Evaluating the Physical Realism of Character Animations Using Musculoskeletal Models

Thomas Geijtenbeek[1], Antonie J. van den Bogert[2], Ben J.H. van Basten[1], and Arjan Egges[1]

[1] Games and Virtual Worlds, Utrecht University, The Netherlands
[2] Orchard Kinetics LLC, Cleveland, OH, USA

Abstract. Physical realism plays an important role in the way character animations are being perceived. We present a method for evaluating the physical realism of character animations, by using musculoskeletal model simulation resulting from biomechanics research. We describe how such models can be used without the presence of external force measurements. We define two quality measures that describe principally different aspects of physical realism. The first quality measure reflects to what extent the animation obeys the Newton-Euler laws of motion. The second quality measure reflects the realism of the amount of muscle force a human would require to perform the animation. Both quality measures allow for highly detailed evaluation of the physical realism of character animations.

1 Introduction

Perceived realism of movement is an important component of character animation. However, its subjective nature makes it an impractical quality. Therefore, it is desirable to develop automatic evaluation methods that approximate the perceived realism of motion.

One approach in constructing such evaluation method is through analysis of results from user studies. Another approach is to develop a measure based on statistical analysis of existing realistic motions. Our approach is based on the idea that there exists a relation between the *perceived realism* and the *physical realism* of a motion. The main rationale behind this idea is that all motion we witness in real life adheres to the laws of physics.

There are several levels of accuracy at which physical realism of character animation can be measured. Some methods focus on the center of mass trajectory of a character, while others take into account the dynamics of the individual body parts. However, these approaches are all limited in their ability to describe human motion, since they do not consider the fact that the joint torques that produce motion are the direct result of muscle forces acting on these joints. The amount of torque a muscle can generate depends on both its maximum force as well as its muscle moment arm, which is pose-dependent. In addition, several muscles operate over multiple joints, creating complex dependencies between joint torques. These aspects can only be accurately described through a model of the human musculoskeletal system.

R. Boulic, Y. Chrysantou, and T. Komura (Eds.): MIG 2010, LNCS 6459, pp. 11–22, 2010.

In the field of computer animation, the development of such musculoskeletal models may be considered too specialized and labor-intensive to be worth the effort. However, in biomechanics research, realistic musculoskeletal models are considered an effective tool for conducting research on human motion. As a result, significant effort has been put into the development of such models. The applicability for computer animation is limited though, because musculoskeletal simulation requires motion data to be augmented with force measurements at external contact points. We have developed a method for estimating these external force measurements, thus enabling the use of musculoskeletal models directly with purely kinematic motion data. In addition, we show how a musculoskeletal model can be employed to measure two principally different types of physical realism.

Our first measure evaluates to what degree a character motion obeys the *Newton-Euler laws of motion*, which dictate that changes in linear and angular momentum must be consistent with external forces due to gravity and contact. Examples of animations not obeying these laws are a character hanging still in mid-air, or a character standing straight at a 45 degree angle without falling. Our measure reflects these errors in a way that uniformly applies to both flight and contact stages.

Our second measure evaluates the realism of the muscle force that a human character would require to perform a motion. This measure detects situations where for example a character is moving too fast, or jumping too high. Such motions need not be in conflict with the Newton-Euler laws of motion, but are physically unrealistic because they require an excessive amount of muscle force.

Both quality measures provide objective feedback on the physical realism of an animation, with a level of detail that is unprecedented in the field of computer animation.

2 Related Work

Human character animation has received a lot of attention during the past decades. There are several classes of techniques, each having its own advantages and disadvantages. See Van Welbergen et al. [1] for an overview of different animation techniques.

The amount of research conducted on the perception of physical realism of animation is limited. O'Sullivan et al. [2] define a number of measures regarding perception of physical realism of the motions of solid objects, based on user studies. Reitsma and Pollard [3] observe through user studies that errors in horizontal velocity and added accelerations in human jumping motions are easier observed by humans than errors in vertical velocity and added decelerations. Point light experiments have shown observers can accurately estimate lifted weight [4], as well as pulled weight [5] from point light displays.

Much work has been done on physical realism in generated motions. Safonova et al. [6] show that certain basic physical properties can be conserved during motion transitions. Ikemoto et al. [7] evaluate transitions by foot skating and by evaluating the *zero-moment point*. Ren et al. [8] present a tool to evaluate the realism of animations using a set of statistical models, based on natural example data. Some work makes use of user studies to evaluate the resulting animations. For example, Van Basten and Egges [9] evaluate the relationship between posture distance metrics and perceived realism of motion transitions. Some techniques modify a existing animations to improve the physical realism. For example, Shin et al. [10] show how motions can be adjusted in such a way that they satisfy the *zero-moment point* constraint, as well as some additional physical constraints. Ko and Badler [11] attempt to achieve physical realism in generated walking motions by constraining maximum joint torque and adjusting balance in an inverse dynamics analysis.

Erdemir et al. [12] provide an overview of the different approaches and applications of musculoskeletal modeling. Veeger and Van der Helm [13] provide an exemplary insight in the complexity of musculoskeletal mobility. In recent years, a number of tools have implemented these techniques. Examples of such tools are the commercially available *AnyBody* system [14], the *Human Body Model* [15], and the open source project *OpenSim* [16]. To date, musculoskeletal models have not been used to evaluate computer animations.

3 Method

In this section we will describe in detail how a musculoskeletal model can be used to evaluate the physical realism of a character animation. We will describe how to compute measures representing the error in the *dynamics* of the animation, as well as the error in the amount of *muscle force* a human would require to perform an animation. Our method is independent of the musculoskeletal model or modeling software that is being used; we will therefore omit implementation specific aspects of musculoskeletal modeling.

3.1 Prerequisites

Musculoskeletal Model. Our evaluation method uses a model that incorporates both the skeletal and the muscular structure of the entire human body. Such a full-body musculoskeletal model can formally be described as a tuple $\mathcal{H}$:

$$\mathcal{H} = \{\mathcal{B}, \mathcal{Q}, \mathcal{U}\} \tag{1}$$

The set $\mathcal{B}$ describes the individual body segments of the model. Each segment has specific mass, inertial properties, and a reference position and orientation. The set $\mathcal{Q}$ describes the n kinematic degrees of freedom of the model. These consist of a global position and orientation, plus those defined by the joints in the model. Each joint connects two segments in $\mathcal{B}$, with specific constraints, at a reference position and orientation. The set $\mathcal{U}$ describes the l muscles in the model. For each muscle, it describes the relation between joint angles and muscle moment arm, as well the maximum force the muscle can produce. The exact form of $\mathcal{B}$, $\mathcal{Q}$ and $\mathcal{U}$ depends on the implementation of the model and is not relevant to our evaluation method.

Animation. The input for our method is an animation $\mathbf{A}$, consisting of a sequence of T frames:

$$\mathbf{A} = \{A_1, A_2, ..., A_T\} \quad , \quad A_t = \{q_t, \dot{q}_t, \ddot{q}_t, S_t^l, S_t^r\} \tag{2}$$

where q_t is a pose vector representing the n kinematic degrees of freedom defined in $\mathcal{Q}$; $\dot{q}_t$ and $\ddot{q}_t$ are first and second order derivatives of q_t. S_t^l and S_t^r are polygons representing the ground contact region at frame t, for both left and right foot.

3.2 Quality Measures

Our method produces the following quality measures:

1. The *dynamics error* measure, $\kappa : \mathbf{A} \rightarrow \mathbb{R}$, which is defined as the total amount of external force and moment that is required for the motion to satisfy the Newton-Euler laws of motion. In other words, it describes the amount of 'magical support' required to make a motion dynamically valid. The measure uniformly applies to both contact and flight stages. Force and moment are normalized using body mass and height, enabling comparison between character models of different weight or height.
2. The *muscle error* measure, $\lambda : \mathbf{A} \rightarrow \mathbb{R}$, which is defined as the total amount of excess muscle force required for a character to perform $\mathbf{A}$, normalized by the total maximum force of all muscles in $\mathcal{U}$. The excess muscle force is defined as the amount of force a muscle must produce on top of its maximum capacity. The normalization enables comparison between character models of different strength.

3.3 Method Overview

Both quality measures are calculated per frame, for each $A_t \in \mathbf{A}$. This is done in four consecutive steps, as depicted in Figure 1.

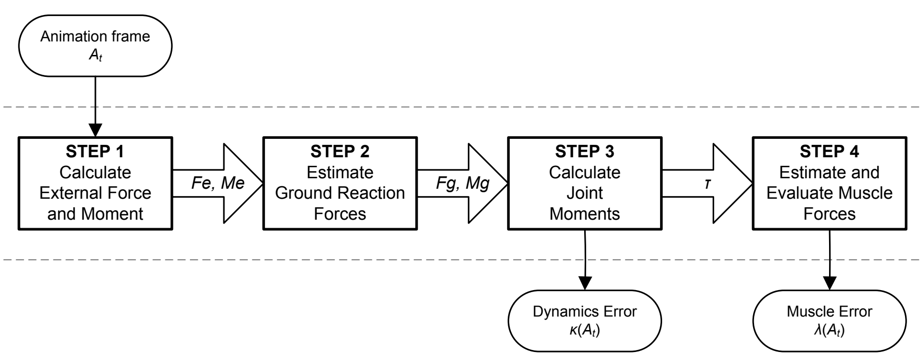

Fig. 1. Overview of the consecutive steps that are performed for each frame A_t

3.4 Step 1: Calculate External Force and Moment

In this step we compute the external force and moment F_e and M_e that act on the character at frame A_t. We do so by solving the dynamics equations of the motion defined by q_t, $\dot{q}_t$ and $\ddot{q}_t$, which can be formulated as:

$$\mathbf{M}(q_t)\ddot{q}_t + \mathbf{T}(q_t)\tau + c(q_t, \dot{q}_t) = 0 \tag{3}$$

where $\mathbf{M}(q_t)$ is an $n \times n$ matrix that describes the pose dependent mass distribution, based on $\mathcal{B}$ and $\mathcal{Q}$. The vector τ contains the (unknown) moments and forces acting on the degrees of freedom in $\mathcal{Q}$. The vector $c(q_t, \dot{q}_t)$ are gravitational, centrifugal and Coriolis forces. $\mathbf{T}(q_t)$ is an $n \times n$ coefficient matrix that has no specific meaning and depends on the form of $\mathbf{M}(q_t)$.

The approach to constructing $\mathbf{M}(q_t)$, $\mathbf{T}(q_t)$ and $c(q_t, \dot{q}_t)$ is not relevant for our method and depends on the modeling software that is employed. The only important aspect for our method is that global position and orientation are part of the dynamics equation. After re-arranging the components in (3) we get:

$$\tau = \mathbf{T}(q)^{-1} \left[\mathbf{M}(q)\ddot{q} + c(q, \dot{q})\right] \tag{4}$$

After solving, the external force and moment acting on the root element of the character correspond to the elements in τ related to global position and rotation, as defined in $\mathcal{Q}$. If F_r and M_r are the force and moment defined in the root coordinate frame, with the origin at r and orientation $\mathbf{R}$, then the external force and moment in the global coordinate frame, F_e and M_e, is defined as:

$$F_e = \mathbf{R}^{-1}F_r \quad , \quad M_e = \mathbf{R}^{-1}M_r + r \times (\mathbf{R}^{-1}F_r) \tag{5}$$

3.5 Step 2: Estimate Ground Reaction Forces

In this step, we estimate the ground reaction force and moment for both left foot, F_g^l and M_g^l, and right foot, F_g^r and M_g^r. The estimate is based on external force and moment F_e and M_e, support polygons S_t^l and S_t^r, and a static friction coefficient μ. We assume that the ground plane is $y = 0$, and that the positive y-axis is pointing upwards. The ground reaction forces are bound by the following constraints:

1. At least one foot must be in contact with the ground before there can be any ground reaction force ($S_t^l \cup S_t^r \neq \emptyset$).
2. The external force F_e must point upwards, otherwise it cannot be attributed to ground contact ($F_{e,y} > 0$).
3. The horizontal component of the ground reaction force must be in agreement with the Coulomb friction model ($||F_{e,xz}|| \leq \mu||F_{g,y}||$).
4. The origin of the ground reaction force and moment for each foot must lie within the support polygon of the corresponding foot.

In our method, we do not employ a dynamic friction model. All ground reaction forces are considered the result of static ground contact.

If constraint 1 and 2 are not met, no external forces or moments will be applied to the character during step 3 ($F_g^l = M_g^l = F_g^r = M_g^r = 0$). Otherwise, we first compute the total ground reaction force and moment, F_g and M_g, and distribute these among both feet afterwards.

We assume that the upward component of F_e can be fully attributed to ground reaction force: $F_{g,y} = F_{e,y}$. We do not limit the maximum amount of ground reaction force in the upward direction, since this is only limited by the amount of pushing force a character can produce. If such a pushing force is not realistic, we will see this reflected in the muscle error $\lambda(A_t)$.

To meet constraint 3, we limit the magnitude of the horizontal component of the ground reaction force, $F_{g,xz}$, based on friction constant μ and the vertical ground reaction force $F_{g,y}$:

$$F_{g,xz} = \begin{cases} F_{e,xz} & \text{if } ||F_{e,xz}|| \leq \mu||F_{g,y}|| \\ \frac{\mu||F_{g,y}||}{||F_{e,xz}||} F_{e,xz} & \text{if } ||F_{e,xz}|| > \mu||F_{g,y}|| \end{cases} \tag{6}$$

The ground reaction moment around y, $M_{g,y}$, is also limited by contact friction between the character and the ground plane. However, since we do not expect large moments around y, we assume that any such moment is automatically countered by static friction: $M_{g,y} = M_{e,y}$.

The ground reaction force and moment applied to each feet must originate from a point on $y = 0$ that lies within the respective support polygon. This imposes a constraint on the ground reaction moment around x and z. To apply this constraint, we first define c_e as the point on $y = 0$ from which F_e and M_e would originate:

$$c_{e,x} = \frac{M_{e,z}}{F_{g,y}} \quad , \quad c_{e,z} = \frac{-M_{e,x}}{F_{g,y}} \tag{7}$$

If c_e lies outside a support polygon S, we define the origin of the ground reaction force, c_g, as the point inside S that is closest to c_e; otherwise: $c_g = c_e$. The ground reaction moments $M_{g,x}$ and $M_{g,z}$ then become:

$$M_{g,x} = -c_{g,z}F_{g,y} \quad , \quad M_{g,z} = c_{g,x}F_{g,y} \tag{8}$$

In cases where only one foot is in contact with the ground we assign F_g and M_g to that foot. In cases where both feet are in contact with the ground, the computation of (8) is performed individually for each foot. The forces and moments are then weighted according to the distance between c_e and support polygons S_t^l and S_t^r. If $d : c \times S \rightarrow \mathbb{R}$ is the distance between a point c and support polygon S, then the weighting factors ω_l and ω_r become:

$$\omega_l = \frac{d(c_e, S_t^l)}{d(c_e, S_t^l) + d(c_e, S_t^r)} \quad , \quad \omega_r = \frac{d(c_e, S_t^r)}{d(c_e, S_t^l) + d(c_e, S_t^r)} \tag{9}$$

where ω_l is the scaling factor applied to acquire F_g^l and M_g^l, and ω_r is the scaling factor applied to acquire F_g^r and M_g^r.

3.6 Step 3: Calculate Joint Moments

Now that we have estimated ground reaction force and moment for both feet, we can add them to the dynamics equation:

$$\tau = \mathbf{T}(q)^{-1}\left[\mathbf{M}(q)\ddot{q} + \mathbf{E}(q)\left[\, {F_g^l}^T \; {M_g^l}^T \; {F_g^r}^T \; {M_g^r}^T \,\right]^T + c(q,\dot{q})\right] \tag{10}$$

where $\mathbf{E}(q)$ is a $12 \times n$ coefficient matrix of which the form is dictated by $\mathbf{M}(q)$. After solving[1] for τ, F_r and M_r correspond to the elements in τ related to global position and orientation of the root element, in the coordinate frame of the root. They are the remaining external force and moment that could not be attributed to ground contact. Their magnitudes are used for the calculation of the dynamics error during frame A_t:

$$\kappa(A_t) = \frac{||F_r(A_t)||}{mg} + \frac{||M_r(A_t)||}{mgh} \tag{11}$$

where m is the subject body mass, g is the gravitational constant and h is the subject height. We employ these variables to ensure that both elements are dimensionless quantities, independent of weight and height.

3.7 Step 4: Estimate and Evaluate Muscle Forces

Each moment $\tau_i \in \tau$ is a result of the muscles acting on the degree of freedom q_i (except for those related to global position and orientation). The relation between the joint moments in τ and the vector of muscle forces, u, can be described as:

$$\tau = \mathbf{D}(q_t)u \tag{12}$$

where $\mathbf{D}(q)$ is a $l \times n$ matrix describing the muscular moment arms of the l muscles in u, derived from $\mathcal{U}$ (given pose q_t). In human characters, the number of muscles is much larger than the number of degrees of freedom ($l >> n$), making the number of solutions for u infinite. However, since our interest is to detect unrealistic muscle usage, we will seek a solution that assumes optimal load sharing between muscles. A common approach to achieve this is to minimize the sum of squared muscle stresses [17]:

$$\underset{u}{\operatorname{argmin}} \sum_{i=1}^{l} \left(\frac{u_i}{u_{max,i}}\right)^2 \tag{13}$$

This optimization is subject to both the moment arm constraints in (12) as well as to $u_i \geq 0$ (muscles cannot produce a negative force). The values for u_{max} are defined in $\mathcal{U}$ and represent the maximum strength of the animated character's individual muscles. Common values of u_{max} are well-document in biomechanics literature and are mostly derived from cadaveric data [17]. If desired, they can be adjusted to match the physique of any virtual character.

[1] When there are no ground reaction forces, equation (10) is equal to (4) and needs not be solved again.

Excess Muscle Force. After we have determined muscle forces u for frame A_t, we wish to evaluate if they are realistic. Our quality measure λ is a dimensionless quantity, defined as the amount of excess muscle force, u_{ex}, normalized by the sum of the elements in u_{max}:

$$\lambda(A_t) = \frac{\sum_{i=1}^{l} u_{ex,i}}{\sum_{i=1}^{l} u_{max,i}} \quad , \quad u_{ex,i} = \begin{cases} u_i - u_{max,i} & \text{if } u_i > u_{max,i} \\ 0 & \text{if } u_i \leq u_{max,i} \end{cases} \tag{14}$$

where $u_{ex,i}$ is the excess muscle force produced by muscle i, based on the maximum force defined by $u_{max,i}$.

3.8 Final Quality Measures

We have shown how to compute quality measures $\kappa(A_t)$ and $\lambda(A_t)$ for any animation frame A_t. There are numerous ways to combine these individual scores to get a score for a full animation clip $\mathbf{A}$. For our research, we have chosen to compute the average of all frames in the animation, which is sufficient when evaluating short animation clips.

4 Experimentation

To demonstrate our method, we investigate the effect of changing character body mass and animation speed on the physical realism of a set of motion captured animations. Changing the body mass of the character reflects a situation where a motion capture performer is much lighter or heavier than the virtual character. Some motions may not be physically realistic in that situation; we expect to see this reflected in muscle error λ, but not in dynamics error κ. Changing the weight of a character should not affect the degree in which its motion is subject to the Newton-Euler laws of motion. We expect that changing animation speed will affect both muscle error λ as well as dynamics error κ.

Our research will be based on the *Human Body Model*, which is a commercially available musculoskeletal model of the entire body [15]. The skeleton consists of 18 segments and 46 kinematic degrees of freedom. The muscle model includes 290 muscles. The inverse dynamics is solved using an approach described in detail by Kane et. al [18]. The squared muscle stress is minimized using the approach described by Xia and Feng [19], which is based on a recurrent neural network and has been developed with real-time performance in mind.

Animations are recorded using a 12 camera Vicon system and a custom 47 marker setup. In order to get smooth first and second order derivatives for q_t, the animation data is filtered using a second order Butterworth filter with a cut-off frequency of 4Hz. Our character ($m = 72$kg, $h = 1.70$m) has performed 16 different motions (see Figure 5). We separately vary the weight and speed multiplier in the range of 0.2 to 3.0 with steps of 0.2. We use a friction coefficient of $\mu = 1$.

5 Results

The results indicate that our measures behave as anticipated. First, there is a clear relation between body weight and muscle error (see Figure 2). This relation is stronger with more labor-intensive motions such as jumping or knee bending, and smaller for the more relaxed motions, such as waving.

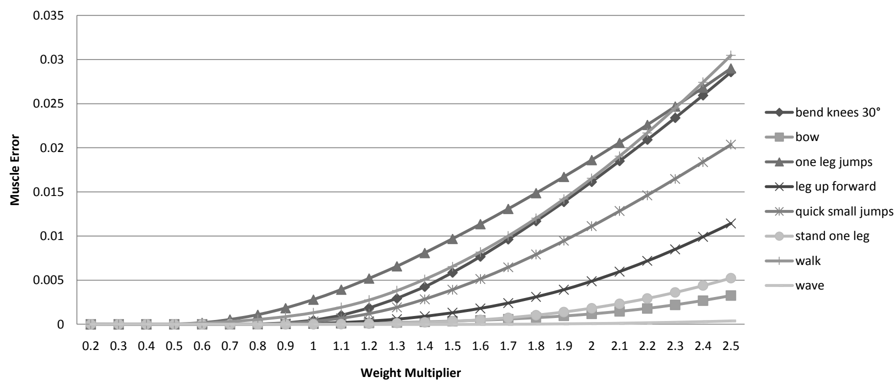

Fig. 2. Muscle error λ, with varying subject weight

We observed no significant change in the dynamics error as a result of weight change, which is in line with our expectations.

The relation between speed change and muscle error is shown in Figure 3. The relation is stronger with highly dynamic motions, such as walking. With such motions, even slowing down the animation results in a slight increase in muscle error.

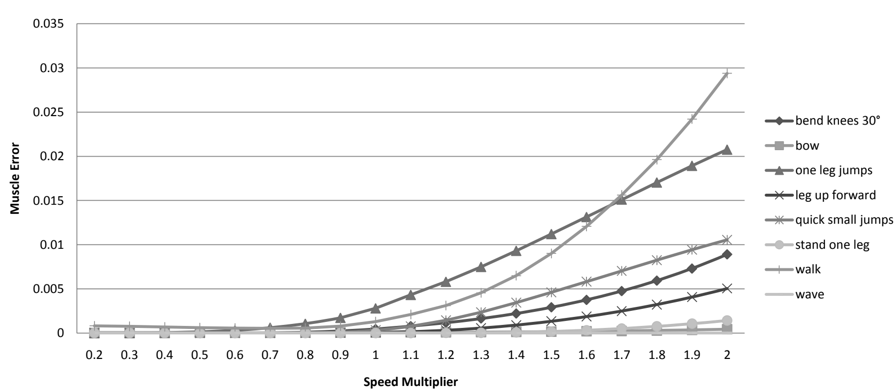

Fig. 3. Muscle error λ, with varying animation speed

The effect of speed change on dynamics error κ is shown in Figure 4. It can be seen that both slowing down and speeding up the animation leads to an increase in dynamics error, but only for dynamic motions such as walking and jumping. This shows it is not dynamically realistic to walk or jump at reduced or increased speed, without adjusting the motion. It can also be seen that increasing speed results in a larger increase in dynamics error than decreasing speed, which is in line with research by Reitsma and Pollard [3].

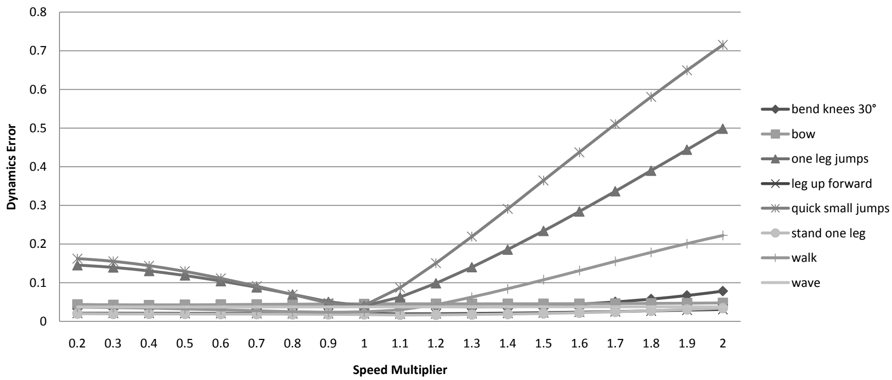

Fig. 4. Dynamics error κ, with varying animation speed

Another interesting observation is that when evaluating a sped up walking animation, there is a relatively high increase in muscle error, compared to the increase in dynamics error. However, when evaluating a sped up jumping motion, there is a high increase in dynamics error, and a relatively low increase in muscle error. One could interpreted this as follows: it is physically more realistic for a muscular animated character to walk with increased speed than to jump with increased speed.

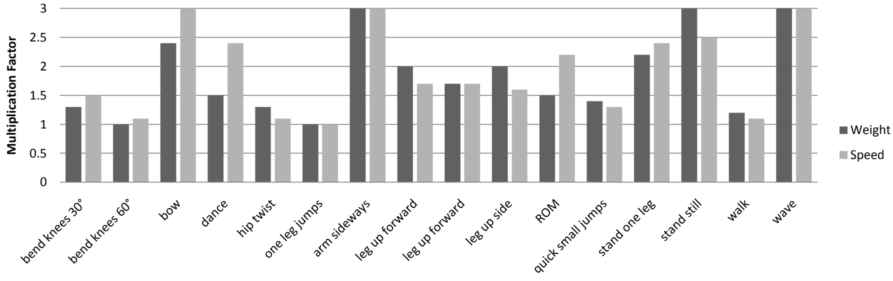

Fig. 5. Maximum weight and speed multipliers for which $\lambda \leq 0.003$. Multipliers are capped at 3; this was the highest multiplier we tested for in our experiments.

Finally, we show the maximum factor by which speed and weight can be increased before the muscle force required to perform a specific animation becomes physically unrealistic (defined by $\lambda > 0.003$). Figure 5 shows these limits for a number of animations.

6 Conclusion, Discussion and Future Work

We have demonstrated a method for evaluating the physical realism of character animations using musculoskeletal models. We have shown how ground reaction forces can be extracted from residual forces and moments. We have also shown how these residuals can be used to uniformly measure the dynamical validity of a motion, according to the Newton-Euler laws of motion. Finally, we have shown how the muscle forces estimated by a musculoskeletal model can be interpreted as a measure for realism of motion. The results of our initial experiments are promising.

We see many applications for our quality measures. For example, the measures can be used to detect good transition points for motion graphs. Also, both measures can be shown as feedback to the animator during interactive motion editing. Muscle error can even be visualized intuitively on a 3D mesh, using color animation.

Preliminary tests indicate that our estimated ground reaction forces correspond well to measured data. We plan to record additional motions with synchronized force plate data to further investigate this claim. We are aware that our method has only limited use in biomechanics research, since it cannot detect situations where both feet exert horizontal forces in different directions. However, for the purpose of measuring physical realism this is not an issue.

Our method of estimating ground reaction forces can still be improved by incorporating a more advanced friction model. At this point, we assume all ground contact is static. We are currently working on incorporating a dynamic friction model, which will make our technique suitable for detecting foot sliding, based purely on dynamics error and muscle error.

For some motions, we have found the muscle error to be above zero, even when neither weight nor speed was adjusted. We expect the reason for this is that some of the maximum muscle forces defined in our model are rather low in comparison to other models. We expect that adjusting these maximum muscle forces to more common values will resolve this issue.

Finally, an important next step in our research would be to evaluate the relationship between our quality measures and the perceived realism via user studies. It would then also make sense to compare our measures to those based on simplified dynamical models, and see if the enhanced accuracy of musculoskeletal models also leads to a more accurate measure of perceived realism.

References

1. van Welbergen, H., van Basten, B.J.H., Egges, A., Ruttkay, Z., Overmars, M.H.: Real Time Animation of Virtual Humans: A Trade-off Between Naturalness and Control. Eurographics - State of the Art Reports, 45–72 (2009)

2. O'Sullivan, C., Dingliana, J., Giang, T., Kaiser, M.K.: Evaluating the visual fidelity of physically based animations. ACM TOG 22(3), 527–536 (2003)
3. Reitsma, P.S.A., Pollard, N.S.: Perceptual metrics for character animation: sensitivity to errors in ballistic motion. ACM TOG 22(3), 537–542 (2003)
4. Runeson, S., Frykholm, G.: Visual perception of lifted weight. Journal of exp. psychology: Human Perception and Performance 7(4), 733–740 (1981)
5. Michaels, C.F., De Vries, M.M.: Higher Order and Lower Order Variables in the Visual Perception of Relative Pulling Force. Journal of Experimental Psychology: Human Perception and Performance 24(2), 20 (1998)
6. Safonova, A., Hodgins, J.K.: Analyzing the physical correctness of interpolated human motion. In: Proc. of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 171–180 (2005)
7. Ikemoto, L., Arikan, O., Forsyth, D.: Quick transitions with cached multi-way blends. In: Proc. of the 2007 Symp. on Interactive 3D Graphics and Games, pp. 145–151. ACM, New York (2007)
8. Ren, L., Patrick, A., Efros, A.A., Hodgins, J.K.: A data-driven approach to quantifying natural human motion. ACM TOG 24(3), 1090–1097 (2005)
9. van Basten, B.J.H., Egges, A.: Evaluating distance metrics for animation blending. In: Proc. of the 4th Int. Conf. on Foundations of Digital Games, pp. 199–206. ACM, New York (2009)
10. Shin, H.J., Kovar, L., Gleicher, M.: Physical touch-up of human motions. In: Proc. of the 11th Pacific Conf. on Comp. Graphics and Appl., pp. 194–203 (2003)
11. Ko, H., Badler, N.I.: Animating human locomotion with inverse dynamics. In: IEEE Computer Graphics and Applications, 50–59 (1996)
12. Erdemir, A., McLean, S., Herzog, W., van den Bogert, A.J.: Model-based estimation of muscle forces exerted during movements. Clinical Biomechanics 22(2), 131–154 (2007)
13. Veeger, H.E.J., Van Der Helm, F.C.T.: Shoulder function: the perfect compromise between mobility and stability. Journal of Biomechanics 40(10), 2119–2129 (2007)
14. Rasmussen, J., Damsgaard, M., Surma, E., Christensen, S.T., de Zee, M., Vondrak, V.: Anybody-a software system for ergonomic optimization. In: Fifth World Congress on Structural and Multidisciplinary Optimization (2003)
15. van Den Bogert, A.J., Geijtenbeek, T., Even-Zohar, O.: Real-time estimation of muscle forces from inverse dynamics. health.uottawa.ca, 5–6 (2007)
16. 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 Trans. on Bio-Medical Eng. 54(11), 1940–1950 (2007)
17. van Der Helm, F.C.T.: A finite element musculoskeletal model of the shoulder mechanism. Journal of Biomechanics 27(5), 551–553 (1994)
18. Kane, T.R., Levinson, D.A.: Dynamics online: theory and implementation with AUTOLEV. Online Dynamics, Inc. (1996)
19. Xia, Y., Feng, G.: An improved neural network for convex quadratic optimization with application to real-time beamforming. Neurocomputing 64, 359–374 (2005)