Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2010)
M. Otaduy and Z. Popovic (Editors)

Editing Dynamic Human Motions via Momentum and Force

Kwang Won Sok[1,2,3] Katsu Yamane[3,2] Jehee Lee[1] Jessica Hodgins[2,3]

[1] Seoul National University
[2] Carnegie Mellon University
[3] Disney Research Pittsburgh

Abstract

We present an integrated framework for interactive editing of the momentum and external forces in a motion capture sequence. Allowing user control of the momentum and forces provides a powerful and intuitive editing tool for dynamic motions. To make a higher jump, for example, the user simply increases the linear momentum in the vertical direction, while our system automatically calculates a motion that maintains both the same landing position and physical plausibility. Our key insight is using trajectory optimization based on normalized dynamics to simultaneously propagate momentum and force space changes. We demonstrate our approach with edits of long sequences of dynamic actions, including kicks, jumps, and spins.

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

1. Introduction

Editing motion capture data is a longstanding problem in computer graphics. Modifications are difficult in part because edits can easily break physical plausibility and because it is easy to make the motion look unnatural given the many subtle details of human motion contained in the original data. The contribution of this work is an integrated framework that allows the user to edit a motion in the momentum and force spaces. We show through examples that this space presents an intuitive and expressive level of control, providing a way to create significant changes to a motion while retaining physical plausibility and detail.

Table 1 shows a number of different motion quantities that could, in principle, be edited. Most previous editing approaches have focused on the motion itself, $m(t)$, because that is what is shown in the rendered animation. In contrast, our approach gives the user the ability to alter the linear and angular momentum and the external forces seen in the behavior. For dynamic motions in particular, this editing space is natural, as it provides control over the impact of a kick, the height of a jump, or the velocity of a spin.

Physical Quantities	Variable Name
motion	$m(t)$
first derivative, linear and angular momentum	$\dot{m}(t)$, $P(t)$,$L(t)$
second derivative, force	$\ddot{m}(t)$,$f(t)$

Table 1: *Motion Analysis.*

The key technical component of our system is the propagation of these changes in the first and second derivative into the motion itself. In general, the position, the velocity and the acceleration of the motion have fixed a relationship characterized by time. However, to incorporate edits from momentum, force and position spaces, the relationship must be changed. To solve this problem, we introduce a trajectory optimization technique based on normalized dynamics, where the time axis can be scaled to realize the edited momentum or force profile while satisfying a position constraint.

In our interface, the user can adjust the shape of the momentum curves, and specify constraints in position and time. For example, Figure 1(a) shows the original momentum profile along the vertical axis for a back flip motion. The profile exhibits three waves which correspond to a preparatory

DOI: 10.2312/SCA/SCA10/011-019

behavior (bending down), jumping and landing. This pattern reflects the way athletes manage momentum to optimize capability. We call this momentum pattern *the strategy*, to perform difficult dynamic motion. We demonstrate that a smooth scaling of the existing strategy provides us with efficient control over the dynamic movement while preserving the strategy.

Just as a user editing in motion space can create an unnatural motion, a user editing in momentum or force space may create a physically impossible motion. Our current implementation does not guard against this eventuality but leaves it up to the user to iteratively adjust the edit when the motion moves outside the space where the motion is perceived physically plausible.

We demonstrate our results on a number of different motion capture sequences. Our system runs at interactive rates even on long sequences, using less than 2 seconds to compute a 5 second sequence. The editing process is made manageable by allowing the user to select the window of time for editing and the axes of momentum or force to be edited. Our results include examples in which a change in momentum is propagated across multiple editing windows, showing functionality that is not generally available in an editing system, except through the ability of the animator to reason about physical motion.

2. Related Work

Researchers have studied techniques for editing a single motion clip to generate rich sets of variations. Motion displacement mapping [Gle98, LS99] has been widely adapted to preserve the original features of input motion data while satisfying a new set of constraints as much as possible. Using structurally similar motion sets, researchers have investigated motion interpolation to obtain continuous spans of high quality motion. From a given motion database, [KG04] developed a method to search for blendable pairs of motion clips. [MK05] introduced a statistical interpolation that optimizes interpolation kernels for parameters at each frame. [SHP04] analyzed the motions produced by interpolation for physical correctness. Recently, researchers combined an interpolation technique with motion graphs to allow parametric interpolation of motion while traversing the graph [SO06, HG07, SH07]. These methods have commonly searched for a way to change the motion trajectory itself.

Velocity and force spaces have also been explored for motion editing. [PBM00] introduced a force-based motion editing technique for locomotion. [MPS06] presented an interactive retiming method in velocity space that preserves physical plausibility and [HdSP07] found a way to guide the time warping with a reference motion. In [TM07], the user acts out the timing information using a mouse or pen-tablet. For complex acrobatic stunts, [MF07] created a physically valid motion editing technique to make the input motion more dynamic. [SL10] presented an animation system to improve the visual quality of motion by correcting physical properties. We extend these approaches by considering changes in velocity, momentum, and external force space in a *unified* framework.

Motion data has been analyzed in combination with a dynamic model to deepen our understanding of the underlying physics driving the motion. [LP, ALP06] used momentum as constraints to preserve the original dynamics of a motion while synthesizing new motion. Physical properties are also used for balancing [SP05, MZS09]. Our work is different in that we are directly editing the momentum profile.

In the robotics literature, the linear relationship between joint velocity and momentum around the center of mass has been derived [KKK*03]. Biomechanics has shown that momentum is being regulated during walking [PE04]. In dance and sports science, researchers have studied momentum to understand how humans perform highly skilled motion [Law86, LB96]. This research suggests that if we preserve the momentum profile of a highly skilled dynamic motion while changing the magnitude, the resulting motion will be a natural variation following the way in which a human improves his or her athletic ability.

3. Overview

We begin with an illustrative example to motivate our technical components and design decisions. In this example, we consider the case where a user wants to edit a back flip motion to make a higher jump. A back flip is a highly dynamic task and thus it is difficult to edit in position space. Although it is possible to make the jump higher by simply changing the root trajectory, it is difficult to maintain the physical plausibility of the original motion.

In our system, the user first selects the *axis of interest*, the direction in which the momentum is to be edited. In this example, the user chooses the vertical axis because the user is interested in making the jump higher. The system then displays the linear momentum in the selected direction as a graph shown in Figure 1(a).

Now the user can scale the original momentum profile using the user interface(Figure 1(b)). The jump is now higher due to the larger linear momentum. However, the edit has a side effect where the character lands in the air because the duration of the motion is unchanged. To solve this problem, the user can direct our system to maintain the original landing height, while respecting the new momentum profile. This additional constraint is enabled by modifying the time scale using the normalized dynamics formulation described in Section 4. The new momentum profile is shown in Figure 1(c), where the duration of the motion is adjusted to land on the floor.

Another problem with naive momentum editing is that the

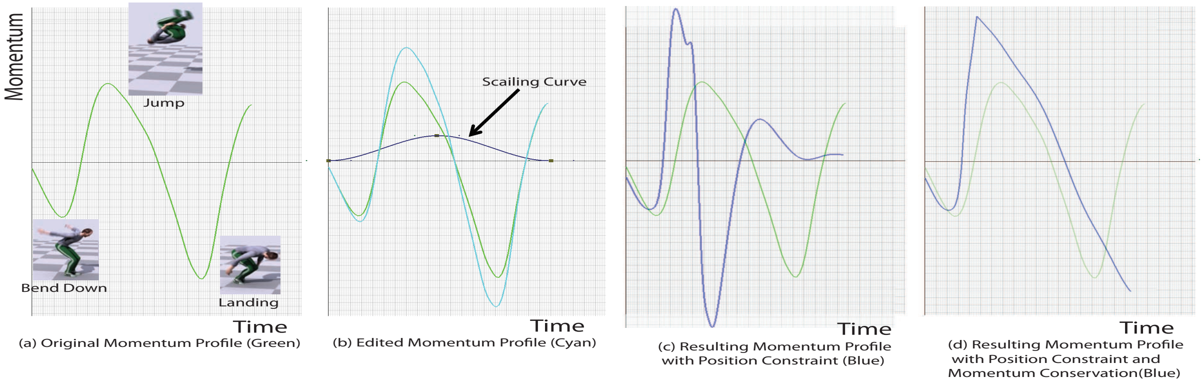


Figure 1: *Original, edited and resulting momentum profiles*

physical plausibility may be lost. In Figure 1(c), notice that the slope of the momentum profile during the flight phase has changed, which implies that gravity is no longer constrained. In the normalized dynamics framework, we can enforce physical rules during a flight phase, such as gravity and conservation of momentum. The momentum profile after correcting the gravity acceleration during the flight phase is shown in Figure 1(d). The user has now obtained a momentum profile that makes a higher jump, maintains the same landing height, and is physically consistent.

The last adjustment the user might want to make is the contact constraint. Because of the different center of mass (COM) trajectory, the contact links may either penetrate the ground or float in the air with the same joint angles as in the original motion. Our system provides an interface to specify active contact constraints and recalculates the pose with inverse kinematics to satisfy those constraints, as described in Section 6.2.

Our system architecture is illustrated in Figure 2. We explain the system architecture focusing on the data flow in three steps: motion analysis, editing constraints and motion synthesis.

4. Motion Analysis

We capture human motions at 120 Hz and take advantage of the high time resolution to calculate the first and second derivatives accurately. After obtaining marker positions from a commercial optical motion capture system, we use the inverse kinematics described in [YN03] to obtain the joint angle trajectories of our character body model with 54 degrees of freedom (DOF) and inertial properties similar to humans. We then obtain the six-axis momentum profile using the model and the joint angle and velocity data from the motion capture data. Note that the estimated momentum profiles are not perfectly accurate because of the model and measurement errors.

5. Editing Constraints

Our system provides the user with an interface for specifying constraints in the momentum/force space and/or in the Euclidean space.

Selecting Editing Window. First, the user chooses the editing window within the sequence of motion.

Selecting Axis of Interest. The user is allowed to edit one axis at a time. Highly dynamic human motions often exhibit one or two principal axes of movement in which case the user can edit multiple axes sequentially. Choosing just one axis allows us to present a simple interface to the user.

Momentum/Force Scaling. The user should choose either momentum or force as the editing channel. Scaling via external force is more appropriate for handling a slipping-intensive scene, for example, as the sliding-on-a-cart example in the supplemental video. In other cases, we usually select momentum.

Time/Position Constraints. The user may want to keep the original duration of motion or final COM position with altered momentum profile. If the user wants to modify both, the user needs to edit the motion in two steps.

Additional Constraints. Modifying COM trajectory often leads to violations of the ground contact constraints. We allow the user to specify any position constraints on a certain body part. In our experiments, we use this functionality for keeping ground contact constraints.

6. Motion Synthesis

Motion synthesis takes the user-edited momentum profiles as well as other conventional constraints such as reference

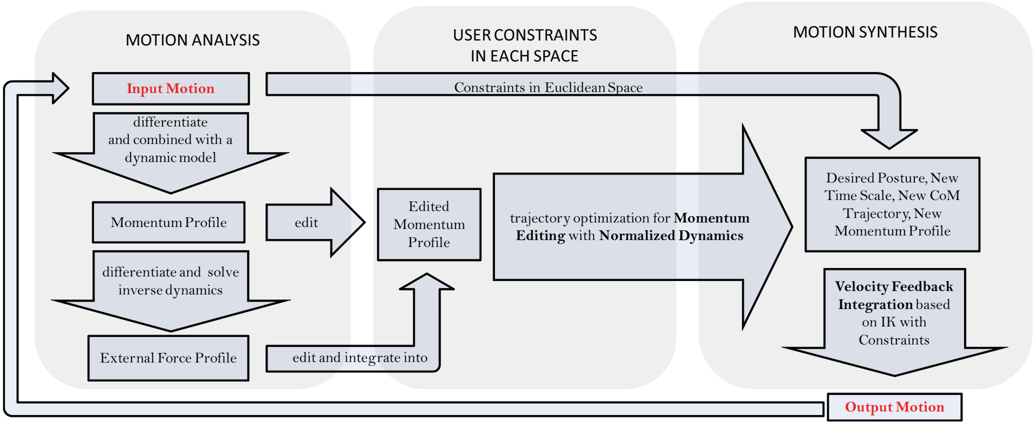

MOTION ANALYSIS
USER CONSTRAINTS IN EACH SPACE
MOTION SYNTHESIS
Input Motion
Constraints in Euclidean Space
differentiate and combined with a dynamic model
Momentum Profile
edit
Edited Momentum Profile
trajectory optimization for Momentum Editing with Normalized Dynamics
Desired Posture, New Time Scale, New CoM Trajectory, New Momentum Profile
differentiate and solve inverse dynamics
Velocity Feedback Integration based on IK with Constraints
External Force Profile
edit and integrate into
Output Motion

Figure 2: *System Architecture.*

joint angles and body part position constraints, and calculates a new motion consistent with those constraints. Motion synthesis consists of two main components. We first obtain the trajectory of the center of mass (COM) using *normalized dynamics* that considers the momentum and position constraints at the same time. The new COM trajectory is then used for velocity feedback integration, which calculates the whole-body joint trajectories considering all constraints. We describe these two components in the following two subsections.

6.1. Trajectory Optimization

Trajectory optimization is used to calculate new COM trajectory consistent with the edited external force or momentum profile. The technique is based on the concept called *normalized dynamics*, where the dynamics are described with a time-scaling parameter. Modifying the external force or momentum profiles usually changes the duration of the motion making the scaling essential.

6.1.1. Normalized Dynamics

The total external force and moment applied to a character and the character's total linear and angular momenta are related by the following equations:

$$f = \dot{P} + mg \qquad (1)$$

$$n = \dot{L} \qquad (2)$$

where f and n are the total external force and moment respectively, P and L are the total linear and angular momenta respectively, m is the total mass of the character, and g is the gravity acceleration vector. The linear momentum of the character is calculated from the velocity of the center of mass, $\dot{x}$, as

$$P = m\dot{x}. \qquad (3)$$

A similar relationship can be found for the angular momentum and velocity:

$$L = I\omega \qquad (4)$$

where ω is the angular velocity when the whole character is considered as a single rigid body and I is the moment of inertia of the entire character model around its center of mass. Notable differences between Eqs.(3) and (4) are that the moments of inertia change over time as the posture changes, and that ω cannot be integrated to obtain the orientation. The latter problem does not appear when we only consider planar motions and the rotation is limited to a single, fixed axis. In this case, the equation becomes

$$L_* = I_*\dot{\theta}_* \qquad (5)$$

where $* = \{x, y, z\}$ denotes the axis of interest.

We derive the *normalized dynamics* by normalizing time by a scaling factor c defined as

$$c = \frac{dt}{d\tau} \qquad (6)$$

where t is the time and τ is the normalized time, or *phase*. Using c, the time derivative is converted to the phase derivative by

$$\frac{d}{dt} = \frac{1}{c}\frac{d}{d\tau}. \qquad (7)$$

Equations (1)–(3), (5) are converted to the phase space as

$$\phi = \frac{dP}{d\tau} + \mu\gamma \qquad (8)$$

$$\nu = \frac{dL}{d\tau} \qquad (9)$$

$$P = \mu\frac{dx}{d\tau} \qquad (10)$$

$$L_* = H_*\frac{d\theta_*}{d\tau} \qquad (11)$$

where the normalized quantities are defined as

$$\begin{aligned}\phi &= cf\\ \nu &= cn\\ \mu &= m/c\\ H_* &= I_*/c\\ \gamma &= c^2 g.\end{aligned}$$

Equations (8) and (10) yield the equations to calculate the linear momentum and position:

$$P(\tau) = P_0 - \mu\gamma\tau + \int_0^\tau \phi(\sigma)d\sigma \tag{12}$$

$$x(\tau) = x_0 + \frac{1}{\mu}\int_0^\tau P(\sigma)d\sigma \tag{13}$$

Note that we assume that c is constant during the integration period and therefore μ and γ are also constant.

Similarly, the angular momentum can be obtained from the external moment as

$$L(\tau) = L_0 + \int_0^\tau \nu(\sigma)d\sigma. \tag{14}$$

The equation for the rotation angle can be obtained only for the single-axis rotation case as

$$\theta_*(\tau) = \theta_{*0} + \frac{1}{H_*}\int_0^\tau L_*(\sigma)d\sigma. \tag{15}$$

We can calculate the external force and momentum for a given motion sequence. We use the time of the original motion as the normalized time, thus $c = 1$. Note that c may not always be 1 for the edited motion.

6.1.2. Force Scaling

We now derive the equations when the external forces along the three axes are independently scaled by constant scaling factors from the original values, i.e.

$$\hat{f}(t) - mg = S(f(t) - mg) \tag{16}$$

$$S \triangleq \begin{pmatrix} s_x & 0 & 0\\ 0 & s_y & 0\\ 0 & 0 & s_z\end{pmatrix} \tag{17}$$

Note that the difference from the gravitational force, rather than the force itself, is scaled.

If we simply plug the new external force into the standard equation of motion, the new COM trajectory would be uniquely determined. Thanks to the normalized dynamics, however, we can still modify the trajectory by modifying the time scale from c to another constant value $\hat{c}$. The normalized mass and gravity become $\hat{\mu}$ and $\hat{\gamma}$ accordingly. The normalized forces are scaled as

$$\hat{\phi}(\tau) - \hat{\mu}\hat{\gamma} = S(\phi(\tau) - \mu\gamma). \tag{18}$$

The new momentum in the normalized time becomes

$$\begin{aligned}\hat{P}(\tau) &= \hat{P}_0 - \hat{\mu}\hat{\gamma}\tau + \int_0^\tau \hat{\phi}(\sigma)d\sigma\\ &= \hat{P}_0 + S\int_0^\tau(\phi(\sigma) - \mu\gamma)d\sigma\\ &= \hat{P}_0 + S(P(\tau) - P_0) \qquad (19)\end{aligned}$$

where we have used Eq. (12) to eliminate the integration term. Eq. (19) leads to

$$\hat{P}(\tau) - \hat{P}_0 = S(P(\tau) - P_0) \tag{20}$$

which means that the increase of the momentum is also scaled by the same ratio as the external force. We can then obtain the new COM trajectory by

$$\begin{aligned}\hat{x}(\tau) &= \hat{x}_0 + \frac{1}{\hat{\mu}}\int_0^\tau \hat{P}(\sigma)d\sigma\\ &= \hat{x}_0 + \hat{c}S(x(\tau) - x_0) + \frac{\hat{c}}{m}(\hat{P}_0 - SP_0)\tau \qquad (21)\end{aligned}$$

where we have used Eq. (13) to eliminate the integration term.

6.1.3. Editing Force/Momentum Scale

In this section, we discuss the usage of Eq. (21) for force/momentum editing. We assume the following editing scenario:

- The user edits one of the six axes by giving a scaling factor s for the force or momentum.
- For the edited direction, the user can choose to keep either the duration of motion (time constraint) or final COM position (position constraint).
- For other directions, the user may choose to constrain the final COM position. In that case, the force scales of those directions will be determined automatically based on the time scale determined from the constraint on the edited direction. The user may subsequently choose to edit another axis.

In the rest of the section, we will use the single-axis version of Eq. (21):

$$\hat{x}(\tau) = \hat{x}_0 + \hat{c}s(x(\tau) - x_0) + \frac{\hat{c}}{m}(\hat{P}_0 - sP_0)\tau \tag{22}$$

where x can be the COM position in any of the x, y or z direction. Also note that Eq. (22) can be applied to the rotational axes by substituting

$$\begin{aligned}x, x_0 &\leftarrow \theta, \theta_0\\ m &\leftarrow I\\ P_0 &\leftarrow L_0.\end{aligned}$$

The time constraint is trivially realized by setting $\hat{c} = 1$. To realize the position constraint, we set $\tau = T$ and $\Delta x_T \triangleq \hat{x}(T) - \hat{x}_0 = x(T) - x_0$, and solve for $\hat{c}$:

$$\hat{c} = \frac{m\Delta x_T}{sm\Delta x_T + (\hat{P}_0 - sP_0)T} \tag{23}$$

where T is the duration of the motion.

6.1.4. No-Contact Case

When there is no external force, the motion during the flight phase is determined by the initial momentum and time scale. Substituting $\phi = 0$ to Eq. (12) and using Eq. (13), we obtain the COM trajectory as

$$x(\tau) = x_0 + \frac{c}{m} P_0 \tau - \frac{1}{2} g c^2 \tau^2. \tag{24}$$

If the difference between the final and initial positions is constrained to be Δx_T, we can calculate a unique time scale. We consider the scalar version of Eq. (24) to separate the direction with gravity from others. In the direction of gravity, we have a second-order equation for c:

$$\frac{1}{2} g T^2 c^2 - \frac{P_0 T}{m} c + \Delta x_T = 0 \tag{25}$$

which can be solved as

$$c = \frac{v_0 + \sqrt{v_0^2 - 2g\Delta x_T}}{gT} \tag{26}$$

where $v_0 = P_0/m$.

In the directions without gravity, the new time scale is obtained by

$$c = \frac{\Delta x_T}{v_0 T}. \tag{27}$$

6.2. Whole-Body Motion Synthesis

The formulation presented in the previous section gives a new COM trajectory consistent with the given force or momentum profile. The optimization can be performed at multiple sections of a time window in which case the time scale becomes a function of time, $c(t)$. In fact, we can even perform the optimization for each frame to realize the same COM trajectory in phase.

We now describe how to synthesize a new whole-body motion considering the COM trajectory as well as other constraints, including the position of a body part to maintain contact and reference joint angles to make the pose similar to the original motion. For this purpose, we employ a velocity feedback integration technique, where we calculate the joint velocities at each frame considering the constraints and integrate them until the end of the time window.

The joint velocities are calculated by Jacobian-based inverse kinematics (IK) algorithm [YN03]. In a normal IK setting, joint velocity calculation and integration are repeated iteratively to obtain a single pose. In our case, we run the process once for each frame and the result of integration becomes the pose of the next frame.

We set reference posture constraints for each body segment except the root to guide the result to reference motion. From the edited momentum and/or external force profiles, we have a new COM trajectory to follow. Note that the number of constraints from these two sets of constraints is already similar to the number of DOFs. We set additional position constraints such as ground contact constraints in some examples.

Joint velocities $\dot{q}$ are calculated using the following linear equation:

$$v = J\dot{q} \tag{28}$$

where v is a set of feedback velocities to maintain the constraints and J is the Jacobian matrix of the constraints. If the number of constraints is greater than the DOF of the character, which is usually the case in our setting, Eq.(28) becomes overconstrained and all constraints may not be satisfied. The IK algorithm described in [YN03] allows any number of constraints without numerical difficulties by grouping the constraints into two priorities, where the high-priority constraints are strictly satisfied and the low-priority constraints are satisfied as long as they do not violate the high-priority constraints. We can tune the relative strength of the constraints by changing their weights and feedback gains. It is not difficult to find a set of gains that produces reasonable results. In our experiments, we used 100-200 as the default parameters for the high-priority gains. The exact value of the gains does not affect the quality in most example. However, when the gains are too small, say 0.1, the resulting motion appeared squash. To avoid this, we used relatively high gains. See [YN03] for details on the algorithm.

The feedback velocity for the COM constraints is

$$v_{COM} = k_{COM}(\hat{x} - x) \tag{29}$$

where $\hat{x}$ and x are the desired and current COM positions respectively and k_{COM} is a positive constant. The Jacobian matrix for the COM constraint, J_{COM}, has been formulated by [SN02].

The feedback velocity for a body part position constraint is given by

$$v_P = k_P(\hat{p} - p) \tag{30}$$

where $\hat{p}$ and p are the desired and current positions of the body part and k_P is a positive constant. The Jacobian matrix for this type of constraint can also be calculated in a number of ways, see for example, [OS84].

We also consider the joint angles to make the edited motion as close as possible to the original motion. The feedback velocity is

$$v_J = k_J(\hat{q} - q) \tag{31}$$

where $\hat{q}$ is the reference joint angles typically taken from the original motion capture data. The corresponding Jacobian matrix is simply an identity matrix. We include the root orientation as part of the joint angle to maintain the original posture when a linear momentum has been changed, while

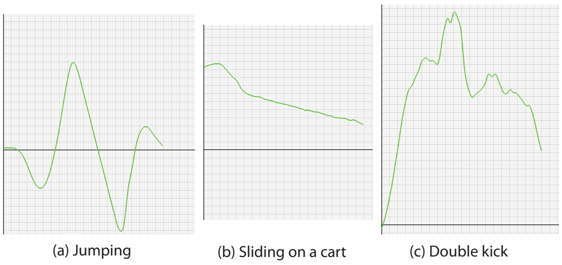


Figure 3: *Example momentum profiles. The x and y axes are phase of input motion and magnitude of momentum repectively. (a) Linear momentum along vertical direction (b) Linear momentum along heading direction (c) Angular momentum around vertical direction*

the root position is excluded because it conflicts with the COM trajectory constraint.

We integrate the joint velocity $\dot{q}$ while considering the current time scale $c(t)$ by

$$q(t+\delta t/c(t)) = q(t) + \int_{t}^{t+\delta t/c(t)} \dot{q}(s)ds. \qquad (32)$$

In our implementation, we treat all constraints as low-priority constraints except for the contact position and COM constraint. We also adapt the effect of each low-priority constraint depending on the location in the time window. In particular, we increase the gain for the reference joint angle constraint towards the end of the window because the pose at the end of the window should match the original in order to smoothly connect to the rest of the motion.

7. Experimental Results

Plotting the Momentum Profile and Editing. Our system is fast enough to be used as an interactive tool. The system automatically suggests all 6 axes to the user for convenience and then, the user chooses axis of interest to be edited. From the displacement of the root joint position, the system calculates the heading direction of the motion parallel to the ground. We suggest the vertical direction as a secondary direction because it aligns with gravity. By applying the cross product between these two, we get the other horizontal axis. We set up three more axes for the angular momentum around the linear momentum axes.

After selecting one of the axes to be edited, the user can see the magnitude of momentum in the graph window (Figure 3). We summarize the constraints used in our experiments in Table 2.

Double Kicks. We choose angular momentum around the vertical axis as our axis of interest. If we constrain time with increased momentum, we get a larger displacement for the root joint orientation around the vertical axis. If we constrain the final orientation, we get faster movement. We can combine these two constraints one after another. We can also make the motion less dynamic by scaling down the momentum profile. If the user wants to edit both time and position constraints, the user may perform two consecutive edits as in Figure 4(b)(RightBottom). In this way, we can synthesize various double kicks from a single input motion within a few minutes.

Playing Around in an Office. We turn a sequence of not-so-dynamic motion into a number of more dynamic moves as an example of momentum propagation and force scaling. This functionality allows us to edit a sequence of dynamic motions in a more physically meaningful way. The sliding on a cart motion has too little linear momentum at the beginning, and slides too slowly. Therefore, we cannot significantly increase the dynamics of the scene even if we scale down the external force during sliding. To fix this problem, we select a time window prior to the sliding motion, increase the linear momentum in the window, and make use of the increased linear momentum along the heading direction to create a much faster sliding motion. We adjust the momentum profile by using force scaling instead of momentum scaling, so that we can have indirect control over the frictional force from the environment.

8. Discussion

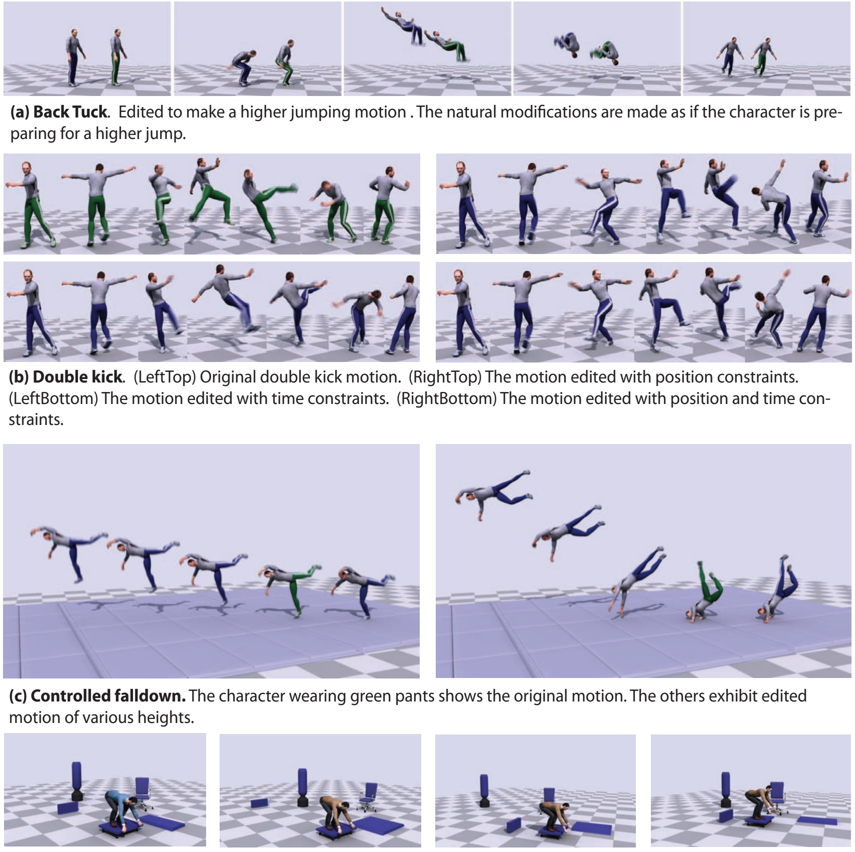
In this paper, we present an algorithm for editing in momentum and force space. This technique is both effective and intuitive for dynamic motions where the momentum and forces have significant magnitude and define the style of the performance. Although this system is complete enough to produce a number of different edits for a variety of behaviors, it does have limitations in the editing capability provided to the user and the ways in which physical feasibility are enforced.

The currently allowable edits in force and momentum space can be extreme and the user is not forced to preserve the momentum pattern of the original motion. Exploring the right balance between control for the user and the maintenance of physical plausibility is an interesting research question. The space should perhaps be constrained by environmental variables, such as gravity and static and dynamic friction. In the current implementation, a very rapidly rotating character will still be able to plant a landing after a back flip, even if the resulting horizontal forces on the ground would be well outside the friction cone.

To informally assess how intuitive the system is we have asked three artists who are familiar with editing human motion data using Maya to use our system. After 30 minutes of training, they understood the concept of the editing in momentum space and could produce good results for a simple examples such as jumping higher. However, because the implementation is still preliminary, they had trouble with more

Input Motions	Constraints
double kicks	angular momentum | position and/or time
controlled falling down	linear momentum | position | momentum conservation during flight
back tuck	linear momentum | time | ground contacts | momentum conservation during flight
jumping higher and landing in the air	linear momentum | time | ground contacts
jumping higher and landing on the ground	linear momentum | position | momentum conservation during flight
running before sliding	linear momentum | position
sliding on a cart	external force | time
jumping over a panel	linear momentum | time
rotating on a chair	angular momentum | time
kicking the punching bag	angular momentum | position

Table 2: *The constraints used in the included examples.*



(a) Back Tuck. Edited to make a higher jumping motion . The natural modifications are made as if the character is preparing for a higher jump.

(b) Double kick. (LeftTop) Original double kick motion. (RightTop) The motion edited with position constraints. (LeftBottom) The motion edited with time constraints. (RightBottom) The motion edited with position and time constraints.

(c) Controlled falldown. The character wearing green pants shows the original motion. The others exhibit edited motion of various heights.

(d) Playing around in an office. (LeftMost) Original motion. (Others) The motion is edited in various ways. See the supplemental video for the animation. The objects are relocated to match the new motion.

Figure 4: *Experimental results.*

complex editing scenarios such as combination of multiple editings and ground contact constraints. The sophisticated edits would require more training and/or an improved user interface. The informal comments that we received indicated that they felt that suggested that the system would be useful as a Maya plug-in especially when they need to create super hero motion.

We would like to experiment with more detailed editing in both motion and momentum/force space. We predict that it might be useful to have control on a per limb basis. A per joint basis might be too fine a level of granularity and cause the motion to no longer contain human-like strategies of inter-limb coordination. As for finer control on the time axis – the user already selects the window for editing so he/she can control the scale there. Editing in motion space could be performed using existing techniques although the physical realism of the motion would not necessarily be maintained. Such a system would likely be required if we are to add editing of quasi-static and stylized motions to the functionality of the system.

Acknowledgements

This work was partly supported by MKE/MCST strategic technology development program (Project No. 2008-F-033-02). The first author was partly supported by Korea Research Foundation Grant funded by the Korean Government (MOEHRD)(KRF-511-2007-1-D00219).

References

[ALP06] ABE Y., LIU C. K., POPOVIĆ Z.: Momentum-based parameterization of dynamic character motion. *Graph. Models 68*, 2 (2006), 194–211. 2

[Gle98] GLEICHER M.: Retargetting motion to new characters. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), ACM, pp. 33–42. 2

[HdSP07] HSU E., DA SILVA M., POPOVIĆ J.: Guided time warping for motion editing. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), pp. 45–52. 2

[HG07] HECK R., GLEICHER M.: Parametric motion graphs. In *Proceedings of Symposium on Interactive 3D Graphics and Games* (2007), pp. 129–136. 2

[KG04] KOVAR L., GLEICHER M.: Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics 23*, 3 (2004), 559–568. 2

[KKK*03] KAJITA S., KANEHIRO F., KANEKO K., FUJIWARA K., HARADA K., YOKOI K., HIRUKAWA H.: Resolved momentum control: humanoid motion planning based on the linear and angular momentum. In *Intelligent Robots and Systems, 2003.* (Oct. 2003), vol. 2, pp. 1644–1650 vol.2. 2

[Law86] LAWS K.: The physics of dance. MacMillan Publishing Company (August 1986). 2

[LB96] LEES A., BARTON G.: The interpretation of relative momentum data to assess the contribution of the free limbs to the generation of vertical velocity in sports activities. In *Journal of Sports Sciences* (December. 1996), vol. 14, pp. 503–511. 2

[LP] LIU C. K., POPOVIĆ Z.: Synthesis of complex dynamic character motion from simple animations. In *ACM Transactions on Graphics 21, 3 (2002)* (New York, NY, USA), ACM, pp. 408–416. 2

[LS99] LEE J., SHIN S. Y.: A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of SIGGRAPH 99* (1999), pp. 39–48. 2

[MF07] MAJKOWSKA A., FALOUTSOS P.: Flipping with physics: motion editing for acrobatics. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2007), Eurographics Association, pp. 35–44. 2

[MK05] MUKAI T., KURIYAMA S.: Geostatistical motion interpolation. *ACM Transactions on Graphics 24*, 3 (2005), 1062–1070. 2

[MPS06] MCCANN J., POLLARD N. S., SRINIVASA S.: Physics-based motion retiming. *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006). 2

[MZS09] MACCHIETTO A., ZORDAN V., SHELTON C. R.: Momentum control for balance. In *ACM Transactions on Graphics, 28, 3 (2009)* (New York, NY, USA, 2009), ACM, pp. 1–8. 2

[OS84] ORIN D., SCHRADER W.: Efficient Computation of the Jacobian for Robot Manipulators. *International Journal of Robotics Research 3*, 4 (1984), 66–75. 6

[PBM00] POLLARD N., BEHMARAM-MOSAVAT F.: Force-based motion editing for locomotion tasks. In *IEEE International Conference on Robotics and Automation, 2000.* (2000), vol. 1, pp. 663–669 vol.1. 2

[PE04] POPOVIC M., ENGLEHART A.: Angular momentum primitives for human walking: biomechanics and control. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on* (Sept.-2 Oct. 2004), vol. 2, pp. 1685–1691 vol.2. 2

[SH07] SAFONOVA A., HODGINS J. K.: Construction and optimal search of interpolated motion graphs. *ACM Transactions on Graphics 26*, 3 (2007), 106. 2

[SHP04] SAFONOVA A., HODGINS J. K., POLLARD N. S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics 23*, 3 (2004), 514–521. 2

[SL10] SHAPIRO A., LEE S.: Practical character physics for animators. In *IEEE Computer Graphics and Applications* (2010). 2

[SN02] SUGIHARA T., NAKAMURA Y.: Whole-body cooperative balancing of humanoid robot using COG Jacobian. In *Proceedings of IEEE/RSJ International Conference on Inteligent Robots and Systems* (2002), pp. 2575–2580. 6

[SO06] SHIN H. J., OH H. S.: Fat graphs: Constructing an interactive character with continuous controls. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), pp. 291–298. 2

[SP05] SULEJMANPASIC A., POPOVIĆ J.: Adaptation of performed ballistic motion. *ACM Transactions on Graphics 24*, 1 (2005), 165–179. 2

[TM07] TERRA S. C. L., METOYER R. A.: A performance-based technique for timing keyframe animations. *Graphical Models 69*, 2 (2007), 89–105. 2

[YN03] YAMANE K., NAKAMURA Y.: Natural motion animation through constraining and deconstraining at will. *IEEE Transactions on Visualization and Computer Graphics 9* (2003), 352–360. 3, 6