

Data-driven Interactive Balancing Behaviors

KangKang Yin¹, Dinesh K. Pai^{1,2}, and Michiel van de Panne¹

¹University of British Columbia, Canada

²Rutgers, State University of New Jersey, USA

Abstract

Animations driven by motion capture data are typically limited by their kinematic nature. In this paper, we describe a new data-driven approach for producing interactive dynamic balancing behaviors for an animated character. The result is a character that can interactively respond to single or multiple pushes in various directions and of varying magnitudes. A database of captured responses to pushes is used to create a model that supports hip, arm, and stepping strategies for balance recovery. An interactive push is modelled as a force impulse, which is then used to compute a momentum-based motion index in order to select the most appropriate recovery motion from the database. The selected motion is then adapted in order to provide a response that is specifically tailored to the given force impulse while preserving the realism and style of the original motion. Based on a relatively small motion database, our system is effective in generating various interactive balancing behaviors, for single and multiple pushes.

Keywords: character animation, motion capture, balance behaviors

1. Introduction

Kinematic methods and dynamic methods each have their own merits as means for creating realistic interactive character animations. Kinematic methods driven by motion capture data offer realism while being limited in their generality. Methods that explicitly model dynamics have the promise of being more general, but require solving difficult control or optimization problems in order to deal with the active nature of human motion. In this work, we apply a data-driven approach to a restricted class of dynamic interactions with the environment, namely that of a character receiving unexpected pushes. We show that a data-driven approach to this problem yields a simple and effective solution for this case, and we expect that a similar methodology may be useful in constructing models of other types of dynamic interaction with the environment.

Human balance movements are classified as *semi-automatic* movements in the movement sciences. These are non-trivial motor tasks that are learned in early childhood. Once acquired, we can quickly select an appropriate motor plan from a repertoire of balancing motions in response to unexpected external perturbations. We develop an analogous

data-driven approach for interactive balancing wherein a dynamic perturbation triggers the selection of a suitable balancing motion from a motion database, which is then further tailored for the particular perturbation.

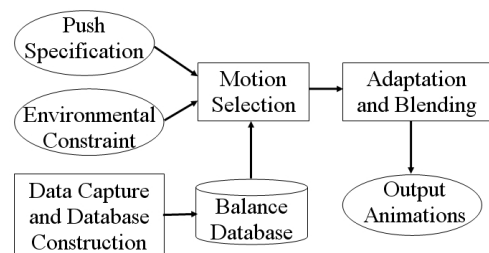


Figure 1: System Overview

Figure 1 is an overview of the system. We first construct a motion database (Section 4) which is populated with motion data from a series of experimental pushes applied to a subject. In the online phase, a user can then interactively apply a push of a desired magnitude and direction. A motion selection algorithm (Section 5) then selects a matching mo-

tion from the database according to the momentum perturbation, as well as possible constraints from the environment that may dictate what kind of balance strategies can and cannot be used. Once a motion is selected, it is adapted in order to more precisely match the current perturbation strength and direction. Finally, the result is blended with the ongoing motion in order to generate a plausible animation, while at the same time satisfying existing foot-ground contact constraints (Section 6).

2. Related Work

Human balancing behaviors are of interest to a broad range of research areas, including biomechanics, robotics, and computer animation.

2.1. Biomechanics

The biomechanics and motor control community has been studying the human balance problem continually and extensively [HvDdB*97, HHSC97, HR98, HR99, SCW01, MMF03, Pat03]. However, a thorough understanding of the underlying neural control mechanism and musculoskeletal dynamic system has yet to be found. A common starting point has been the study of natural body swaying and postural adjustments under small perturbations. For larger perturbations, various balancing strategies have been observed and analyzed, including the hip strategy, the arm rotation strategy, as well as change-of-support strategies such as stepping.

Two important concepts related to balance control are commonly used in the biomechanics literature. Humans regulate their body muscles to adjust the net Ground Reaction Force (GRF) between foot and ground to regain balance. The point of application of the GRF is constrained to be within the foot support polygon, and the GRF can only push but not pull the foot. The Center of Pressure (CoP) is defined as the point on the ground where the resultant normal GRF acts. It is mathematically equivalent to the Zero Momentum Point (ZMP) (see Section 2.3) commonly used in the robotics literature ([Gos99]).

2.2. Robotics

Biped balancing is a topic of considerable interest in robotics [Gos99, SPH03, GK04, PHH04, AG05, KLKK05]. For humanoid robots, reference trajectory tracking based on regulation of various balance indices, such as momentum, ZMP, or CoM, forms the framework for the majority of the work ([KKI02, LCR*02, IAK03, KKT*00, KKK*03a, KKK*03b]). This framework is intended to deal with small perturbations caused by imperfections or by noise in the underlying dynamical system and the environment. Balance recovery from large perturbations remains a goal to be achieved. However, this work provides a solid starting point for an improved understanding of balance behaviors.

2.3. Computer animation

Balance is an essential feature that makes human character animations realistic. Spacetime optimization [WK88, Coh92] and simulation methods ([HWBO95, PBM00, FvdPT01, ZH02, FP03]) emphasize the physical plausibility of motions, by incorporating dynamic constraints into the modelling and simulation procedure. These methods are usually computationally expensive and do not account for style. In contrast, motion editing techniques use measured motion data and emphasize rapid or interactive generation of new motions ([PSS02, AF03, KGP02, LCR*02, LWS02, AFO03]). Recent developments are beginning to combine the advantages of both techniques (e.g., [ALP04, SHP04]), either to reduce the computation and increase the realism of synthesized motions, or to generalize motion capture data while preserving dynamic effects.

Recent results based on explicitly characterizing momentum patterns [LP02, ALP04] are particularly relevant to our own work. To preserve the dynamic behavior of the input motion, a spline-based parameterization is introduced for the linear and angular momentum patterns of a captured motion. A family of similar motions are first optimized off-line, and then interpolated online to generate new motions. In this work, we also treat the momentum characteristics as a significant tool for modelling dynamic motions. We focus on applying fast and simple transformations that preserve the momentum characteristics as much as possible. Our approach works for momentum patterns such as the example shown in Figure 2, while the work of [LP02] uses a more constrained parameterization of momentum pattern tailored to a particular class of motion.

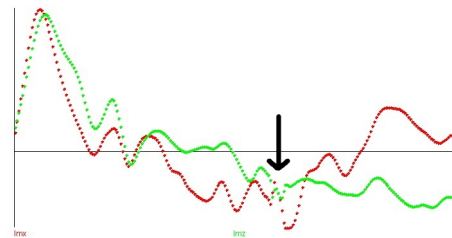


Figure 2: Total linear momentum of an arm-rotation balance strategy plotted as a function of time. The arms rotate three times before the subject recovers. The red and green curves are the X and Z components of the linear momentum. A Y-up right hand coordinate system is used. The broken segment labelled by the black arrow is due to motion capture noise.

The Zero Moment Point (ZMP) is a balance index that is widely used in robotics, first introduced by [VJ69]. The ZMP is defined as that point on the ground at which the net moment of the inertial forces and gravity forces has no component along the horizontal axes. For physically plausible motions, the ZMP always remains inside, or on the

boundary of, the support polygon, even when the character is falling. Key-framed animations and captured motions can both violate the ZMP constraint due to errors and noise introduced during the modelling, capture and processing procedures. [TSK00, SKG03] each describe an interesting approach that modifies voluntary motions to make them satisfy the ZMP constraint exactly at all times. The reasoning behind using the ZMP in computer graphics is that while computer-generated motions may well be physically impossible, they look better when they are physically plausible.

We do not attempt to edit motions based on their ZMP constraints for several reasons. First, it is not obvious how to associate a computed ZMP pattern with the multimodal (hip, arm, stepping) balance strategies that we wish to support. Second, we do not know the perturbation forces for the balance-recovery motions that we capture, and so we cannot precisely reconstruct the ZMP motion for these motions. Last, the computations for the ZMP are based on accelerations and are thus sensitive to noise in the captured data. Additionally, it requires a model of the mass and inertial properties of the links, which may be difficult to estimate accurately. The issues in computing ZMP locations from motion data also arise in the work of [TSK00, SKG03], where terms are dropped from the ZMP equation in order to obtain a simpler and more robust estimate of the ZMP. In our work, we design transformations to respect ZMP constraints, rather than explicitly computing the ZMP.

Most recently, [AFO05] describe experiments with pushing response animation synthesis. They parameterize push impulses as 6 dimensional vectors (3 dimensions for the location on the body being pushed and 3 for the velocity of this location). Then a user-trained oracle selects a response motion from the database that will give the best visual quality when the necessary kinematic transitions and deformations are applied. They use a large database, consisting mainly of stepping balance responses, and can deal with unconstrained pushes on the upper body. Dynamic properties of the motions, such as the momentum characteristics and the ZMP constraint, are not directly taken into account.

3. Background

3.1. Balance Strategies

We capture hip, arm, and stepping balance strategies for populating our motion database.

- Hip strategy: characterized by body sway resembling a two link inverted pendulum by bending at the hip. This is typically elicited during perturbations that are large, on compliant support surfaces, or when the task requires a large or rapid shift in CoM.
- Arm strategy: characterized by rapid arm rotation. This is typically elicited during perturbations too large to recover by just lower limb strategies, and with environment

constraints (i.e., standing on a ledge) or instructions that prevent stepping.

- Stepping strategy: characterized by asymmetrical loading and unloading of the legs to move the base of support under the falling CoM. This is typically elicited when there are no surface or instructional constraints, or when the perturbations are extremely large and in-place balance is not possible.

Although every strategy has its own kinematic and dynamic characteristics, they form a continuum of “mixed” strategies in the postural and balance response space, instead of being “discrete” strategies. In classifying the type of strategy that is observed in a given motion, we order the balance strategies as follows, from weakest to strongest: (1) hip strategy; (2) arm strategy; and (3) stepping strategy. We classify a motion according to the strongest balance strategy that is displayed. For example, if the subject rotated her arms as well as stepping, we label the motion as being a stepping strategy.

3.2. Momentum

Linear and angular momentum (about the CoM), denoted as P and H_c , can be calculated kinematically as follows:

$$\begin{aligned} P &= \sum P_i = \sum m_i \mathbf{v}_i = m \mathbf{v} \\ H_c &= \sum H_i + \sum \mathbf{c} \mathbf{c}_i \times P_i \\ H_i &= I_i \omega_i \end{aligned} \quad (1)$$

where m is the total mass, \mathbf{c} is the location of the CoM, \mathbf{v} is the velocity of the CoM. Quantities with subscript i represent the quantity for just the i^{th} link of the rigid body system, e.g., ω_i is the angular velocity of the i^{th} segment, I_i is the moment of inertia of the i^{th} segment. The vector pointing from \mathbf{c} to \mathbf{c}_i is denoted $\mathbf{c} \mathbf{c}_i$. We estimate mass-inertia parameters by approximating each limb of the motion capture subject with cylinders of matching sizes. We then estimate a uniform density to match the total mass of the virtual character with that of the motion capture subject.

The rate of change of linear and angular momentum can be calculated as follows:

$$\begin{aligned} \dot{\mathbf{f}} + \mathbf{f}_{perturb} + m\mathbf{g} &= \dot{P} \\ \mathbf{c} \mathbf{s} \times \mathbf{f}_{perturb} + \mathbf{c} \mathbf{p} \times \mathbf{f} + (0, t_y, 0)^T &= \dot{H}_c \end{aligned} \quad (2)$$

where the external perturbation force $\mathbf{f}_{perturb}$ applies at point \mathbf{s} . $\mathbf{c} \mathbf{s}$ is the vector pointing from the CoM \mathbf{c} to \mathbf{s} . Resultant GRF \mathbf{f} applies at the CoP \mathbf{p} , and t_y is the resultant moment around the vertical axis exerted by GRF. We only consider the instability about horizontal axes. That is, we do not model the rotation about the vertical axis that a strong off-axis push might introduce, such as a push to the right shoulder.

For all our captured balancing motions, we apply perturbations to a neutral stance pose. The pushes are applied at a consistent height and towards the central vertical axis of the

character. We model the perturbations as impacts, that is, impulses of short duration that therefore do not allow the subject to have actively modulated the GRF during the course of the perturbation. During a perturbation, we thus assume that the GRF continues to go through the CoM and counteracting gravity, and therefore having no effect about horizontal axes for both \dot{P} and \dot{H}_c . This allows for the following approximation:

$$\begin{aligned} \mathbf{f}_{\text{perturb}} &\cong \dot{P} \\ \mathbf{cs} \times \mathbf{f}_{\text{perturb}} &\cong \dot{H}_c. \end{aligned} \quad (3)$$

We can thus use Equation 1 to compute the maximum momentum of the system during a perturbation and its subsequent recovery. This provides an estimation of the perturbation momentum injected by the external force, with the assumptions that Equation 3 holds, and the balancer only removes momentum from the system.

3.3. Push Impulse Parametrization

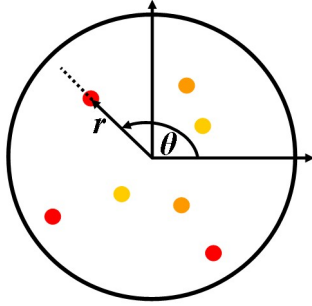


Figure 3: *Parametrization of push impulses. A similar plot also serves as a visualization tool for the motion database, with the colour indicating the type of balance strategy that is invoked for a push of the given magnitude and direction.*

We parameterize the push impulse and the corresponding captured balancing motion using the polar coordinate plot shown in Figure 3. The polar angle θ represents the perturbation direction. The polar radius r represents the perturbation magnitude. We estimate the direction and magnitude of perturbation impulses according to Equation 1. The color represents the balance strategy: hip strategies are shown in yellow; arm strategies are shown in orange; stepping strategies are pink.

4. Motion Capture and Database Construction

We captured six sessions of balancing motions, each being 1 to 2 minutes in duration, at 60Hz, using a Vicon optical motion capture system. The balancing subject was asked to balance naturally, and not to fake vigorous balancing motions on a small perturbation. An assistant who served as a “pusher” was asked to push through the subject’s central

vertical axis, so as to not to cause unnecessary momentum about the vertical axis. He was asked to deliver the pushes at a consistent height towards the middle of the torso, in order to make the relationship between the linear and angular momentum consistent across all perturbations. The pusher was requested to vary the magnitude and direction of the applied pushes. The pushes were to be with sufficient force to make the subject visibly move but sufficiently small so that the subject would not fall.

We hand-segmented the resulting data into distinct balance-motion clips and hand-classified the balance strategy invoked by the subject. A total of 66 motion clips form our motion database. The data is passed through a series of automatic post-processing stages. For stepping motions, foot-ground contacts and breaks are detected using a position/velocity threshold algorithm. The foot-ground contact information is used later by the motion selection and motion blending algorithm to favor plausible matches and smooth transitions.

We parametrize the motions using the polar coordinates described in Section 3.3. For motions involving stepping, however, an interesting artifact is that a separate sideways shift of the momentum occurs in order to shift the center of mass (CoM) over the stance foot in preparation for stepping. Our momentum estimate produced by Equation 1 will therefore implicitly include this secondary effect, thereby potentially introducing an error in terms of estimating the magnitude and direction of the momentum perturbation applied by the push. Empirically and intuitively, the shift of CoM from the start to the end of the stepping conforms well with the real perturbation direction. Hence, we choose the direction of the CoM shift as the polar angle for stepping strategies. The perturbation magnitude estimated from Eq. 1 is then projected onto this new direction. For consistency, we can apply the same direction and magnitude correction for in-place balance strategies as well (i.e., hip and arm strategies). The CoM shifts for in-place balance motions agree very well with the perturbation direction from Equation 1. Thus, there is no harm in applying this operation for in-place motions, while it improves the estimate for stepping motions.

Finally, we exploit symmetry to increase the number of motions in the database even further. As shown on the left of Figure 6, motion 1 is mirrored with respect to the sagittal plane to produce motion 2. For all motions classified as being in-place balance strategies, we also interpolate motion 1 and 2 in order to get a motion that is neutral with respect to the sagittal plane, e.g., motion 3.

5. Motion Selection

Given a user-specified momentum perturbation and possible environmental constraints, we first employ a nearest neighbor (NN) algorithm to find the best matching balance motion for subsequent transformation and blending. Because

our motion database is sparse and sample motions are not evenly distributed, we choose to use a NN algorithm over other potentially more sophisticated scattered data interpolation techniques.

Environmental constraints, such as that of a character standing on a balance beam can be used to further constrain the strategies that the character can adopt. We first enumerate the allowed strategies and then apply the NN algorithm among the motions that use the allowed strategies. We now describe in detail how the NN algorithm chooses a matching motion based on an input perturbation momentum.

For every pose Ω in the database, we define a recoverable momentum $r(\Omega)$ that this pose can regain balance from, i.e., there is a trajectory in the database such that the character returned to a stable neutral pose successfully from this pose with momentum $r(\Omega)$. Usually we just define $r(\Omega)$ to be the actual momentum that this pose has in the database motion projected onto the perturbation direction (due to the same reason we explained in Section 4). It is not necessarily the true upper bound for the recoverable momentum for this pose, but is what we can deal with given only the data found in the database. For every motion snippet, denote the initial pose as Ω_0 , and the pose where the character reaches the maximum momentum as Ω_m . Then we set

$$r(\Omega_i) = r(\Omega_m) \quad \text{for } i \in [0, m] \quad (4)$$

The reason for this modification is that Ω_i 's are further away from the balance constraint boundary than Ω_m is, so they should recover from at least momentum $r(\Omega_m)$. Ideally if we could deliver the perturbation instantly during our experiments, m would be zero and this operation would not be necessary.

We also define a relationship operator \succcurlyeq between two momenta such that $P_1 \succcurlyeq P_2$ denotes that the difference between their aligned polar radius Δr_a is greater or equal to zero. For a stepping strategy, $\Delta r_a = r_1 - r_2$. For in-place strategies, $\Delta r_a = \frac{r_1 \sin \theta_1}{\sin \theta_2} - r_2$ as illustrated in Figure 6 (left). Section 6.2 will describe in detail how we align momenta of different directions for different strategies.

We begin by defining the pose distance between two poses as $dp(\Omega, \Omega')$, which we take to be a weighted sum of the squared joint angle differences. Denote the distance between two momenta as $dm(P, P')$. We will take this to be a weighted sum of the squared differences of their polar coordinates, properly normalized and aligned.

$$dm(P, P') = \left(\frac{\Delta \theta}{\Delta \theta_{max}} \right)^2 + k \left(\frac{\Delta r_a}{\Delta r_{max}} \right)^2 \quad (5)$$

We take $\Delta \theta_{max} = \pi$, and Δr_{max} be the largest momentum in the motion database. Weight k is determined experimentally.

Given an input momentum P applied to pose Ω with mo-

mentum $m(\Omega)$, we wish to find a new pose Ω' by

$$\min_{\Omega'} (w * dp(\Omega, \Omega') + dm(r(\Omega'), m(\Omega) + P)) \quad (6)$$

s.t. $r(\Omega') \succcurlyeq m(\Omega) + P$

where weight w is determined experimentally. $m(\Omega)$ is defined as:

$$m(\Omega) = \begin{cases} 0 & \Omega = \Omega_0 \\ r(\Omega) & \text{otherwise} \end{cases} \quad (7)$$

Motion selection upon an initial push from a static neutral pose, and motion selection for a second response upon another push during the balancing of a first push are treated in almost the same way. The only difference is that in Equation 7 $m(\Omega) = 0$ when $\Omega = \Omega_0$. I.e., when the balancer is in the static neutral pose, the momentum should be zero because no push has been delivered yet.

If there are N motions in the database, and K frames for each motion, then a naive search algorithm for Ω' would be NK . However, we can reduce the computation by pruning away entire motions. First, only a handful of motions in the database are close to the direction of $m(\Omega) + P$, and other motions need not to be considered at all. Second, if, for a motion J of K_J frames, $r(\Omega_{J0}) \not\succcurlyeq m(\Omega) + P$, then motion J is immediately eliminated, since $r(\Omega_{J0}) \succcurlyeq r(\Omega_{Ji})$, for all $i \in [0, K_J]$. For a momentum perturbation applied to a neutral pose, the lookup is further simplified because the best matching pose defaults to Ω_{J0} , i.e., the 0th frame of motion J , due to the fact that we captured all the motions from a common starting neutral pose. It is only for a second push occurring during the course of recovering from a first push that we need to search within a motion.

6. Motion Adaptation

After a best matching motion is selected from the motion database, we need to adapt this motion to make up the difference of perturbation momenta between the user input and the selected database perturbation. The basic principle is that we only transform the motion in the direction that poses less of a challenge to the underlying dynamic system, in terms of dynamic constraints (ZMP and torque limits) and kinematic constraints (joint angle limits). For example, if a motion is captured under a perturbation momentum ΔP , then an adaptation of this motion is very likely to handle a perturbation momentum $\Delta P/2$. In contrast, extrapolating this motion to recover from a perturbation momentum $2\Delta P$ can lead to harmful results, without formulating all the constraints into the algorithm explicitly. The principle of only allowing constraint-respecting transformations motivates our definition of the \succcurlyeq relationship operator in the motion selection algorithm (Section 5).

6.1. Scaling

If the perturbation magnitudes do not match, we apply a scaling transformation on the selected motion. We represent

joint rotations using exponential maps \mathbf{q} , i.e., the rotation axis scaled by the rotation angle about the axis ([Gra98]). Scaling of \mathbf{q} by a scalar α amounts to scaling of the rotation magnitude.

$$\mathbf{q}' = \alpha \mathbf{q} \quad (8)$$

where $0 < \alpha \leq 1$, with corresponding scaling of the relative root joint displacements as well. We further apply an empirically-derived linear warp of the motion in time, observing that smaller perturbations allow a faster recovery. We can speed up the scaled down motions, ideally according to some space-time relationships for various balance strategies. We use a simple relationship that warps the time by $(1 + \varepsilon/2)$ while scaling in space by $(1 - \varepsilon)$. To achieve the original momentum scale factor, we solve for ε according to:

$$(1 - \varepsilon) * (1 + \frac{\varepsilon}{2}) = \alpha \quad (9)$$

Figure 4 shows the effects of the scaling operation on linear momentum.

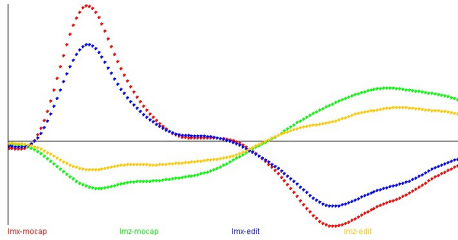


Figure 4: The X and Z components of the linear momentum with respect to time of the original (red and green) motion and the scaled motion (blue and yellow).

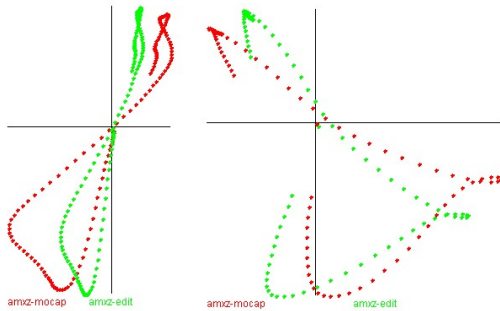


Figure 5: Angular momentum about CoM of the original (red) and rotated (green) motions. Horizontal axis is the X component. Vertical axis is the Z component. Left: in-place rotation. Right: stepping rotation.

6.2. Rotation

If the direction of perturbations do not match, we perform a rotation transformation on the selected motion. If the perturbation is rotated about the vertical axis by an angle, the output motion and its momentum should also rotate. For example, if we push someone forward, she steps forward. If we push her to the left, then we expect her to step to the left. Of course we can push her in such a way that makes her to rotate to the left and then step forward. However as described in Section 4, we do not consider perturbation momentum around the vertical axis. Thus by rotation we do not refer to the character rotating her facing direction, but rather we refer to rotating the perturbation momentum and its corresponding balance motion.

We apply separate algorithms for in-place motion rotation and stepping rotation, as will be described shortly. Hence, the calculation for the aligned magnitude difference between two momenta Δr_a will also be different. Figure 5 shows an example of the effects of rotation operation on angular momentum. Linear momentum transforms similarly.

6.2.1. In-place rotation

For in-place balance motions, the foot support polygon is fixed and not radially equidistant. In Figure 6 (left), if we want to rotate motion 1 towards the sagittal plane X, we interpolate motion 1 and its sagittal-plane-neutral motion 3 (as described in Section 4). Given a new user input, the green dot, its magnitude difference Δr_a from the database sample 1 is the distance marked by the bracket in the illustration.

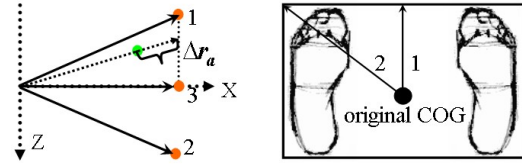


Figure 6: Left: In-place rotation illustration. Right: Foot support polygon.

The motivation for the projection is as follows. By the principle of respecting constraints during motion adaptations, only rotations to less constrained directions are allowed. In Figure 6 (right), 1 is the direction of the original motion with a shorter radius, and suppose now we rotate it to direction 2. Since the support polygon has longer radius in direction 2, the dynamical system has a larger stability margin and hence a longer time to remove perturbation momentum before it hits the ZMP boundary (i.e., foot support polygon boundary). Thus it is reasonable to say that the rotated new motion is physically valid given the same amount of perturbation momentum. However, we cannot justify the validity of a motion rotated from direction 2 to 1 because we may lose the ability to recover balance due to the more

constrained nature of the support polygon in this direction. If motion 2 already hits the boundary of various constraints, such as the ZMP constraint, then the rotated motion now in direction 1 will likely be implausible physically because the ZMP will shoot outside of the foot support polygon. That being said, we can still rotate a motion from direction 2 to 1, if we take into account the loss of recoverability when calculating Δr_a during our motion selection phase (Section 5), by properly projecting the database momentum onto direction 1.

6.2.2. Stepping rotation

For stepping motions, the goal is to change the stepping direction. The foot support polygon changes during the course of the motion. We thus do not need to project momentum while rotating. Hence Δr_a would just be the difference between the original polar radius of two momenta.

We denote the rotation about the vertical axis Y by matrix R , then

$$\mathbf{q} = R * \mathbf{q} \quad (10)$$

with corresponding rotation of the relative root joint displacements as well.

For single DoF joints such as the elbow, we keep the original joint rotations unchanged. Because the above operation will introduce additional degrees of freedom and render the transformed motion unrealistic when the rotation angle in R is large. For knee joints, such cancelling will move the foot position; we thus displace the hip joints accordingly to counteract such effects.

6.3. Blending

Selected and transformed motions from the database have to be blended with the current pose or motion in order to ensure a smooth transition. We use simple linear blending techniques together with simple root displacement techniques. Better blending, foot skating elimination, and IK algorithms ([RCB98, KGS02]) could also be applied as necessary.

Certain constraints, such as foot-ground contact constraints, also have influence on the motion selection phase (Section 5). For example, a stepping pose Ω with the left foot in air cannot be easily blended to a stepping pose Ω' with the right foot in air. To disallow such blending, we set $dp(\Omega, \Omega') = \infty$ in Equation 6.

7. Results and Conclusions

We tested our algorithm using a database consisting of 66 captured balance behaviors. Users applied perturbations interactively by selecting the desired direction and magnitude using a polar coordinate plot GUI similar to Figure 3. The resulting selected and adapted balance motions along with the perturbations were then computed and animated in 3D in real-time.

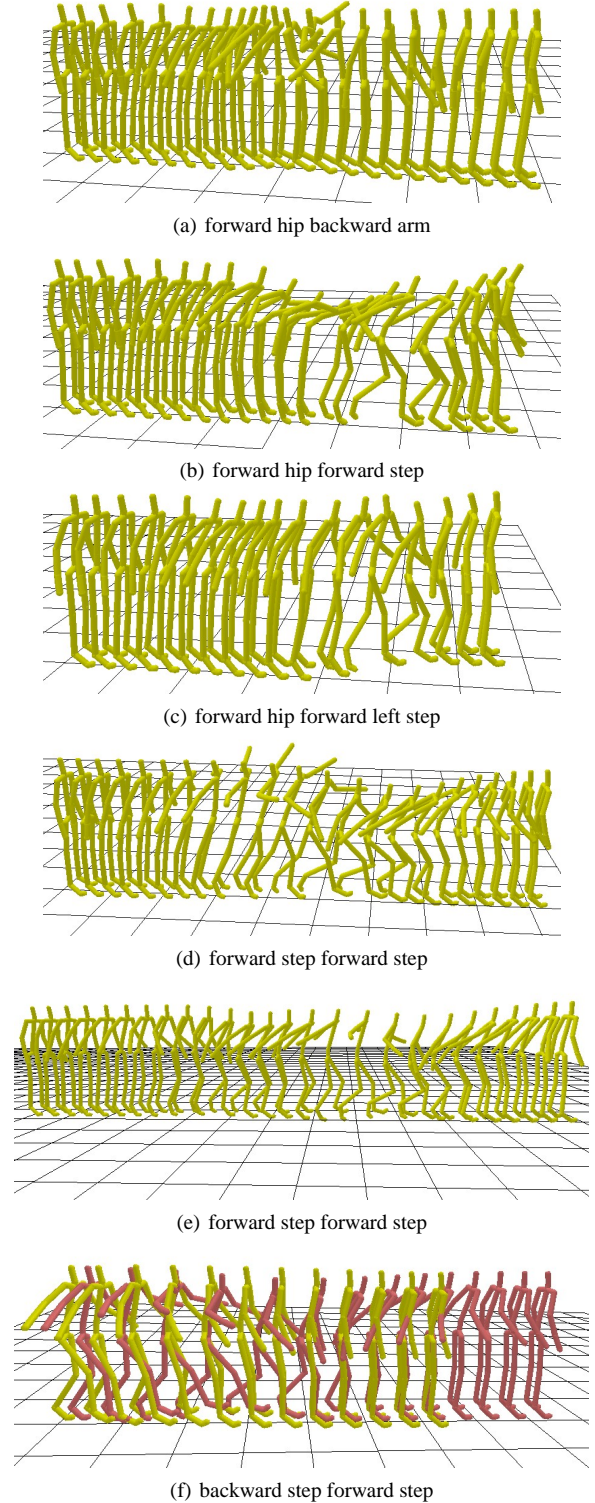


Figure 7: Balance behaviors under perturbations of different directions and magnitudes. Each motion is labelled with the directions of the perturbations and the strategies of the matching database motions.

Figure 7 shows a number of motions interactively generated from user inputs. We label each motion with the directions of the perturbations and the strategies of the matching database motions. For example, Figure 7(a) is a motion generated by a forward push then followed by a second backward push. The first matching motion is a balance response using hip strategy, the second matching motion is a balance response using arm strategy. Figure 7(f) is a backward stepping followed by a forward stepping. The forward stepping is shown in a different color and interlaced with the first backward stepping. Further examples are available in the accompanying video.

Although our motion database only contains balance behaviors under single pushes, our system successfully generated motions that respond to multiple pushes, as well as to single pushes. This is important, because it is prohibitive to properly sample the space of all possible multiple-push scenarios.

Our system is fast and effective, even when using a relatively small motion database. It should be useful for interactive video games, fast balance behavior choreography, autonomous avatar control in virtual reality applications, and reference trajectory formation for humanoid robots.

In the future, we plan to investigate the other dimensions of the perturbation momentum we neglected in this work. For example, we wish to consider perturbations from different heights, and with an angular component around the vertical axis. We are also interested in studying responses to pushes during walking and running. The current blending of a response under a second push into the first response can introduce physical implausibility, and this needs to be resolved for real robotics applications. More formal user evaluation of the results are very useful too. We also plan to explore more interactive motor task synthesis using this kind of data-driven approach, with dynamic indices and constraint-respecting transformations.

Acknowledgements: This work was supported in part by the IRIS NCE and by NSF grants IIS-0308157 and EIA-0215887.

References

- [AF03] ARIKAN O., FORSYTH D. A.: Interactive motion generation from examples. *ACM Transactions on Graphics* 21, 3 (2003), 483–490.
- [AFO03] ARIKAN O., FORSYTH D. A., O'BRIEN J. F.: Motion synthesis from annotations. *ACM Transactions on Graphics* 22, 3 (2003), 402–408.
- [AFO05] ARIKAN O., FORSYTH D. A., O'BRIEN J. F.: Pushing people around. In *Symposium on Computer Animation* (2005).
- [AG05] ABDALLAH M., GOSWAMI A.: A biomechanically motivated two-phase strategy for biped upright balance control. In *International Conference on Robotics and Automation* (2005), pp. 2008–2013.
- [ALP04] ABE Y., LIU C. K., POPOVIC Z.: Momentum-based parameterization of dynamic character motion. In *Symposium on Computer Animation 2004* (2004), pp. 173–182.
- [Coh92] COHEN M. F.: Interactive spacetime control for animation. In *Proceedings of SIGGRAPH 1992* (1992), pp. 293–302.
- [FP03] FANG A. C., POLLARD N. S.: Efficient synthesis of physically valid human motion. *ACM Transactions on Graphics* 22, 3 (2003), 417–426.
- [FvdPT01] FALOUTSOS P., VAN DE PANNE M., TERZOPOULOS D.: Composable controllers for physics-based character animation. In *Proceedings of SIGGRAPH 2001* (2001), pp. 251–260.
- [GK04] GOSWAMI A., KALLEM V.: Rate of change of angular momentum and balance maintenance of biped robots. In *IEEE Intern. Conf. on Robotics and Automation* (2004), IEEE, pp. 3785–3790.
- [Gos99] GOSWAMI A.: Postural stability of biped robots and the foot rotation indicator (fri) point. *International Journal of Robotics Research* 18, 6 (1999), 523–533.
- [Gra98] GRASSIA F. S.: Practical parameterization of rotations using the exponential map. *The Journal of Graphics Tools* 3.3 (1998).
- [HHSC97] HORAK F. B., HENRY S. M., SHUMWAY-COOK A.: Postural perturbations: new insights for treatment of balance disorders. *Physical Therapy* 77, 5 (1997), 517–534.
- [HR98] HSIAO E. T., ROBINOVITCH S. N.: Common protective movements govern unexpected falls from standing height. *Journal of Biomechanics* 31, 1 (1998), 1–9.
- [HR99] HSIAO E. T., ROBINOVITCH S. N.: Biomechanical influences on balance recovery by stepping. *Journal of Biomechanics* 32, 10 (1999), 1099–1106.
- [HvDdB*97] HOOGVLIET P., VAN DUYL W. A., DE BAKKER J. V., MULDER P. G. H., STAM H. J.: A model for the relation between the displacement of the ankle and the center of pressure in the frontal plane, during one-leg stance. *Gait Posture* 6, 1 (1997), 39–49.
- [HWBO95] HODGINS J. K., WOOTEN W. L., BROGAN D. C., O'BRIEN J. F.: Animating human athletics. In *Proceedings of SIGGRAPH 1995* (1995), pp. 71–78.
- [IAK03] ITO S., ASANO H., KAWASAKI H.: A balance control in biped double support phase based on center of pressure of ground reaction forces. In *7th IFAC Symposium on Robot Control* (2003), pp. 205–210.

- [KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion graphs. In *SIGGRAPH 2002 Conference Proceedings* (2002), pp. 473–482.
- [KGS02] KOVAR L., GLEICHER M., SCHREINER J.: Footskate cleanup for motion capture editing. In *ACM SIGGRAPH Symposium on Computer Animation* (2002), pp. 97–104.
- [KKI02] KUDOH S., KOMURA T., IKEUCHI K.: The dynamic postural adjustment with the quadratic programming method. In *International Conference on Intelligent Robots and Systems* (2002).
- [KKK*03a] KAJITA S., KANEHIRO F., KANEKO K., FUJIWARA K., HARADA K., YOKOI K., HIRUKAWA H.: Biped walking pattern generation by using preview control of zero-moment point. In *International Conference on Robotics & Automation* (2003).
- [KKK*03b] 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 *IEEE Intern. Conf. on Intelligent Robots and Systems* (2003), IEEE, pp. 1644–1650.
- [KKT*00] KAGAMI S., KANEHIRO F., TAMIYA Y., INABA M., INOUE H.: Autobalancer: An online dynamic balance compensation scheme for humanoid robots. In *Fourth Int. Workshop on Algorithmic Foundations on Robotics* (2000).
- [KLKK05] KOMURA T., LEUNG H., KUDOH S., KUFFNER J.: A feedback controller for biped humanoids that can counteract large perturbations during gait. In *International Conference on Robotics and Automation* (2005), pp. 2001–2007.
- [LCR*02] LEE J., CHAI J., REITSMA P. S. A., HODGINS J. K., POLLARD N. S.: Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics* 21, 3 (2002), 491–500.
- [LP02] LIU C. K., POPOVIC Z.: Synthesis of complex dynamic character motion from simple animations. In *Proceedings of SIGGRAPH 2002* (2002), pp. 408–416.
- [LWS02] LI Y., WANG T., SHUM H. Y.: Motion texture: A two-level statistical model for character motion synthesis. In *SIGGRAPH 2002 Conference Proceedings* (2002), pp. 465–471.
- [MMF03] MAKI B. E., MCILROY W. E., FERNIE G. R.: Change-in-support reactions for balance recovery. *IEEE Engineering in Medicine and Biology Magazine* 22, 2 (2003), 20–26.
- [Pat03] PATLA A. E.: Strategies for dynamic stability during adaptive human locomotion. *IEEE Engineering in Medicine and Biology Magazine* 22, 2 (2003), 48–52.
- [PBM00] POLLARD N. S., BEHMARAM-MOSAVAT F.: Force-based motion editing for locomotion tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation* (2000).
- [PHH04] POPOVIC M., HOFMANN A., HERR H.: Angular momentum regulation during human walking: Biomechanics and control. In *IEEE Intern. Conf. on Robotics and Automation* (2004), IEEE, pp. 2405–2411.
- [PSS02] PARK S. I., SHIN H. J., SHIN S. Y.: On-line locomotion generation based on motion blending. In *ACM SIGGRAPH Symposium on Computer Animation* (July 2002), pp. 105–112.
- [RCB98] ROSE C., COHEN M. F., BODENHEIMER B.: Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications* 18, 5 (1998), 32–41.
- [SCW01] SHUMWAY-COOK A., WOOLLACOTT M. H.: *Motor Control: theory and practical applications*, second ed. Lippincott Williams & Wilkins, 2001.
- [SHP04] SAFONOVA A., HODGINS J. K., POLLARD N. S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph.* 23, 3 (2004), 514–521.
- [SKG03] SHIN H., KOVAR L., GLEICHER M.: Physical touch-up of human motions. In *The Eleventh Pacific Conference on Computer Graphics and Applications* (2003), pp. 194–203.
- [SPH03] SAFONOVA A., POLLARD N., HODGINS J. K.: Optimizing human motion for the control of a humanoid robot. In *2nd International Symposium on Adaptive Motion of Animals and Machines (AMAM2003)* (March 2003).
- [TSK00] TAK S., SONG O.-Y., KO H.-S.: Motion balance filtering. In *Computer Graphics Forum (Eurographics 2000)* (2000), vol. 19(3), pp. 437–446.
- [VJ69] VUKOBRATOVIC M., JURICIC D.: Contribution to the synthesis of biped gait. *IEEE Transactions on Biomedical Engineering* 16 (1969), 1–6.
- [WK88] WITKIN A., KASS M.: Spacetime constraints. In *Proceedings of SIGGRAPH 1988* (1988), pp. 159–168.
- [ZH02] ZORDAN V. B., HODGINS J. K.: Motion capture-driven simulations that hit and react. In *ACM SIGGRAPH Symposium on Computer Animation* (July 2002), pp. 89–96.