each skin vertex position is computed using a weighted sum of the positions of nearby joints. However, when linear-blend skinning is applied to a human character, the skin often lacks realism because of artifacts and because small details in the skin deformation are missing. A number of algorithms have been created to overcome these problems [2, 7].

More realistic models fall into two broad classes: simulated and data-driven. To model human-like creatures, researchers have proposed a layered approach in which the skin is driven by interactions between multiple underlying layers with different properties, based on anatomy. Chadwick et al. [8] introduced a deformable layer between the rigid skeleton and the skin surface to represent muscles and soft tissues. Various parts of the human body have been modeled in detail, such as the face [2], torso [9], hands [10], and upper body [11]. With this approach, the research focus has been on modeling the shape and deformation of muscles to reduce artifacts and express details in skin deformation. The muscle shape has been approximated using ellipsoids [12], or learned from anatomical data as in BodyParts3D [13]. Deformation of muscles and soft tissues is often simulated by physics-based models such as mass–spring–damper models [14] and volumetric models such as the finite element method [15–17] or the finite volume method [18].

To avoid issues with stability and computation, data-driven approaches model skin deformation directly from data rather than simulating the behavior of each layer in the musculoskeletal structure. Anguelov et al. [19] used SCAPE to build a pose deformation model and a body shape variation model from three-dimensional scan data, and to generate skin deformation for a new posture. Park and Hodgins [20] used motion capture to collect data, and then trained the parameters of a mass–spring model from motion captured data to generate skin deformation from skeletal motion capture data. Their mass–spring model treated the body parts as a homogeneous medium rather than having separate models for muscles, fat, and interstitial tissue as we do. Wang et al. [21] combined a mass–spring system and a data-driven wrinkle model to achieve wrinkling cloth simulation that was both efficient and physically plausible. Their wrinkle model relates the cloth wrinkles to joint angles. While we could build a similar model for muscle deformation, it would need to take joint angles, velocities, and accelerations as inputs, making the database size significantly larger than ours.

In biomechanics and robotics, many musculoskeletal models have been developed for simulation and analysis of human body dynamics [5, 22]. We use one of the musculoskeletal models developed in this field to obtain muscle pathways and tension [5]. However, these models focus on accurate simulation and analysis of human motion and do not include computation of the skin or muscle shape needed for animated characters.

3 Method

During dynamic motion, actuated muscles cause bone motion as well as muscle bulging due to tension and length changes. Additionally, muscles and soft tissue, including skin, fat, and viscera, deform passively in response to the bone and muscle movement as well as external forces from the environment. Accordingly, our skin deformation model comprises three sub-models (Fig. 1):

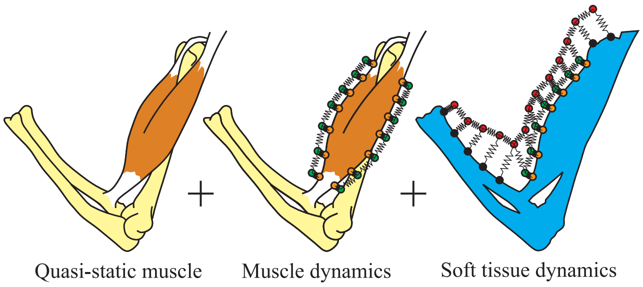


Fig. 1 The three models used in our skin deformation model. Left: the quasi-static muscle model that determines the quasi-static muscle shape. Middle: the muscle dynamics model that determines the dynamic deformation of muscles. Right: the soft tissue dynamics model that determines the dynamic deformation of fat and viscera.

1. A quasi-static muscle model that relates muscle length and tension to the quasi-static muscle shape. This model represents muscle bulging and relaxation due to different activation levels. We use a musculoskeletal model to estimate muscle length and tension from motion capture data.
2. A muscle dynamics model that describes the passive dynamics of muscles using a mass–spring–damper system. The model consists of point masses placed at the vertices of a muscle polygon model, connected by springs and dampers. Each point mass is also connected to the corresponding vertex of the quasi-static muscle by a spring and damper.
3. A soft tissue dynamics model that describes the passive dynamics of the skin and subcutaneous fat using a mass–spring–damper system. The model consists of point masses placed at the vertices of the polygons on the skin surface with springs and dampers connecting them to neighboring vertices and to the point masses on the dynamic muscle or bone surfaces.

The quasi-static muscle model includes a number of parameters that would be tedious to adjust by hand. We therefore provide a method to identify the quasi-static muscle model parameters from a sequence of measured skin geometry data. Although this method requires skin deformation data measured with a dense set of markers (400–450), the identification process is performed only once for each subject or body type. Having identified the parameters, our system can simulate skin deformation from novel motion data captured using standard marker sets (i.e., joint angle data). We describe the details of the modeling and identification processes in Section 3.2.

Figure 2 gives block diagrams of the identification and simulation processes, where the blocks with red borders are the new components developed

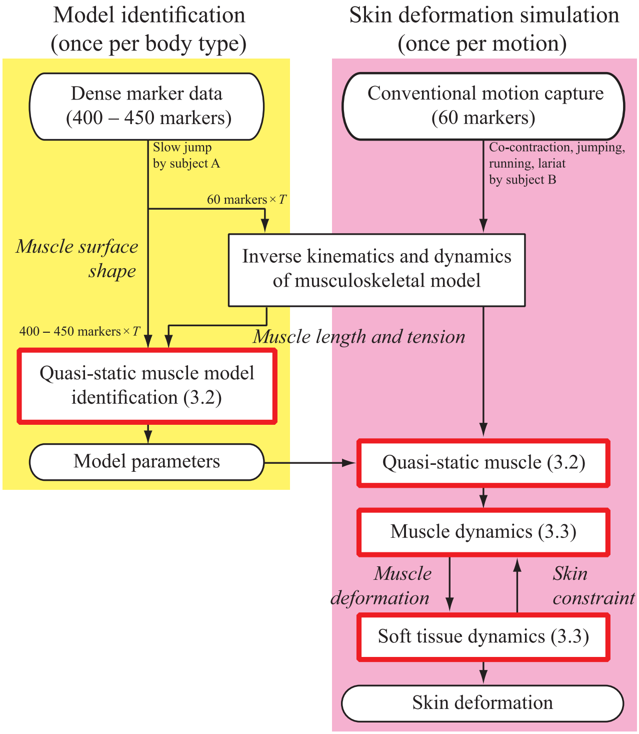


Fig. 2 Block diagram of the identification and simulation processes. The red borders depict the new components developed in this work. The identified model parameters are used in the simulation of motion sequences.

in this work. The details of each process are explained in the following subsections. In Section 3.1, we review the skin deformation data collection process [20] and the musculoskeletal model [5]. We then present the quasi-static muscle model and parameter identification process in Section 3.2. In Section 3.3, we describe the algorithm to simulate dynamic skin deformation using muscle and soft tissue dynamics models.

3.1 Skin deformation data and musculoskeletal model

Identifying the quasi-static muscle model parameters requires sample skin deformation data. We use the data recorded by Park and Hodgins [20] using an optical motion capture system with 400–450 reflective markers. The marker trajectories were recorded by 16 near-infrared Vicon MX-40 cameras at a rate of 120 fps. To identify the quasi-static muscle model parameters, we use a slow jump motion that is approximately 300 frames (2.5 s) in length. We intentionally use a slow sequence in which soft tissue dynamics do not play a big role in skin deformation.

In order to obtain the input data for the quasi-static muscle model, we apply inverse kinematics and dynamics algorithms from a musculoskeletal model [23] using the trajectories of 60 markers manually chosen from the full set of 400–450. The musculoskeletal model used in our work [5] consists of skeleton and musculo–tendon network models. The skeleton model has 155 degrees of freedom (DOF), and the inertial parameters (mass, inertia, and local center of mass) of the bone segments are computed from an average human model [24]. The musculo–tendon network model includes 989 muscles to drive the skeleton, as well as 50 tendons, 177 ligaments, and 34 cartilages to passively constrain the skeleton. Each of the muscles, tendons, and ligaments is represented by two end points (origin and insertion points), any number of via points, and straight pathways between them. Each origin, insertion, or via point is fixed with respect to a bone, and their locations are computed by solving forward kinematics (see Fig. 3).

We first obtain the joint angles of the skeleton model for each frame with an iterative inverse kinematics algorithm using the positions of the 60 markers as soft constraints [25]. Then, the joint torques required to execute the measured motion are computed by applying a recursive inverse dynamics algorithm for articulated rigid bodies [26]. Finally, we compute the muscle tensions required to produce the joint torques. The number of muscles is much larger than the number of joint torques, and this redundancy is resolved with mathematical optimization that minimizes the quadratic sum of the muscle tensions. The detailed algorithm is given in Refs. [5, 23, 27]. If EMG data are recorded at the

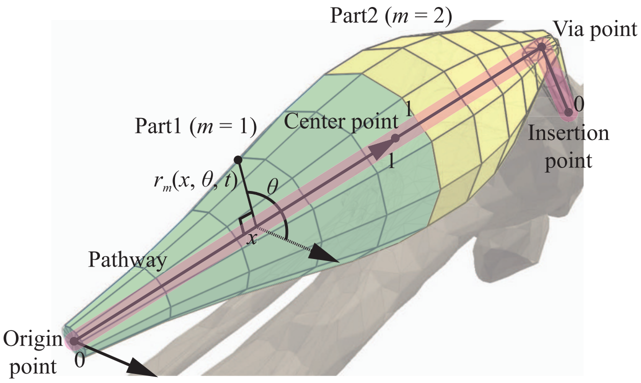


Fig. 3 Muscle shape and its local coordinate system. The red line represents the muscle pathway that connects the origin point, one or more via points, and the insertion point. x is the location along the muscle pathway, θ is the angle from the polar axis, and r_m is the distance from the pathway.

same time, we can obtain physiologically plausible muscle tensions for actions that are not observable from the motion, such as co-contraction [28].

3.2 Modeling and identification of quasi-static muscle model

We next develop a quasi-static muscle model that computes the quasi-static muscle shape from the muscle length and tension. The length terms realize the muscle volume constraint, and the tension terms realize the muscle bulging even for isometric contraction. We first choose about 300 surface muscles from the 989 muscles in Ref. [5] because we cannot identify the parameters of the inner muscles from surface data. The remaining 700 muscles are still used for inverse dynamics because their tensions do affect the tensions of the surface muscles. We construct the following quasi-static muscle model around the pathway of these 300 muscles.

Because most skeletal muscles have spindle-like shapes, we approximate the quasi-static muscle surface with a spindle whose cross-section perpendicular to the pathway is an ellipse, the size of which varies along the pathway according to a sigmoid function (see Fig. 3). The pennate muscles such as *pectoralis major*, whose cross-sectional shapes are quite different from an ellipsoid shape, are modeled with multiple thin spindle-shaped wires. The sigmoid parameters and the eccentricity are represented as functions of the muscle length and tension. In addition, we divide some muscles at their center point into two parts with different sets of sigmoid function parameters to represent asymmetric muscles such as the *soleus*. The same identification and simulation method can be applied to any muscle shape if the pathway of the muscle is given. In the following equations, we omit the muscle index for clarity.

We represent the quasi-static muscle surface shape in a cylindrical polar coordinate system for each part whose longitudinal axis is the muscle pathway (see Fig. 3). For a point on the m-th ($m = 1, 2$) part of a muscle, the distance from the pathway, r_m, is described by the location along the pathway x, the angle from the polar axis θ, and the current frame number t ($t = 1, 2, \ldots, T$) as

$$r_m(x,\theta,t) = \left(\frac{k_{m,3}(t)}{1+\mathrm{e}^{k_{m,1}(t)-k_{m,2}(t)x}} + k_{m,4}(t)\right) \times \sqrt{1-\varepsilon^2(t)\sin^2\theta} \tag{1}$$

where sigmoid function parameters $k_{m,n}(t)$ ($m = 1, 2, n = 1, 2, 3, 4$) and the eccentricity $\varepsilon(t)$ are functions of the muscle length $l(t)$ (the distance between its end points through its via points) and tension $\tau(t)$ that are computed by inverse kinematics and dynamics and mathematical optimization [5, 23, 27]:

$$\begin{aligned} k_{m,n}(t) &= \alpha_{m,n}l(t) + \beta_{m,n}\tau(t) + \gamma_{m,n} \\ &\qquad (n = 1, 2, 3, 4) \qquad (2) \\ \varepsilon(t) &= \alpha_5 l(t) + \beta_5\tau(t) + \gamma_5 \qquad (3) \end{aligned}$$

In our implementation, the x axis is normalized for each part so that $x = 0$ represents the origin or insertion of a muscle and $x = 1$ represents the center point. The local coordinate system of each part is defined with respect to the closest bone's local coordinate system at the initial skeleton posture. Therefore, the local coordinate system does not change discontinuously as long as the skeleton motion is continuous, and there will not be a sudden jump between the coordinate systems of two consecutive parts.

In this model, the total number of parameters to identify is 27 ($\alpha_{m,n}, \beta_{m,n}, \gamma_{m,n}$ ($m = 1, 2,\ n = 1, 2, 3, 4$), $\alpha_5, \beta_5, \gamma_5$) for each muscle. We determine these parameters at each muscle independently so that the muscle shape fits the skin deformation around the muscle during the motion capture sequence.

Let us define a muscle *segment* as a section of a muscle between two neighboring origin, insertion, or via points along the pathway, and denote the number of segments in a muscle by L. At each motion capture frame t, we find a user-defined number of markers closest to the pathway that belongs to each segment and represent their positions in the local cylindrical polar coordinate system of the muscle as $(\hat{r}_{k,t}, \hat{\theta}_{k,t}, \hat{x}_{k,t})$ $(k = 1, 2, \ldots, L)$. The closest marker of a segment may differ across frames, but the number of markers used is determined once for each body part.

We then solve an optimization problem to adjust the model parameters so that the total distance between the muscle surface and the positions of the closest markers is minimized. We used a gradient algorithm based on the interior reflective Newton method [29] to minimize the following quadratic cost function:

$$Z = \frac{1}{2}\left(Z_r + a_v Z_v + a_t Z_t + a_s Z_s\right) \tag{4}$$

where a_* are user-defined positive weights.

The first term Z_r represents the total squared distance between the muscle surface and measured marker data and is formulated as

$$\begin{aligned} Z_r &= \sum_{t=1}^{T}\sum_{k=1}^{L} \Delta r_{k,t}^{\mathrm{T}} \Delta r_{k,t} \qquad (5)\\ \Delta r_{k,t} &= \hat{r}_{k,t} - r_m(\hat{x}_{k,t}, \hat{\theta}_{k,t}, t)/\beta \qquad (6) \end{aligned}$$

where m represents the part containing segment k and β is a manually chosen fat thickness parameter defined in Fig. 4. We use $\beta = 1.0$ in the experiment. The second term Z_v represents the variance of the muscle volume across the entire motion sequence, formulated by

$$Z_v = \sum_{t=1}^{T} \left(V_1(t) + V_2(t) - \bar{V}\right)^2 \qquad (7)$$

where $\bar{V}$ is the average of the total volume during the whole motion sequence and $V_m(t)$ is the volume of part m at frame t computed by

$$V_m(t) = l_i(t) \int_0^1 r_m^2(x, 0, t)\pi\sqrt{1 - \varepsilon^2(t)}\mathrm{d}x \qquad (8)$$

This term represents the conservation of muscle volume [30].

The third term Z_t is added to constrain the radius at the origin and insertion so that the muscle is smoothly connected to the tendons at the ends. Z_t is formulated as

$$Z_t = \sum_{t=1}^{T} \left[(w - r_1(0,0,t))^2 + (w - r_2(0,0,t))^2\right] \qquad (9)$$

where w is a manually chosen tendon radius. We use $w = 0.01m$ in the experiment.

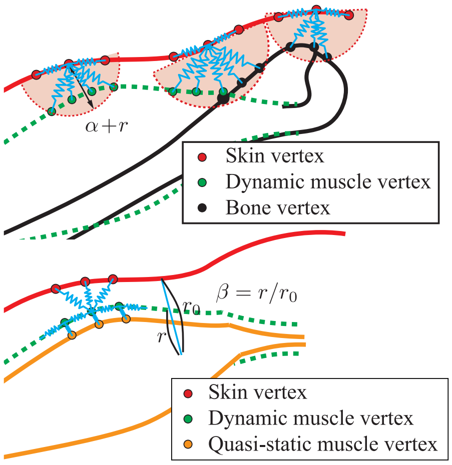


Fig. 4 Spring–damper connections between skin, muscle, and bone vertices, where r is the distance from each skin vertex to the nearest vertex on the muscles or bones. Muscle vertices (orange dots) or bone vertices (black dots) within $\alpha + r$ $(\alpha > 0)$ are connected to the skin vertex by springs and dampers. β represents the average relative thickness of muscles with respect to soft tissue.

The last term, Z_s, represents the difference between the radii at the ends of the two parts and is formulated as

$$Z_s = \sum_{t=1}^{T} (r_1(1,0,t) - r_2(1,0,t))^2 \qquad (10)$$

This term ensures that the two parts are connected smoothly at the boundary. We set the weights for Eqs. (9) and (10) (a_t and a_s) high so that the muscle shape is smooth after optimization ($a_v = 10^0, a_t = 10^2, a_s = 10^2$).

3.3 Muscle and soft tissue dynamic deformation

Having identified the quasi-static muscle model parameters, we can use the same set of parameters to simulate the dynamic deformation of the muscles and the soft tissue for new joint angle data sequences. Here, the shape of the quasi-static muscle model defines the rest shape of the dynamic muscle model from its length and tension. Our method models the bones, the quasi-static muscles, the dynamic muscle, and the skin surface as polygonal surfaces. Let $\mathcal{P}^{\mathrm{s}}$ denote the set of skin vertices, $\mathcal{P}^{\mathrm{qm}}$ the vertices on the quasi-static muscle surfaces, $\mathcal{P}^{\mathrm{dm}}$ the vertices on the dynamic muscle surfaces, and $\mathcal{P}^{\mathrm{b}}$ the vertices on the bone surfaces. In the soft tissue dynamics model (see Fig. 4, top), each skin vertex $p_{\mathrm{s}} \in \mathcal{P}^{\mathrm{s}}$ is connected to:

1. the adjacent skin vertices,
2. a set of nearby muscle vertices: those within the hemisphere whose center is at p_{s} and radius is $\alpha + r$ (here, $\alpha = 2.0$ cm), where r is the distance between p_{s} and its nearest vertex in $\mathcal{P}^{\mathrm{dm}} \cup \mathcal{P}^{\mathrm{b}}$ and α (> 0) is the offset, and
3. the bone vertices included in that hemisphere.

Note that a skin vertex may be connected to multiple muscles. These connections allow the skin to slide over the muscle surface to the extent allowed by the spring stiffness. In the muscle dynamics model (see Fig. 4, bottom), each muscle vertex $p_{\mathrm{dm}} \in \mathcal{P}^{\mathrm{dm}}$ is connected to:

1. the adjacent dynamic muscle vertices,
2. the skin vertices that have been connected to p_{dm}, and
3. the corresponding quasi-static muscle vertex p_{qm}.

As a result, the muscle deforms not only because of the skeleton motion but also based on the change in the quasi-static muscle shape due to muscle activation.

If p_i is connected to p_j via a spring and damper pair, the force applied to vertex p_i from p_j, f_{ij}, is computed by

$$f_{ij} = k_{ij}(||x_{ij}|| - l_{ij})\frac{x_{ij}}{||x_{ij}||} + c_{ij}\frac{(v_{ij})^{\mathrm{T}} x_{ij}}{||x_{ij}||}\frac{x_{ij}}{||x_{ij}||} \quad (11)$$

where x_i and v_i are the position and velocity of vertex p_i, $x_{ij} = x_j - x_i$, $v_{ij} = v_j - v_i$, and k_{ij} and c_{ij} are the stiffness and damping coefficients of the spring connecting vertices p_i and p_j.

The spring and damper coefficients have to be chosen so that the simulation results are stable and realistic. We use different spring and damper coefficients for the springs between skin vertices, between muscle vertices, and between skin and muscle vertices because they have different material properties. The skin has high stiffness whereas the soft tissue has lower stiffness so that it moves more dynamically [31]. The individual spring coefficients are determined based on a few manually selected global spring parameters shown in Table 1. These parameters are selected such that the skin becomes stiffer at locations closer to the bones such as around the elbow and ankle, and more compliant at other places, to emulate the effect of thick soft tissue and muscle layers. To compute the spring coefficients of individual springs, K_{ss}, K_{mm}, and K_{dqm} are scaled by the size of the polygon that vertices belong to, whereas K_{smb} is determined to be inversely proportional to the distance between the skin and bone vertices. In all cases, the damping coefficient is set to $d = \sqrt{k}/50$ for a connection with a spring coefficient of k. While these parameters are manually chosen, it is easy to find a set of values that yield reasonable simulation results.

We add all the forces for each vertex in $\mathcal{P}^{\mathrm{s}} \cup \mathcal{P}^{\mathrm{dm}}$, and compute its acceleration by dividing by its mass. We use the *velocity Verlet integration* method [32] to update the positions and velocities of the skin and muscle surface vertices. This method allows us to achieve high stability at no significant computational cost over the explicit Euler method. Although an implicit integration method [33] would allow a larger time step than explicit integration, that class of method is not suitable for our application because it adds extra damping that diminishes the jiggling of the surface of the skin that we are modeling.

Table 1 Types of springs and their global parameters

Parameter	Springs between vertices of	Value
K_{ss}	Skin–skin	10^4
K_{mm}	Dynamic muscle–muscle	10^5
K_{dqm}	Dynamic muscle–quasi-static muscle	10^2
K_{smb}	Skin–dynamic muscle	10^7

4 Results

The sample skin deformation data used for quasi-static muscle model identification were recorded with 400–450 reflective markers using 16 near-infrared Vicon MX-40 cameras at a rate of 120 fps [6]. We used 300 frames of a slow jump motion sequence to identify the quasi-static muscle model parameters. The identified parameters are used for co-contraction, jumping, running, and impact examples. In contrast, Ref. [20] used several thousands frames to build a static and dynamic deformation model. We need many fewer frames because our model generalizes to different motion types. The motion data used for the simulations (co-contraction, jumping, running, and impact) were recorded with 60 reflective markers using the same motion capture system. We also recorded the contact force between the subject and the ground using two AMTI AccuSway PLUS force plates, each of which can measure the six-axis contact force and momentum at a rate of 1 kHz. An Aurion ZeroWire system with 16 pairs of electrodes was used to capture electromyography (EMG) data for muscles beneath the electrodes at a rate of 5 kHz. The EMG data were processed by mean subtraction, rectification, and a Butterworth bandpass filter with a cut-off frequency of 10–1000 Hz. The force plates and the EMG sensors are optional, although they allow us to consider the internal force and muscle co-contraction respectively. A high-speed video camera was also used for some of the motions to capture dynamic skin deformation at 1 kHz, for ground truth.

4.1 Evaluation of identified quasi-static muscle model

We first demonstrate the advantage of using musculoskeletal and muscle deformation models to obtain the underlying muscle shapes. As mentioned in Section 3, we use skin deformation data captured by markers densely placed on the skin to identify the muscle parameters. The distance between a marker and the closest point on the simulated skin surface indicates how well the muscle deformation model matches the actual skin deformation. We evaluated the quasi-static muscle deformation using two motion sequences: a slow jump motion used for identification, and a slow walk motion for cross validation. The active deformation of the muscle is several millimeters even at the maximum muscle activity, which is much smaller than the deformation caused by the skeleton motion. As shown in Table 2, nonetheless, the means and standard deviations of the distances are smaller with muscle deformation in both motions. Specifically, applying dynamic deformation to the quasi-static muscle deformation model results in larger improvement on average (~2 mm) than the dynamic deformation used in Ref. [20] (~1 mm). We also qualitatively compared our results with Ref. [20], and there is no significant visual difference between them.

4.2 Simulation results

We now show simulated deformation of different parts of the skin for various motions (co-contraction, jumping, running, and impact) to demonstrate our method. These motions are measured from a different subject to the one used for identifying the quasi-static muscle model using a standard marker set, force plates, EMG, and a high-speed camera recording for reference. Video clips of the simulated and recorded skin deformations are shown in the Electronic Supplementary Material (ESM).

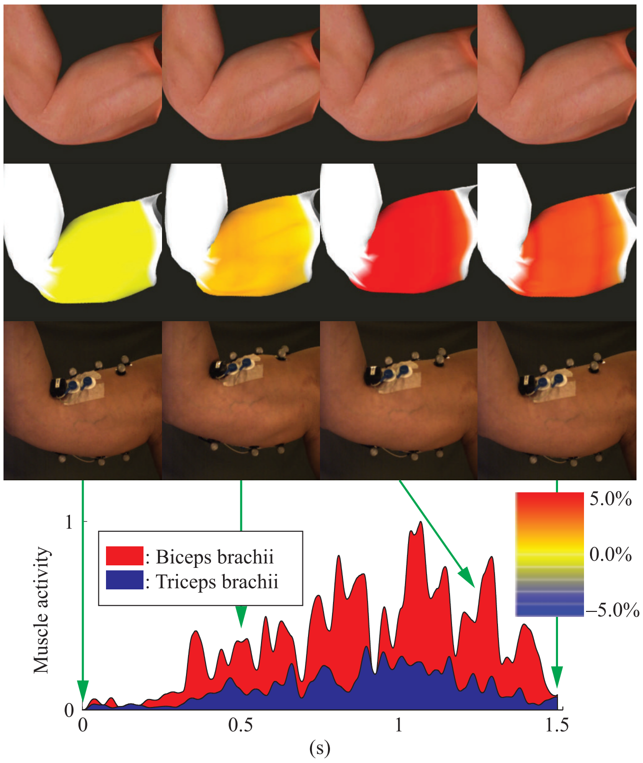


Fig. 5 Skin deformation simulation with muscle co-contraction. Top: skin deformation simulation. Middle: the increase in the upper arm perimeter. Yellow and red colors indicate the amount of muscle bulging. Bottom: activities of the *biceps brachii* and *triceps brachii* measured by EMG.

Figure 5 shows a co-contraction motion that is a body building pose with tensing of the upper arm muscles. The top row represents the simulated skin deformation, the middle row represents the increase in the upper arm perimeter from the initial state, where the color changes from yellow to red as the perimeter increases, and the bottom row represents the corresponding snapshots from the high-speed video camera. The bottom graph shows the normalized activities of the *biceps brachii* and *triceps brachii* obtained by post-processing the EMG data. The result shows that our method effectively simulates the bulging of the muscles during co-contraction, which is mainly detected by the EMG data because co-contraction of antagonistic muscles does not appear as joint motion. The skin jitters observed in the ESM come from noise in the EMG signal, which remains even after Butterworth bandpass filtering.

Table 2 Means and standard deviations of the distances between measured markers and their closest points on the skin surface with and without quasi-static muscle deformation

Motion	With deformation		Without deformation	
	Mean	Std	Mean	Std
Slow jump	16.2 mm	2.3 mm	18.4 mm	2.5 mm
Slow walk	18.6 mm	2.4 mm	20.2 mm	2.5 mm

Figure 6 shows the simulation results for a running motion and Fig. 7 represents the simulation results for a jumping motion. In this motion capture session, we attached several markers in a grid pattern to quantitatively compare the actual and simulated skin deformations.

The simulated skin deformation with our algorithm, the simulated skin deformation without the passive dynamics of muscles and skin, and the corresponding movie from the high-speed camera recording are shown in the ESM. Figure 7 plots the average trajectories of the 16 markers from the measured and simulated skin deformations. Each trajectory is represented in a local coordinate system fixed to the lower leg bone. The blue and red lines represent the measured and simulated

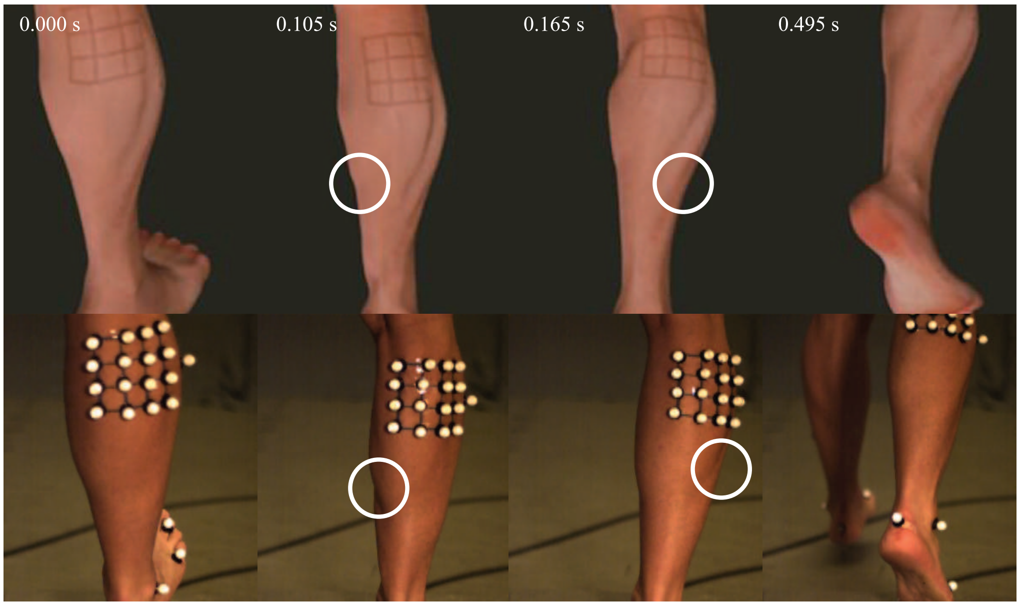


Fig. 6 Skin deformation simulated by our method (top) and corresponding images from high-speed video recording (bottom). Similar skin folds are to be seen, especially in the white circle.

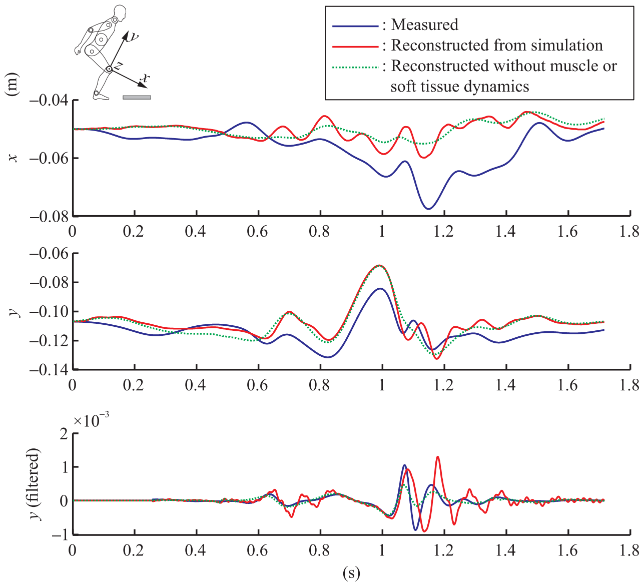


Fig. 7 Trajectories of the measured and simulated markers during the jumping motion. Each trajectory is represented in the local coordinate system fixed to the lower leg bone. The graphs only show four selected markers for clarity, but all other markers show similar trajectories. The three axes are defined as shown in the upper-left.

trajectories of four selected markers, which are on the softest part of the leg. The green dotted lines represent the marker trajectories that are simulated without muscle or soft tissue dynamics. The difference between the simulated skin deformation with muscle and soft tissue dynamics and the one without them is obvious. The third graph represents the y-direction trajectory, post-processed by a Butterworth high-pass filter with a cut-off frequency of 100 Hz. This graph shows that the amplitude, frequency, and duration of the jiggles in the simulated skin deformation with muscle and soft tissue dynamics are similar to those in the measured motion, especially in the y direction just after landing. This effect would not be realized without muscle or soft tissue dynamics.

In Fig. 8, the subject hits an object with his arm. The top row represents the simulated skin deformation, and the middle row represents the corresponding snapshots from a high-speed camera. The high-speed camera shows the skin wrinkle around the elbow caused by the impact, which is also seen in our simulation. We include a parameter that determines the average relative thickness of muscles with respect to soft tissue (β in Fig. 4, bottom) to simulate different body types. The third row of Fig. 8 shows the skin deformation when this parameter set to $\beta = 0.75$, so that the model is 25% less muscular than the one in the top row. The amplitude of the skin jiggle becomes larger than that observed in the muscular model as expected.

5 Discussion

In this paper, we developed a new algorithm for simulating skin deformation in novel motion sequences based on an anatomical model of the musculoskeletal system and a passive dynamics model of soft tissue. This algorithm directly

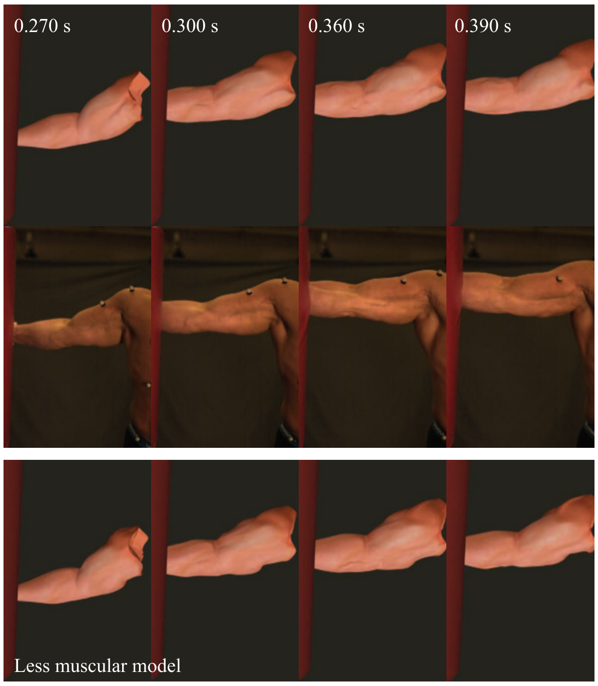


Fig. 8 Skin deformation simulation of an arm with an external impact. The bottom row shows a simulation result with a less muscular model.

generates the skin deformation from skeletal motion data.

- The quasi-static muscle model allows us to compute the quasi-static muscle shape from muscle length and tension information for a wide range of motions. The resulting muscle shape is consistent with the dynamics of the motion because it is based on muscle pathway and tension data obtained by inverse kinematics and dynamics algorithms for a musculoskeletal model.
- The passive dynamics of the soft tissue effectively describes the interaction between the skin and internal bones and muscles. Our model can simulate skin deformations that depend on the underlying structure, such as different jiggling patterns when the skin hits the front side (tibia) and the calf side of the lower leg.
- This algorithm can simulate physiologically realistic skin deformations that are difficult to estimate only from standard motion capture data if EMG data are recorded along with the motion data. An example is muscle co-contraction, which cannot be estimated only from motion data because the activations of antagonistic pairs of muscles do not cause joint motion.
- The model can be generalized to a variety of body types by applying our identification method.

Earlier simulation-based methods tend to handle relatively static body parts and motions, while data-based methods find it difficult to handle novel motions. Our method combines simulation-based and data-driven approaches: simulation allows us to obtain realistic results for a wide variety of motions, while a small set of data can be used to adapt the model to different body types.

We model the muscle and soft tissue dynamics with a mass–spring–damper system. This mass–spring–damper system is based on a realistic body shape created by a modeler, and a simple spindle-like muscle shape is only used to indicate how the detailed skin shape should deform. We chose to use a mass–spring–damper model because finite element method (FEM) would require significantly more parameter tuning for a similar simulation resolution, even if, as shown in Ref. [17] FEM may be applied in principle. Our experimental results show that our simple mass–spring–damper model with well-optimized parameters realizes a skin deformation simulation that corresponds well to experimentally measured data. There are oscillation artifacts seen in our simulation that may be caused by the explicit integration. Applying implicit integration [33] may decrease these artifacts.

Our method has several limitations. The quasi-static muscle model parameter identification process requires some frames of skin deformation data captured with a dense set of markers. Setting up such a motion capture session is cumbersome and may not be possible for every subject. As an alternative to measured skin deformation data, a modeler could provide the skin shapes at a few frames in a motion sequence. It is also possible that modern depth cameras such as the Kinect 2 could be used to provide this data. The identification process described in Section 3.2 could proceed in exactly the same way with this sparser data set. The other limitation is that we identified the quasi-static muscle model parameters assuming that the measured skin deformation data are not affected by the soft tissue dynamics. We chose to use slow motions for identification to minimize this potential coupling. We then manually adjusted the passive dynamics model parameters (the spring and damper coefficients) to obtain desirable skin deformations. It may be possible to identify the two sets of parameters simultaneously using dynamic skin deformation data which might provide more accurate results.

Electronic Supplementary Material Supplementary material is available in the online version of this article at http://dx.doi.org/10.1007/s41095-016-0065-1.

References

[1] Autodesk. Maya 2015. 2015.

[2] Lewis, J. P.; Anjyo, K.; Rhee, T.; Zhang, M.; Pighin, F.; Deng, Z. Practice and theory of blendshape facial models. In: Proceedings of Eurographics 2014—State of the Art Reports, 2014.

[3] Pons-Moll, G.; Romero, J.; Mahmood, N.; Black, M. J. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics* Vol. 34, No. 4, Article No. 120, 2015.

[4] Si, W.; Lee, S.-H.; Sifakis, E.; Terzopoulos, D. Realistic biomechanical simulation and control of human swimming. *ACM Transactions on Graphics* Vol. 34, No. 1, Article No. 10, 2014.

[5] Murai, A.; Takeichi, K.; Miyatake, T.; Nakamura, Y. Musculoskeletal modeling and physiological validation. In: Proceedings of 2014 IEEE International Workshop on Advanced Robotics and its Social Impacts, 108–113, 2014.

[6] Park, S. I.; Hodgins, J. K. Capturing and animating skin deformation in human motion. *ACM Transactions on Graphics* Vol. 25, No. 3, 881–889, 2006.

[7] Vaillant, R.; Barthe, L.; Guennebaud, G.; Cani, M.-P.; Rohmer, D.; Wyvill, B.; Gourmel, O.; Paulin, M. Implicit skinning: Real-time skin deformation with contact modeling. *ACM Transactions on Graphics* Vol. 32, No. 4, Article No. 125, 2013.

[8] Chadwick, J. E.; Haumann, D. R.; Parent, R. E. Layered construction for deformable animated characters. In: Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques, 243–252, 1989.

[9] Tsoli, A.; Mahmood, N.; Black, M. J. Breathing life into shape: Capturing, modeling and animating 3D human breathing. *ACM Transactions on Graphics* Vol. 33, No. 4, Article No. 52, 2014.

[10] Sueda, S.; Pai, D. K. Dynamic simulation of the hand. In: *The Human Hand as an Inspiration for Robot Hand Development*. Balasubramanian, R.; Santos, V. J. Eds. Springer International Publishing, 267–288, 2014.

[11] Lee, S.-H.; Sifakis, E.; Terzopoulos, D. Comprehensive biomechanical modeling and simulation of the upper body. *ACM Transactions on Graphics* Vol. 28, No. 4, Article No. 99, 2009.

[12] Pratscher, M.; Coleman, P.; Laszlo, J.; Singh, K. *Outside-in* anatomy based character rigging. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 329–338, 2005.

[13] Mitsuhashi, N.; Fujieda, K.; Tamura, T.; Kawamoto, S.; Takagi, T.; Okubo, K. BodyParts3D: 3D structure database for anatomical concepts. *Nucleic Acids Research* Vol. 37, D782–D785, 2009.

[14] Zordan, V. B.; Celly, B.; Chiu, B.; DiLorenzo, P. C. Breathe easy: Model and control of simulated respiration for animation. In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 29–37, 2004.

[15] Koch, R. M.; Gross, M. H.; Carlsy, F. R.; von Büren, D. F.; Fankhauser, G.; Parish, Y. I. H. Simulating facial surgery using finite element models. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, 421–428, 1996.

[16] Blemker, S.; Teran, J.; Sifakis, E.; Fedkiw, R.; Delp, S. Fast 3D muscle simulations using a new quasistatic invertible finite-element algorithm. In: Proceedings of International Symposium on Computer Simulation in Biomechanics, 2005.

[17] Webb, J. D.; Blemker, S. S.; Delp, S. L. 3D finite element models of shoulder muscles for computing lines of actions and moment arms. *Computer Methods in Biomechanics and Biomedical Engineering* Vol. 17, No. 8, 829–837, 2014.

[18] Fan, Y.; Litven, J.; Pai, D. K. Active volumetric musculoskeletal systems. *ACM Transactions on Graphics* Vol. 33, No. 4, Article No. 152, 2014.

[19] Anguelov, D.; Srinivasan, P.; Koller, D.; Thrun, S.; Rodgers, J.; Davis, J. SCAPE: Shape completion and animation of people. *ACM Transactions on Graphics* Vol. 24, No. 3, 408–416, 2005.

[20] Park, S. I.; Hodgins, J. K. Data-driven modeling of skin and muscle deformation. *ACM Transactions on Graphics* Vol. 27, No. 3, Article No. 96, 2008.

[21] Wang, H.; Hecht, F.; Ramamoorthi, R.; O'Brien, J. Example-based wrinkle synthesis for clothing animation. *ACM Transactions on Graphics* Vol. 29, No. 4, Article No. 107, 2010.

[22] Delp, S. L.; Anderson, F. C.; Arnold, A. S.; Loan, P.; Habib, A.; John, C. T.; Guendelman, E.; Thelen, D. G. OpenSim: Open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering* Vol. 54, No. 11, 1940–1950, 2007.

[23] Nakamura, Y.; Yamane, K.; Fujita, Y.; Suzki, I. Somatosensory computation for man-machine interface from motion-capture data and musculoskeletal human model. *IEEE Transactions on Robotics* Vol. 21, No. 1, 58–66, 2005.

[24] AIST. Japanese body dimension data 1997–98. 1998. Available at http://www.dh.aist.go.jp/database/97-98/e_index.html.

[25] Yamane, K.; Nakamura, Y. Natural motion animation through constraining and deconstraining at will. *IEEE Transactions on Visualization and Computer Graphics* Vol. 9, No. 3, 352–360, 2003.

[26] Luh, J. Y. S.; Walke, M. W.; Paul, R. P. C. On-line computational scheme for mechanical manipulators. *Journal of Dynamic Systems, Measurement, and Control* Vol. 102, No. 2, 69–76, 1980.

[27] Nakamura, Y.; Yamane, K.; Murai, A. Macroscopic modeling and identification of the human neuromuscular network. In: Proceedings of the 28th Annual International Conference of the IEEE, 99–105, 2006.

[28] Yamane, K.; Fujita, Y.; Nakamura, Y. Estimation of physically and physiologically valid somatosensory information. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2624–2630, 2005.

[29] Coleman, T. F.; Li, Y. On the convergence of interior reflective Newton methods for nonlinear minimization subject to bounds. *Mathematical Programming* Vol. 67, No. 1, 189–224, 1994.

[30] Kardel, T. Niels Stensen's geometrical theory of muscle contraction (1667): A reappraisal. *Journal of Biomechanics* Vol. 23, No. 10, 953–965, 1990.

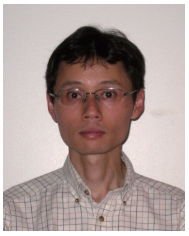
[31] Howell, J.; Chleboun, G.; Conatser, R. Muscle stiffness, strength loss, swelling and soreness following exercise-induced injury in humans. *The Journal of Physiology* Vol. 464, No. 1, 183–196, 1993.

[32] Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of Chemical Physics* Vol. 76, 637–649, 1982.

[33] Baraff, D.; Witkin, A. Large steps in cloth simulation. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, 43–54, 1998.

Akihiko Murai received his B.S. degree in mechanical engineering, and M.S. and Ph.D. degrees in mechano-informatics, in 2003, 2005, and 2009 respectively from the University of Tokyo, Japan. He is currently a research scientist at the National Institute of Advanced Industrial Science and Technology (AIST). Prior to joining AIST, he was a postdoctoral researcher at Disney Research, Pittsburgh, and a project assistant professor at the University of Tokyo. Dr. Murai is a recipient of Young Investigation Excellence Award of RSJ (2016), Robotics-Mechatronics Division Annual Prize (2010), SICE Research Award (2009), and JSPS Research Fellowship for Young Scientists (2009). His research interests include anatomical human modeling, the human neuro-musculoskeletal system, human motion measurement and analysis, and character animation.

Q. Youn Hong received her bachelor degree from Seoul National University, Seoul, Repulic of Korea, in 2006. She is currently working toward a graduate degree at Carnegie Mellon University. Her research interests include character animation and skin deformation.



Katsu Yamane received his B.S., M.S., and Ph.D. degrees in mechanical engineering in 1997, 1999, and 2002 respectively from the University of Tokyo, Japan. He is currently a senior research scientist at Disney Research, Pittsburgh, and an adjunct associate professor at the Robotics Institute, Carnegie Mellon University. Prior to joining Disney, he was an associate professor at the University of Tokyo. Dr. Yamane is a recipient of numerous awards including King-Sun Fu Best Transactions Paper Award and Early Academic Career Award from the IEEE Robotics and Automation Society, and Young Scientist Award from the Ministry of Education, Japan. He has served as an associate editor of *IEEE Transactions on Robotics* and an area chair of the Robotics: Science and Systems Conference, as well as program committee member of various international conferences in robotics and graphics. His research interests include humanoid robot control and motion synthesis, human–robot interaction, character animation, and human motion simulation.

Jessica K. Hodgins is a professor in the Robotics Institute and Computer Science Department at Carnegie Mellon University and a director of Disney Research, Pittsburgh. Prior to moving to Carnegie Mellon in 2000, she was an associate professor and assistant dean in the College of Computing at Georgia Institute of Technology. She received her Ph.D. degree in computer science from Carnegie Mellon University in 1989. Her research focuses on computer graphics, animation, and robotics with an emphasis on generating and analyzing human motion. She has received NSF Young Investigator Award, Packard Fellowship, and Sloan Fellowship. She was editor-in-chief of *ACM Transactions on Graphics* from 2000 to 2002 and ACM SIGGRAPH Papers Chair in 2003. In 2010, she was awarded the ACM SIGGRAPH Computer Graphics Achievement Award.