

# Interactive Editing of Motion Style Using Drives and Correlations

Michael Neff<sup>†</sup> and Yejin Kim<sup>‡</sup>

Department of Computer Science and Program for Technocultural Studies, University of California, Davis

---

## Abstract

*Animation data, from motion capture or other sources, is becoming increasingly available and provides high quality motion, but is difficult to customize for the needs of a particular application. This is especially true when stylistic changes are needed, for example, to reflect a character's changing mood, differentiate one character from another or meet the precise desires of an animator. We introduce a system for editing animation data that is particularly well suited to making stylistic changes. Our approach transforms the joint angle representation of animation data into a set of pose parameters more suitable for editing. These motion drives include position data for the wrists, ankles and center of mass, as well as the rotation of the pelvis. We also extract correlations between drives and body movement, specifically between wrist position and the torso angles. The system solves for the pose at each frame based on the current values of these drives and correlations using an efficient set of inverse kinematics and balance algorithms. An animator can interactively edit the motion by performing linear operations on the motion drives or extracted correlations, or by layering additional correlations. We demonstrate the effectiveness of the approach with various examples of gesture and locomotion.*

---

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

## 1. Introduction

One of the most important yet challenging animation tasks is to make stylistic variations to motion in order to suggest a particular character, mood or role. This is particularly difficult with motion capture data, where the style is “baked in” to the data. There are two main applications where editing the style of motion capture, or other existing animation, is required. The first is offline editing, where a piece of animation does not meet the exact needs for a scene and must be adjusted. The second is online editing, where an automated system in a game or virtual world must change a character's movement to reflect his changing circumstances or internal state (e.g. the character becomes angry). We introduce

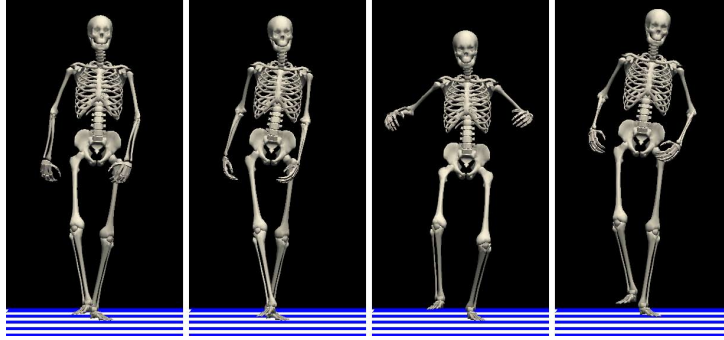
a system for editing skeletal animation data that is particularly well suited for adjusting the style of motion. Our light weight algorithms allow an animator to interactively edit the motion style to adjust the clip, making the system well suited to the first application. By providing instant feedback, the interactive editing approach allows an animator to leverage off his/her subjective understanding of motion. Furthermore, because the algorithms are light weight and propose an effective parameterization of motion, they are also appropriate for applications that need to adjust a character's motion on the fly, such as games or virtual worlds.

Our system operates by transforming the input joint angle motion representation to a set of key parameters more related to the expressive impact of the motion, such as the center of mass, ankle and wrist positions, and pelvic orientation (see Table 1). These data, both spatial and angular, effectively describe the main form of the character (shape of its body) at any point in time. Their variation over time thereby drives the motion and hence we refer to these parameters as motion *drives*. Editing them will change the motion's form. They in turn can also be used to drive *correlations* that will

---

<sup>†</sup> neff@cs.ucdavis.edu

<sup>‡</sup> rokkim@ucdavis.edu



**Figure 1:** An original walking motion (left) is edited to produce a runway walk, a sneaky walk and a limp.

control the shaping of other parts of the body, such as the torso. These correlations can be extracted from sample motion or they can be hand designed. We currently focus on linear correlations between wrist position and torso angles. The main editing operations in our system are linear adjustments to drives, the application and adjustment of correlations and blending between motion samples. These simple, easy to understand operations prove very effective in practice and should be intuitive for an animator.

The algorithmic heart of the system is a set of fast inverse kinematics and balance control routines that are used to accurately reconstruct the input motion at every frame based on the current motion parameters. Any parameter in the pose solver can be adjusted interactively.

The system has been applied to walking and gesturing data to achieve a variety of effects. For example, new styles of motion are generated that are not in our data set, such as a female runway walk and a sneaky walk, both edited from an input clip of a neutral male walk. We also show how the given style in a motion can be emphasized or diminished, for instance to reflect changing character mood. Finally, we show examples of transferring style elements between clips and blending motions, including supporting motion morphs over multiple steps. Overall, the system explores the use of body shape change as a way to transform style.

Our work makes two main contributions: First, we reformulate the problem of editing the style of a character's motion from needing to change the joint angles that define animation data to needing to adjust a set of parameters that relate more directly to the expressive nature of a character's movement: motion drives, that control the external form of the motion (see Section 5.1 for a discussion of their expressive role), and correlations, which control the pattern of shaping within the body. This reformulation allows simple linear edits to be used to effectively adjust the motion style. These are intuitive to work with and easy for an animator to understand. We provide examples of how these parameters relate to key style changes. The second contribution is a set of algorithms for reconstructing the joint angle pose data

from this new set of parameters. These algorithms are efficient, allowing the motion to be edited interactively, and extremely accurate, allowing the input motion to be precisely reconstructed.

## 2. Background

A number of researchers have developed systems that allow the style from one motion to be transferred to a second motion [ABC96, UAT95, BH00, HPP05, SCF06]. This interesting technique, known as *style transfer*, allows sample data to be used to define motion style. A shortcoming of these approaches is that it is difficult to control or adjust the extracted style. Interaction is generally high level, for instance allowing different sampled styles to be blended, but there is no way to vary specific attributes of the style. This work aims to provide animators with that much needed control.

The system developed by Liu et al. [LHP05] extracts plausible physical parameters for a human model from sample motion done in a particular style. In an offline process, these parameters can be used to synthesize new motions with the same style that are constrained to be physically correct. We approximate physical correctness by adjusting balance and maintaining foot constraints, but allow an animator to interactively change the style of a motion clip.

Another data-driven approach to style is to directly blend motions of varying styles [RCB98]. Many other techniques have focused on the direct reuse of motion capture data, such as the work on motion graphs [KGP02, AF02, LCR\*02], but these generally do not support style editing.

Other work on motion style has followed a procedural approach, using knowledge about movement garnered from the arts literature [CCZB00, NF05] or personal experience [Per95] to develop algorithms for controlling character motion. These algorithms offer good control over motion, but the final quality of the animation may not be as good as data-based approaches without extensive animator time, as the entire motion must be constructed by hand. We take

inspiration from these approaches, trying to achieve similar control but using motion data as input.

Several previous systems have looked at directly editing motion capture data. Bruderlin and Williams divide input motion into frequency bands that can be individually edited [BW95]. Witkin and Popovic [WP95] allow any pose in a motion sequence to be varied and use low frequency offset curves to blend this edit into the motion sequence. Kovar and Gleicher [KG04] extract and blend logically similar motions to provide parameterized motion control. Other motion editing work has looked at related problems like retargeting [Gle98].

Our system employs inverse kinematics (IK) routines to reconstruct the motion data. IK has been a significant subject of research in animation (e.g. [GM85, YN03]). Style is explicitly addressed in [NF04] for standing characters using *shape sets* inspired by the arts literature. Kulpa et al. [KMA05] present a system that also uses positional data instead of joint angles for reconstructing motion, but their intent is different, adapting motion to different character morphology rather than editing motion style. They use a different formulation to solve for a pose based on Cyclic Coordinate Descent, rather than the combination of analytic IK and correlations used in our approach.

Correlations have previously been used for editing and creating motion. Pullen and Bregler [PB02] extract correlations from sample motion and use them to either complete the motion of unspecified degrees of freedom (DOFs) or add texture to roughly animated DOFs. They rely on a frequency band decomposition of correlations and use correlations between angular DOFs, whereas we use correlations between translational motion *drives* and angular pose parameters. Neff et al. [NAS07] use linear correlations between mouse input and pose parameters to create a performance animation system, but make no use of motion data in their approach. Commercial packages such as Maya allow a user to set one parameter to be calculated based on the value of another.

### 3. Overview

Figure 2 provides an overview of our system. The process begins with an animator selecting a motion clip to edit. The clip should consist of skeleton joint angle data and could be from motion capture, a keyframe system, or any other process that can generate a bvh file. Our skeleton is defined in a Y-up coordinate frame. It includes 12 DOFs in the spine, two for each collar bone, three for thighs and shoulders, two for wrists, ankles and elbows, and one for knees.

In order to facilitate editing, the system transforms the motion representation into a set of adjustable pose parameters that provide meaningful control over the motion. The joint angle data is regenerated at each frame using a set of IK tools and the current parameter values. Parameters to

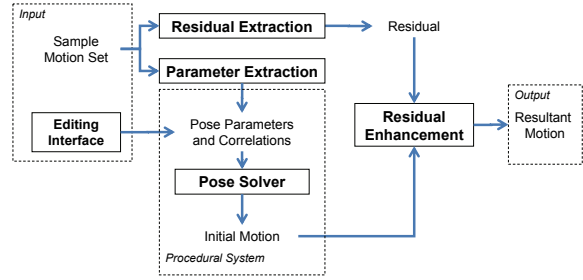


Figure 2: System overview.

regenerate the original motion are calculated during a pre-processing step, *Parameter Extraction*. These include parameters that define the basic form of the movement (ankle position, wrist position, COM) as well as additional data that supports the reconstruction of movement details, as discussed in Section 5.1. Correlations are also calculated between drives in the motion, such as the position of a character's wrist, and changes in the DOFs of the character's torso, which includes the spine and collar bones.

The pose parameters and correlations are used to reconstruct the motion. This faithfully reproduces the original motion with the exception that there may be some error between the original and torso DOFs which are reconstructed using linear correlations. During the pre-processing of the data, this error is calculated as a residual  $\mathbf{R}$  at each frame (*Residual Extraction*):

$$\mathbf{R}_i = \mathbf{T}_{O,i} - \mathbf{T}_{C,i} \quad (1)$$

where  $\mathbf{T}_{O,i}$  is a vector of the original torso DOF values for frame  $i$ ,  $\mathbf{T}_{C,i}$  is the calculated torso parameters at frame  $i$  and  $\mathbf{R}_i$  is the residual at frame  $i$ . Once the pre-processing has been completed, the original motion data is discarded.

The output motion is calculated at each frame by the *Pose Solver* using the current set of pose parameters and correlations (Section 5). The residual is added to the torso DOFs of this result (*Residual Enhancement*). The pose solver very accurately reproduces the original motion. For the gorilla walk shown in the accompanying video, the highest Root Mean Squared Error for any DOF in the reconstruction is 0.0015 radians and the average RMSE over the state vector is 0.00018. For the basic neutral walk, these numbers are 0.0051 and 0.000037 respectively. The pose solver is also fast, taking an average of 3.6 ms to run on a 3.16 GHz Intel Core Duo. This time is for unoptimized debug code, so it should be possible to significantly improve this if needed. The overall system runs in real time and reconstructs 100 samples per second during editing. Our system is built on top of the DANCE framework [SFNTH05].

The animator edits the motion by adjusting any of the pose or correlation parameters, layering on additional correlations, or adjusting a small set of other parameters (Sec-

tion 6). These edits are designed to adjust the relationship between body parts over a duration of motion, rather than to act at a particular frame. This makes them well suited for adjusting the overall style of the animation.

The system provides a continuous motion reconstruction provided the input motion parameters are themselves continuous. The pose solver operates by analytically solving a series of geometric problems that all have unique solutions, so the system cannot jump between local minima. As the parameters are extracted from continuous motion capture data, they will initially be continuous and the linear scale and shift edits will not break this continuity. If the edits force an arm end effector out of range, i.e. the arm can no longer reach the desired constraint, the system points the straightened arm in the direction of the constraint, providing graceful degradation of the solution. There are two cases where a discontinuity is possible: First, if the user types in a discontinuous change to a pose parameter, the system will jump to the new solution. Second, some edits well beyond the human range of motion can lead to flips in the trigonometry functions used to solve the geometric problems. Such unrealistic edits could be disallowed without loss of expressivity.

#### 4. Correlations

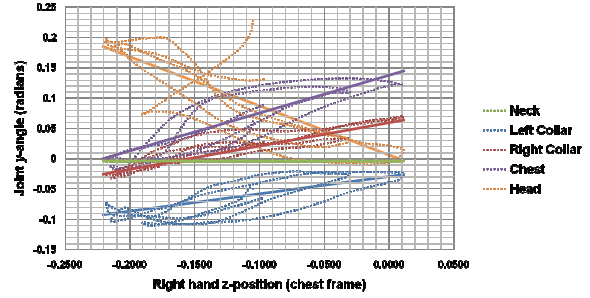
*Correlations* provide a method for specifying relationships between different parts of the body during motion. A correlation between a drive, such as wrist position, and a character DOF, such as the forward rotation of the chest, forms a relationship between these two parameters. In our system, we generally define correlations between a drive and a *set* of DOFs or pose parameters. This describes not only a relationship between the drive and driven DOF, it also forms a relationship between the members in the set of driven DOFs, describing how they vary together to create body shapes, such as a forward curl or the S curve of the beauty line in Figure 8.

These sets of correlations can be viewed as a way to drive body shape over time. Animation is a time-based art form and correlations are a particularly useful editing tool because they describe this shaping over time, rather than on a single frame. In a sense, they act like a dynamic texture that varies with the movement of the character.

We use linear correlations between spatial position data (a motion *drive*) and (generally) angular joint data. Currently we use wrist positions to drive our correlations as this is an intuitive parameter and generally correlates well with body motion. Any motion drive, however, could be used to control a correlation. A chief advantage of linear correlations is that they are very easy for an animator to understand and give intuitive results when adjusted. They also, of course, capture the low frequency relationships in the motion.

Most correlations are specified between a single drive and a set of DOFs in the skeleton, formally defined as:

$$\mathbf{P} = D\mathbf{m} + \mathbf{b} \quad (2)$$



**Figure 3:** Extracted correlations between the right wrist and the transverse DOFs of the torso.

where  $D$  is the drive parameter,  $\mathbf{P}$  is a vector of pose parameters, and  $\mathbf{m}$  and  $\mathbf{b}$  are vectors of *scale* and *shift* values respectively. The *scale* values determines how much the driven DOFs change based on the drive's movements. In effect, they control the range of the controlled DOF during the motion. The *shift* parameter allows this range window to be repositioned. For instance, if the correlation is a torso *crunch* that causes the character to curl his spine in the sagittal plane as he moves his hand, the scale parameter determines how much variation there is in the curl and the shift adjusts where the variation occurs, for instance back and forth around a vertical orientation or from hunched over to severely hunched over.

##### 4.1. Extracted Correlations

When a clip is loaded, the system automatically extracts three correlation sets, corresponding to the three planes of torso movement: coronal, sagittal and transverse. These planes are chosen because they provide easy to understand control for the animator. For each plane, the system computes the wrist DOF that best correlates with the DOFs that generate movement in that plane. For example, for the transverse plane, the selected drive might be the  $z$  position of the right wrist and the driven DOFs in the torso would be Abdomen\_Y, Chest\_Y, Neck\_Y, Head\_Y, LeftCollar\_Y and RightCollar\_Y – the DOFs that rotate the torso parallel to the transverse plane. The exact skeleton or joint angle representation is not important as long as these warpings can be defined. Figure 3 shows an example set of correlations extracted for a walking motion for the transverse plane with the  $z$  position of the right wrist as a drive. It should be noted that wrist positions are expressed in a modified form of the chest frame as discussed in Section 5.1.

The correlations are extracted by performing linear least squares [PTVF92] on the data for the drive and each torso DOF. In order to select the best drive for each plane, we compute the rank correlation [PTVF92] for each of the six potential wrist position DOFs and the driven angular DOFs in that plane. The wrist position DOF that produces the strongest

correlation is chosen as the drive. The user is free to override this selection and choose any drive for the plane.

There is no restriction against having multiple drives control a single torso DOF. The value of the DOF will simply be the sum of the input from these correlations. It is common to layer correlations when editing a motion and different drives can be used.

#### 4.2. Hand Defined Correlations

Users can easily define correlations by specifying a linear relationship between a drive DOF and a driven pose DOF. Our prototype provides an interface for doing this visually, or they can simply be written down. Importantly, the correlations do not need to be defined with much precision as they can be easily tuned when in use by adjusting the scale and shift values while editing motion.

### 5. Motion Reconstruction

Motion reconstruction is based on a combination of local IK routines, balance adjustment and the application of correlations. This approach to solving for a pose takes inspiration from the geometric view of IK adopted in [NF04], but varies in important ways, including additional lower body parameters, different prioritization of the parameters, a different approach to balance correction, and no use of optimization.

#### 5.1. Reconstruction Parameters

Designing a good parameterization is critical for supporting effective motion control. Our parameters can be divided into several groups. The primary motion drives are shown in Table 1. These define the basic form of the character's pose and are all expressively meaningful. The COM projection locates the body in space, and its behavior over time captures the overall sway of the body. The positions of the feet locate the legs, and define the character's stance (wide or narrow). Taken together, the foot positions and the COM capture a character's balance point (e.g. if the character is squarely balanced with the COM in the middle of the two feet, or near the edge of balance) and their combined movement defines a character's balance tendencies over time (e.g. if the character carries his weight over his toes or stands back on his heels; if the character makes large weight shifts or stays near the center of support). The positions of the wrists relative to the chest place the arms in space, orienting them relative to the body (e.g. in front, behind, above, to the side). The extent of upper body movement is also defined by the wrist positions, i.e. whether movements are close to the body or far from it. The pelvic rotation describes the relationship between the upper and lower body (e.g. is the torso leaning forward, back or to the side). Taken over time, the changes in pelvic orientation describe a character's hip sway. The root knee angle (or equivalently, the length of the root foot-hip

Attribute	DOFs
Wrist Position (Chest Relative)	3 per wrist
Ankle Position	3 per foot
Pelvis Orientation	3
COM on Ground Plane	2
Desired Root Knee angle	1

**Table 1:** Primary motion drives.

Attribute	DOFs
Arm Swivel	1 per arm
Leg Swivel	1 per leg

**Table 2:** Secondary motion parameters.

vector) serves as a proxy for the height of the character's pelvis off the ground. Taken together, these parameters can be seen to define the outer form of the body in space. Any of these parameters can be interactively scaled or shifted by an animator to change the reconstruction.

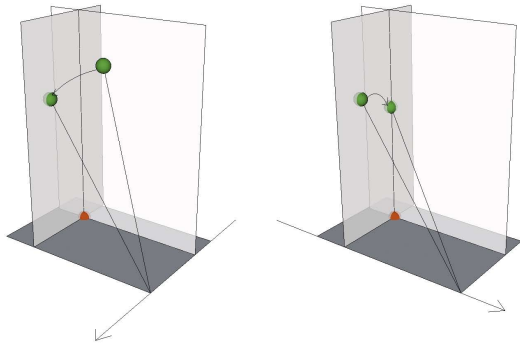
Table 2 shows the secondary pose parameters that complete the basic form of the character. These consist of the swivel angles for the arms and legs. The swivel angles are used to orient the triangle going from the shoulder to the wrist (or hip to the ankle). These are expressively important as they relate to how open or closed a character appears. They do not effect the position of the extremities in space, but rather control the placement of the knee or elbow relative to the already placed hip and ankle or shoulder and wrist. They are therefore termed *secondary*.

The wrist positions are expressed in a modified version of the chest frame that is aligned with the chest but always uses world up as its up vector. This allows the constraints to move with the character as he walks and chest deformations are added. Note that it is preferable to track the character's center of mass (COM) than the character root as this allows automatic balance adjustment to be performed. For instance, if the character is made to bend forward, the system will adjust the lower body to maintain the balance point, which would not happen if the root position was tracked instead.

The shape of the torso is captured by the correlations. Editing these will not alter the wrist positions. Other character state data, such as the angles of the wrist and finger DOFs, are simply passed through from the input data to the pose solver. Optionally, these DOFs can be directly edited, controlled with correlations, or they can be set by additional IK routines, for instance, to orient the hand.

#### 5.2. Pose Solver

The character pose is solved at each frame by updating the pose from the previous frame based on the current set of input parameters. Each time step begins by applying any user



**Figure 4:** To correct the balance, rotations are calculated for the root ankle, represented by the front corner of the plane, in order to place the actual COM (green sphere) above the projection of the desired COM on the ground plane (red sphere).

specified scale or shift edits to the input pose parameters. Next, all active correlations are applied. These can be defined to use arbitrary input pose parameters (e.g. wrist positions) to calculate arbitrary output pose parameters (e.g. torso angles) that will then be used in solving the pose at this time step. The correlations can thus adjust any pose parameters (e.g. primary or secondary drives) used in the solve.

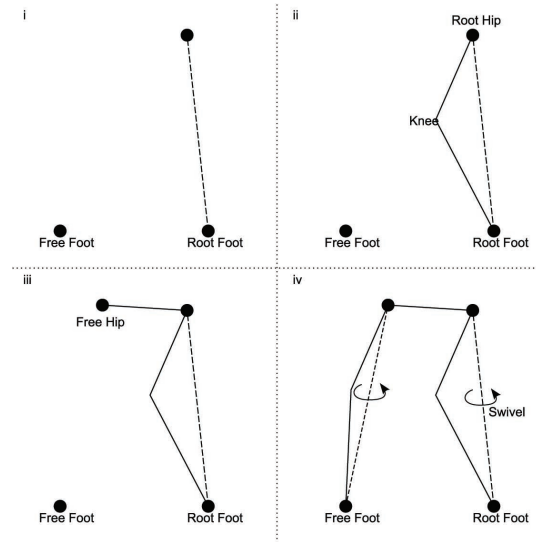
The torso DOFs are set by summing the contribution from all active correlations that modify torso angles. The residual is added to the torso DOFs to determine their final value. The torso is thus the first portion of the body that is calculated and its shape is determined using forward kinematics. The following steps solve for the remaining portions of the body:

### 5.2.1. Balance Adjustment

Balance adjustment is actuated through the ankle of the root foot (since we know the position of each foot, either can be the root). The goal of balance adjustment is to align the projection of the actual center of mass (COM) on the ground plane with the projection of the desired COM. The desired location of the COM was calculated from the corresponding frame of the input clip. The actual COM is calculated for the current state of the skeleton.

To perform the calculation, both the actual and desired COM are transformed into the local frame of the ankle. Offsets to the current  $x$  and  $z$  rotation are then computed to align the actual COM with the desired, as shown in Figure 4.

This correction is applied to the root-hip vector and the final value of the ankle angle is calculated by fitting the thigh-shin triangle to this vector. Small adjustments must still be applied to the rest of lower body loop, such as small movement of the free leg, to satisfy the constraints on pelvic rotation and free foot placement, which could lead to error. The pose updates were too small to generate noticeable error in



**Figure 5:** The lower body kinematic chain is calculated by placing each bone, starting from the root foot and tracing to the free foot.

practice, but the algorithm could be iterated if large changes were desired.

### 5.2.2. Lower Body Loop

The lower body is solved at every frame as a single kinematic chain running from the root foot to the free foot, as illustrated in Figure 5. The algorithm operates by positioning a series of line segments in space corresponding to the desired bone locations and then determining the angles necessary to achieve this configuration.

Starting at the root foot, the two angles determined in the balance routine are used to orient the root-foot to root-hip vector and hence determine the location of the root hip (i). The length of the vector is determined by the requested knee angle. The shin and thigh are then fit to this vector (ii). The pelvic rotations are used to position the pelvic vector and hence determine the location of the free foot (iii). Rotation around the pelvic vector is also applied at this time. The free foot location is an input. The shin and thigh are fit to the free foot-hip vector, determining the free knee angle. Finally, the two leg triangles are rotated around their foot-hip vectors by their respective swivel angles (iv). For our walking and gesturing motions, we define the swivel angles relative to the ground (a proxy for the motion path) as this matches the types of adjustments normally desired.

Once the bone vectors have been placed, the thigh angles required to achieve this configuration are calculated. Additional adjustments can also be made to the ankle angles in order to keep the feet parallel with the floor, etc. The skeleton is re-rooted at the foot, tracing back from the root foot

to determine the position and orientation of the skeleton's pelvic root.

The above algorithm is satisfactory for reconstructing the input motion, but can break if significant adjustments to the lower body are made. We extend the algorithm to support such edits. Problems occur because the chain is over-constrained and it is not possible to satisfy all combinations of desired foot constraints, pelvic orientation, balance point and root leg length as these are edited. Enforcing a leg length constraint can require significant pelvic rotation in order to satisfy foot placement and balance. This produces a noticeable artifact in the motion. We instead relax this constraint and prioritize maintenance of the pelvic orientation, which leads to more natural movement. This is done as follows: The loop is calculated up to the free hip location, using the desired pelvic angles. If the free hip location is farther away than the maximum length the leg allows, the length of the root foot-hip vector is shortened to eliminate this error. The following equation represents the intersection of two lines: A) a line segment with leg length and one end attached to the free foot constraint and B) a line running from the free hip location that has the direction of the root hip-foot vector. We wish to slide the free hip location down line B) to bring it within reach of the foot location, i.e. to the point where the line segment A intersects the line. Formally,  $l$  is the maximum length of the free leg,  $h$  is the desired location of the free hip,  $v$  is the vector from the root hip to the root foot and  $e$  is the desired location of the free foot, giving the following equation for the intersection of A and B.

$$l^2 = \sum_{i=x,y,z} (h_i + v_i u - e_i)^2 \quad (3)$$

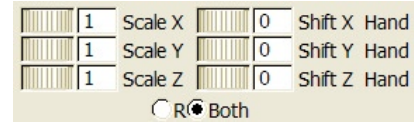
This can be solved for the parameter  $u$ , which indicates how much the root hip-foot vector needs to be shortened to move the free hip to be within range of the foot constraint. The lower loop algorithm is re-run with this new value.

### 5.2.3. Arm Positioning

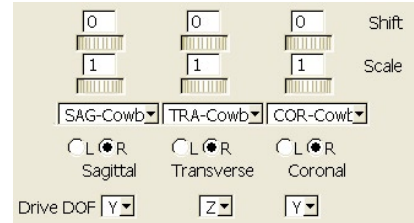
With the lower body and torso DOFs set, the position of the shoulders is defined. The upper arm-forearm triangle is positioned in order to place the wrist at the constraint point. The triangle is then rotated around the wrist-shoulder vector by the swivel angle, which is defined relative to the chest.

### 5.3. Dealing with Footskate

Undesired foot sliding during ground contact, known as *foot skate*, may occur when the motion capture data is replayed. The system allows an animator to specify periods of required foot contact. The system then calculates the average foot position and swivel angle for the planted foot and holds these fixed during these periods, eliminating foot skate.



**Figure 6:** Linear shift and scale adjustments for the wrist position drive.



**Figure 7:** Controls for the extracted correlation. All values are set automatically and the user can edit the shift and scale as desired.

## 6. Editing Animation

An animator interactively edits motion by changing sliders in an editing interface and viewing the effect on the animation (e.g. Figures 6 and 7 and the accompanying video). This instant feedback is key in allowing animators to use their subjective judgement to achieve a desired motion change. Once loaded, and after a few seconds of pre-processing to extract the motion parameters, the motion clip(s) is cycled continuously. The system uses the current settings of scale and shift values for the motion drives and correlations when calculating the parameter values used in the pose solver. These adjustments act over the entire duration of the motion being edited, rather than applying only at a particular key-time, as in most traditional frameworks. The user can slow the playback and zoom, pan and tilt his view of the 3D animation. The system supports the following controls:

**Editing Primary Drives:** The animator can scale or shift any of the primary drives in Table 1. Formally,  $p' = mp + b$  where  $p$  is the original drive value,  $p'$  the updated value used in the pose solve,  $m$  the scale and  $b$  the shift. The GUI for the hand control is shown in Figure 6. This drive editing provides good spatial control of the motion, e.g. bringing hands in or away from the body or adjusting stance width, while also giving control of the character's hip movements and balance tendencies. It should be noted that changing drive values will also change the DOFs that are correlated to them. For instance, if the hand is scaled closer to the character's body, the torso movement that is driven by the wrist will also be automatically reduced. This scales torso deformation in an intuitive way as wrist values are edited.

**Adjusting Correlations:** The three extracted correlations for the coronal, sagittal and transverse planes of the torso

are activated automatically. An example GUI for controlling these is shown in Figure 7. All the values in the GUI are initially set automatically. The user can adjust the amount of variation (*scale*) or the range over which the variation lies (*shift*) interactively by changing the dials. The user can also turn off extracted correlations, change them or layer more correlations on ten additional correlation channels. Any of the active correlations can be scaled or shifted to control its behavior. In the examples, we only use correlations to shape the character's torso, but the system allows any DOF or pose parameter to be driven by any number of correlations.

**Disabling Parameters:** Any of the pose parameters can be included or left out of the reconstruction. The residual can be scaled.

**Blending Motions:** The pose parameters provide an effective representation for motion blending. The system supports blending between locomotion clips by using Dynamic Time Warping to align the clips based on the timing of foot ground contacts (see the Appendix for details). Different blending operations are supported: The user can specify any weighted blend between clips. Correlations are calculated for each clip and then contribute to the final parameters proportional to their blend weight. The user can specify a transition between clips by specifying the starting foot step for the transition in each clip and the number of steps over which the transition should occur. Significantly, this allows the motion warp to take place over a number of steps, allowing the character to gradually adjust from one motion to the other (e.g. becoming sad over a series of steps). Previous approaches have generally blended motion over a single step, which allows less expressive control. Finally, the user can vary the blend weights of different components of the reconstruction, for instant creating a motion that uses more of the hand and correlation data from one motion and more of the foot and center of mass data from another motion. This allows different aspects of the two motions to be combined.

**Additional Controls:** The length of the root leg can be scaled, which acts as a control on the pelvis height. The time of the animation can also be scaled. In addition, offsets can be specified for any DOF or pose parameter. IK routines are also available for head tracking and orienting the hands. While the system provides a large set of controls, generally only a few parameters are needed for any particular editing task.

### 6.1. Correlation Library

Correlations are automatically extracted anytime a motion clip is loaded. These correlations are saved and can be used to edit any subsequent clip. Correlations can also be hand defined. Certain pose relationships based on earlier work drawn from the arts literature [NF04] make particularly useful correlations. They include the four coarse posture changes shown in Figure 8. *Crunch* is a curling of the



**Figure 8:** Useful hand designed posture correlations: *crunch*, *side lean*, *beauty curve* and *twist*.

spine in the sagittal plane. *Side lean* makes a large curve of the spine and collar bones in the coronal plane and may also include a COM adjustment. The *beauty curve* creates an S line in the coronal plane. The transverse plane *twist* rotates along the axis of the spine, also involving the collar bones and pelvis. In addition, a correlation that adjusts the swivel angles of the arms is very useful in practice.

## 7. Example Results

The editing process is generally fast as the user is adjusting relationships over the entire duration of the motion. Most examples discussed below were generated in a few minutes or less and often involved changing only a few parameters. Instant feedback is key in allowing an animator to understand the effect of any edits and easily achieve desired results.

To demonstrate the system's editing capacity, a number of sample animations were generated by a user who is familiar with the system, but is not a trained animator. The first set of examples show how a motion can be edited to change its style. Frames from these are shown in Figure 1. The original clip is of a male subject walking forward in his normal style. In the first example, we transform the clip into an exaggerated runway walk. This is done by first increasing the amplitude of the Z and Y pelvic rotation and also increasing the scale of the lateral weight shifts. This gives "her" more hip movement. The foot spacing is narrowed to increase this effect. Two correlations are added to complete the transformation: a *beauty curve* is driven by the Z position of the wrist and the swivel angles of the arm are rotated backwards.

In a second example, we generate a sneaky walk from the original clip. For this, the X rotation of the pelvis is rotated backwards and its scale is increased. The stance is widened slightly and the COM is shifted backwards to place the character more on its heels. The wrist drives are shifted and scaled to move the arms up and out from the character's side. The captured arm swivel is scaled down and augmented with an additional correlation. The effect is completed by adding a *side lean* correlation.

In a third example, we generate a limping walk by changing the lower body loop solver to prioritize maintaining leg length over achieving the desired pelvic rotation. This formulation tries to maintain the free knee angle extracted from



the motion capture sequence. Scaling this to eighty percent of the original value gives the character a stiff leg, producing a limp. The animation is completed by scaling the Z dimension of the COM and foot constraints to cause him to take shorter steps, raising the arms slightly, scaling down the vertical height of the feet and adding a side-lean correlation. The timing is also slowed slightly.

To show how the system can be used to create emotional changes, we edit a captured “Charlie Chaplin” style walk to create a “sad” version of it. It is also possible to edit motion to take into account physical changes in the environment. We show this by having our character walk up stairs. A linear ramp is applied to the foot height to have him walk up. Increasing the scale of the pelvic X rotation, shifting his weight slightly forward and giving him a slight lean gives the illusion of increased physical effort required to climb up stairs. In a final clip we apply correlations extracted from a more flamboyant walk to our base walk.

The next examples show how more subtle adjustments can be made to the motion style to exaggerate or diminish the motion. For example, we captured a tired walk, but our actor still appears too perky for our needs. We quickly address this by reducing the scale of the extracted transverse and sagittal correlations and pelvic Y rotation. We also add in *crunch* correlations to hunch him over, add a correlation to lower the shoulders and bring the hands a little bit forward and lower, to fit the new posture.

Other examples show how an exaggerated “gorilla-walk” clip can be scaled down to a more normal walk that still maintains gorilla like tendencies to create a motion caricature. Similar exaggeration and diminishment edits were done on captured gesturing motion.

The final set of examples relates to motion blending. By changing a blend weight, the user gets different combinations of a “bouncy” walk and a “cowboy” walk. Discrete weight changes alter the duration of the motion, causing the motion to jump to the new timing in the video, but the motion is continuous once the weight is changed. We show a combined motion which uses mostly the “cowboy” motion for the upper body movement and the “bounce” motion for the lower body movement. Finally, we show a blend between an ape-like walk and a normal walk, performed over four steps. This demonstrates how the system can smoothly blend between two motions over multiple steps.

The video shows an example of the generally small impact of the residual. In some cases, for instance a nervous head twitch that does not correlate with the rest of the body, the residual may contain more significant information.

## 8. Discussion and Conclusion

We have presented an effective system for using shape change to control the style of animation data. Our approach

transforms the motion to a parametric representation better suited for stylistic editing. By performing linear edits on these parameters, and layering correlations to drive additional shape relationships in the motion, an animator can effectively edit the style of a motion, vary the intensity of style components, blend motions and apply shaping relationships extracted from one motion to a new motion.

One limitation of the approach relates to its use of correlations. In repetitive motions like walking, the correlations are generally strong over the full duration of motion. This will not be the case for more randomly changing motions. For instance, correlations are often effective for editing a particular type of gesture, but different correlations may be desirable for different gestures. This suggests that the correlation based editing approach may be best applied in a windowing manner when editing motion that quickly changes form.

The techniques presented here are designed for editing rather than generating novel motion and are most easily applied to refining a motion. They offer no way of changing the temporal structure of motion, so the final animation is driven by the same (possibly scaled) timing as the original. Nonetheless, it is noteworthy how effective shape change alone is for altering motion style. Offering more complex control of timing is fertile ground for further research. We would also like to add an articulated foot model and associated IK routines to allow proper foot rolls as the foot makes contact with the ground, simply kept parallel to the ground. It is hoped that pose parameterization presented here will be useful in further animation research.

## Appendix: Time Warping

In order to align our locomotion clips for blending, we use the dynamic time warping algorithm described in [RCB98] which performs alignment based on the footplant timing. We manually annotate the foot contacts.

Allowing motion blends over multiple steps requires some extensions to the standard time warping algorithm if the stride length varies. The user can specify the number of transition steps and the starting step in each clip. We assume that the forward velocity within each clip is approximately constant during the blend, but the velocity may vary between clips and the stride length may also be different. A visually correct blend needs to take these into account or steps can become unnaturally short during the transition.

Our method operates in two stages. First, we precalculate the desired foot placements so that the step length blends between the stride length of each clip over the morph, starting with the stride length of the source and ending with the stride length of the destination. Formally, the stride length at step  $i$  is given by  $s_i = (1 - u_i)s_A + u_i s_B$  where  $s_A$  is the length of the stride in source clip  $A$  and  $s_B$  is the length of the stride in destination clip  $B$  and  $u_i = i / \text{numTransitionSteps} - 1$ .

At runtime, all of the drive parameters and correlations

are blended at each step using linear interpolation based on a progress variable  $p$  that runs from  $[0..1]$ . The forward motion of the center of mass and foot locations must be handled more carefully to meet the foot constraints. A separate transition weight  $p'$  is calculated at the beginning and end of each foot plant that will correctly align the forward movement of the foot with this location. The system then blends between these  $p'$  values during each stride. For the center of mass, the  $p'$  values are calculated at the start and end of the motion and interpolate over the full transition. The timing of the motions is also varied during the transition based on their relative durations. Similar timing blends are performed when the user interactively blends between two motions.

### Acknowledgements

Support for this work was provided by the Department of Computer Science and Program for Technocultural Studies, UC Davis and through a software donation from Autodesk. We thank the anonymous reviewers for their helpful comments.

### References

- [ABC96] AMAYA K., BRUDERLIN A., CALVERT T.: Emotion from motion. *Graphics Interface '96* (1996), 222–229.
- [AF02] ARIKAN O., FORSYTH D. A.: Interactive motion generation from examples. *ACM Transactions on Graphics* 21, 3 (2002), 483–490.
- [BH00] BRAND M., HERTZMANN A.: Style machines. *Proceedings of SIGGRAPH 2000* (2000), 183–192.
- [BW95] BRUDERLIN A., WILLIAMS L.: Motion signal processing. *Proceedings of SIGGRAPH 95* (1995), 97–104.
- [CCZB00] CHI D. M., COSTA M., ZHAO L., BADLER N. I.: The EMOTE model for effort and shape. In *Proc. SIGGRAPH 2000* (2000), pp. 173–182.
- [Gle98] GLEICHER M.: Retargeting motion to new characters. *Proceedings of SIGGRAPH 98* (1998), 33–42.
- [GM85] GIRARD M., MACIEJEWSKI A. A.: Computational modeling for the computer animation of legged figures. *Computer Graphics (Proceedings of SIGGRAPH 85)* 19, 3 (1985), 263–270.
- [HPP05] HSU E., PULLI K., POPOVIĆ J.: Style translation for human motion. *ACM Transactions on Graphics* 24, 3 (Aug. 2005), 1082–1089.
- [KG04] KOVAR L., GLEICHER M.: Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics* 23, 3 (Aug. 2004), 559–568.
- [KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion graphs. *ACM Transactions on Graphics* 21, 3 (2002), 473–482.
- [KMA05] KULPA R., MULTON F., ARNALDI B.: Morphology-independent representation of motions for interactive human-like animation. *Computer Graphics Forum* 24, 3 (Sept. 2005), 343–352.
- [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.
- [LHP05] LIU C. K., HERTZMANN A., POPOVIĆ Z.: Learning physics-based motion style with nonlinear inverse optimization. *ACM Transactions on Graphics* 24, 3 (Aug. 2005), 1071–1081.
- [NAS07] NEFF M., ALBRECHT I., SEIDEL H.-P.: Layered performance animation with correlation maps. *Computer Graphics Forum (Eurographics)* 26, 3 (Sept. 2007), 675–684.
- [NF04] NEFF M., FIUME E.: Methods for exploring expressive stance. In *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2004* (2004), pp. 49–58.
- [NF05] NEFF M., FIUME E.: AER: Aesthetic Exploration and Refinement for expressive character animation. In *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2005* (2005), pp. 161–170.
- [PF02] PULLEN K., BREGLER C.: Motion capture assisted animation: Texturing and synthesis. *ACM Transactions on Graphics* 21, 3 (2002), 501–508.
- [Per95] PERLIN K.: Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics* 1, 1 (1995), 5–15.
- [PTVF92] PRESS W. H., TUKOLSKY S. A., VETTERLING W. T., FLANNERY B. P.: *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1992.
- [RCB98] ROSE C., COHEN M. F., BODENHEIMER B.: Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications* 18, 5 (1998), 32–40.
- [SCF06] SHAPIRO A., CAO Y., FALOUTSOS P.: Style components. In *Graphics Interface 2006* (June 2006), pp. 33–39.
- [SFNT05] SHAPIRO A., FALOUTSOS P., NG-THOW-HING V.: Dynamic animation and control environment. *Graphics Interface '05* (2005), 61–70.
- [UAT95] UNUMA M., ANJYO K., TAKEUCHI R.: Fourier principles for emotion-based human figure animation. *Proceedings of SIGGRAPH 95* (1995), 91–96.
- [WP95] WITKIN A., POPOVIĆ Z.: Motion warping. *Proc. SIGGRAPH 1995* (1995), 105–108.
- [YN03] YAMANE K., NAKAMURA Y.: Natural motion animation through constraining and deconstraining at will. *IEEE Transactions on Visualization and Computer Graphics* 9, 3 (2003), 352–360.