Eurographics/SIGGRAPH Symposium on Computer Animation (2003)
D. Breen, M. Lin (Editors)

Aesthetic Edits For Character Animation

Michael Neff † and Eugene Fiume

Department of Computer Science University of Toronto

Abstract

The utility of an interactive tool can be measured by how pervasively it is embedded into a user's workflow. Tools for artists additionally must provide an appropriate level of control over expressive aspects of their work while suppressing unwanted intrusions due to details that are, for the moment, unnecessary. Our focus is on tools that target editing the expressive aspects of character motion. These tools allow animators to work in a way that is more expedient than modifying low-level details, and offers finer control than high level, directorial approaches. To illustrate this approach, we present three such tools, one for varying timing (succession), and two for varying motion shape (amplitude and extent). Succession editing allows the animator to vary the activation times of the joints in the motion. Amplitude editing allows the animator to vary the joint ranges covered during a motion. Extent editing allows an animator to vary how fully a character occupies space during a movement – using space freely or keeping the movement close to his body. We argue that such editing tools can be fully embedded in the workflow of character animators. We present a general animation system in which these and other edits can be defined programmatically. Working in a general pose or keyframe framework, either kinematic or dynamic motion can be generated. This system is extensible to include an arbitrary set of movement edits.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three Dimensional Graphics and RealismAnimation;

1. Introduction

Tools such as Photoshop are effective for artistic work in large part because they allow an artist to work at an appropriate level of control and because they provide rapid feedback. When working with an effective imaging tool, an artist can directly control aesthetic parameters such as color balance, tone, sharpness, and contrast, evaluate the results, and then make adjustments as required. Such interactions are usually a higher level than direct bit-map editing, which for most (but not all) tasks would be too tedious, but offer the artist a finer scale of control than high level directives such as "make the picture dark and moody". Such declarations, while evocative and helpful for setting context, can be interpreted by every artist and viewer in a different way and may still not provide the space for exercising an artist's unique style. Finding analogous levels of aesthetic control in the creation of computer animation is important to broadening its accessibility and appeal. This paper introduces a useful new class of such controls called *aesthetic edits*, which are intended to directly adjust salient aesthetic aspects of a motion.

Like our imaging metaphor, aesthetic edits operate at a higher level than keyframe editing, but lower than character directives. A choreographer could ask a dancer to perform a motion "more sadly" and hope the dancer and choreographer reconcile their views. The choreographer could instead provide further direction regarding limb flow and co-ordination of successive motions that would achieve the desired expressive intent. Aesthetic edits operate at this latter level. They are more efficient than direct keyframe editing and are easier to define, understand, and control than evocative directions such as "act more sadly". Three example edits will be presented: *succession*, which relates to joint timing, and *amplitude* and *extent*, which relate to how a character moves through space. Such edits allow the expresssion of qualities of a character's motion that set it apart from parameterized or optimized control.

† {neff|elf}@dgp.toronto.edu

We present a general framework in which animation edits can be defined. The framework generalizes the idea of keyframes or poses, which have proven to be an effective representation for both kinematic and physical animation. An extensible animation software framework is presented. Other movement edits can be added to the system by coding *movement property objects*, which are then available to the animator. Working from a pose-based representation, the system can generate either dynamic or kinematic motion.

2. Previous Work

Philips and Badler[8] presented a system that allows animators to directly adjust a character's balance, targetting an expressive aspect of motion.

Other work[13, 1] has focused on extracting the emotional content from a piece of captured motion. This extracted transform can then be applied to other motions. Bruderlin and Williams[3] adjust motion by treating movement as a signal and adjusting the gain of various frequency bands of the signal, arguing that different bands capture different aesthetic qualities of the motion. These works share our focus on editing motion, but seek to either extract an emotional state such as "happy" or "angry", or vary a frequency band of a motion, whereas our edits are aimed at unambiguous aesthetic properties of motion such as motion flow, succession and extent.

Rose et al.[10] present a system for expressive motion generation based on interpolating captured motion. A given action such as walking, called a "verb" in their work, is captured being performed in various ways. These variations define an "adverb" space. Interpolating between the captured motions gives a continuous range of expression.

Brand and Hertzmann[2] provide for very high level editing of a motion's style. They learn a style from captured motion and can then apply this style to other movement sequences. Pullen and Bregler work at a similarly high level, allowing an animator to specify key frames and then using a statistical model drawn from motion capture data to texture the key framed motion with a particular style [9].

Chi et al.[4] use Laban's Effort-Shape movement analysis to define a fixed set of parameters that can be used to modify the style of a motion . Our work shares their emphasis on expressive aspects of motion, but we aim at a more open, extensible system and we target a different set of properties.

In previous work[7], we identify another mid-level parameter – the amount of tension in a character's body – that directly effects the expressive impact of an animation. Here, we incorporate tension changes in our dynamic simulation to provide animators with an additional expressive edit.

3. Three Motion Edits

In this section, three motion edits are described which directly affect aesthetic aspects of character motion.

3.1. Succession

Poses, or keyframes, have proven to be a useful abstraction for specifying motion. The human body, however, does not move all at once. Some parts will lead, and others follow. As Walt Disney observed, "Things don't come to a stop all at once, guys; first there is one part, and then another." [12](cited on p.59). If an animation transitions from one pose to the next, bringing all parts of the body into the pose at the same time, as commonly happens in physical and kinematic control solutions, the result will have a very robotic appearance.

Successions deal with how movement spreads through the body[11]. They are very important for giving a movement a sense of flow. There are two types of successive movement: *normal* or *forward* successions and *reverse* successions. Forward successions start at the hip and move out to the limbs. Reverse successions start at the extremities and move inwards to the root.

Most motions have at least a slight forward succession. In the early days of the Disney Studio, animators spent a great deal of time studying motion. "[Their] most startling observation from films of people in motion was that almost all actions start with the hips"[12](p.72) and then the rest of the body follows through. This flow of motion is what a succession captures.

Reverse successions generally have a negative association, such as falsity, insincerity or evil, whereas forward successions are generally positive[11]. Altering the degree of the succession effects how flowing the motion will appear.

3.2. Amplitude

An amplitude edit acts in a similar manner to a scale operation in modeling. It adjusts the joint ranges over which a motion occurs. As the amplitude increases, the joint ranges spanned by a motion are increased and as amplitude decreases, the joint ranges are decreased. There are numerous ways to define an amplitude edit. Our current edit scales movement relative to an interpose average, as discussed in the implementation section below.

Bold and excited gestures often have large amplitude. Shy or nervous characters often will make small amplitude movements. The amplitude edit allows an animator to very quickly change the feel of a movement. Subtle yet expressive movements can often be obtained by taking a large motion and reducing it to a small proportion of its full amplitude while maintaining the same energy.

3.3. Extent

The concept of extent refers to the proximity of an action to a character's body [6]. It is generally applied to arm movements. There are three general extent ranges: near, mid and far. Near movements take place within a few inches of a character's body. These actions often suggest a character is timid, nervous or shy. Mid extent movements include most daily activities such as shaking hands. They occur at a medium range from a character's body and appear relaxed and normal. Far extent movements occur as far from the body as possible. Generally the arms are fully extended and stretched out from the body or above the head so that the character is occupying as much space as possible. Such actions suggest excitement and confidence. They also read more clearly if a character is viewed at a distance so are often used on stage or in long shots in film.

Extent edits and amplitude edits are particularly effective when used in conjunction with each other.

4. Implementation

We are building an extensible animation system into which new movement properties can be incorporated, much as new shaders can be incorporated into a renderer.

4.1. Underlying Representation

The fundamental representation in our animation system is based on the idea of poses, which define the configuration of the degrees of freedom of a character at specific times. This is a common representation in both kinematic keyframe systems and dynamic state-machine based control systems. We generalize the idea of a pose by allowing a given pose to define any subset of the degrees of freedom of the character. At any time, different poses may be active, controlling different subsets of the character's DOFs.

Each DOF of the character is represented by a single time-ordered track in the underlying representation used by the system. Tracks are populated with *transition elements* that define the duration of the transition to a desired pose, the desired end value for the transition (i.e. joint angle), how long the DOF is to be held in position once achieved, a curve that can be used to shape the transition, and tension values that further vary expressive shaping in a dynamic simulation. By definition, the initial value for the transition element will be the state of that DOF when the transition element becomes active.

Transition elements can be added directly to the underlying representation, but it is more usual for them to be generated by adding an *action* to the movement script. Actions are an abstraction for a unit of movement, such as a wave or a gesture. They are based on poses and defined hierarchically. Each action consists of one or more cycles, each cycle consists of one or more poses and a pose is defined by a set of transition elements. Cycles and poses are serial, so one cylce completes before the next cycle is started and similarly, one pose is completed before the next pose is begun. Cycles are useful for repetitive motions like a wave. Transition elements define a pose, so all transition elements within a given pose are executed in parallel. The action representaion is shown in Figure 1

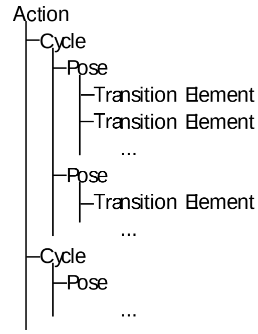

Figure 1: *A hierarchical action description.*

The action defines initial values for the properties contained in its transition elements. As a convenience, these definitions flow through the hierarchy. For instance, if a transition curve was specified at the cycle level, it would be applied to all the transition elements in all the poses contained in that cycle. It can also be freely overwritten for a particular transition element. This is facilitated with a simple labeling scheme that provides a name for each action and a unique label for each subentry based on its location in the hierarchy. For instance, the a transition element for DOF 23 in the second pose of the first cycle or an action called "wave" might have the label "wave_0_1_23". Cycle repetitions are noted by appending a repetition number to the end of the label.

Edits can be applied at arbitrary levels. An edit can be applied to the entire movement script, to an individual action or set of actions, to a specific pose or to individual transition elements. Some edits will naturally only make sense when applied at certain levels, but in general edits can be applied at any desired granularity.

A basic version of the architecture is shown in Figure 2. The script contains a set of actions that is mapped down to the tracks in the underlying representation. The aesthetic edits are then applied to the underlying representation to modify the nature of the motion. The rest of the architecture is discussed at the end of this section.

4.2. Movement Property Edits

Movement property edits are modules of code that encapsulate a particular movement idea such as succession or amplitude. They are the implementation of aesthetic edits in our system. In use, an animator selects a particular edit, de-

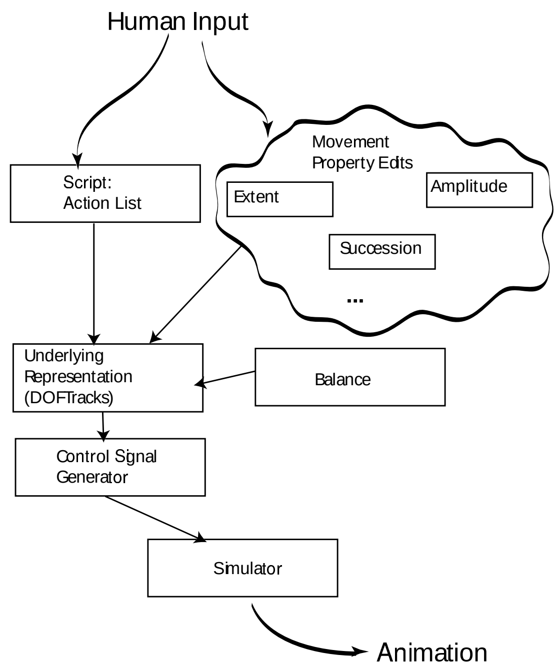

Figure 2: *A simplified version of the system architecture.*

cides what to apply it to and specifies any necessary parameters. The edit then operates by directly modifying the transition elements in the underlying motion representation. Movement properties have full access to the representation to both query and set values.

It is also possible to create reactive movement properties that can be used in dynamics simulation. These properties have full access to character state and continuously update the underlying representation to control properties such as balance. An example balance reactive controller is shown in the architecture diagram.

Animators or technical directors can freely add or modify movement properties as needed. Since these edits are procedural, they can be arbitrarily simple or complex. They can also be modified to meet an animator's exact needs.

4.3. Implementation of Edits

4.3.1. Succession

A succession takes two parameters: whether the succession is normal or reverse and how much of a time offset (t) to use between the joints involved in the motion. The edit determines all of the transition elements it is being applied to and shifts their starting time based on where they are in the character's joint hierarchy. For instance, a normal succession would not modify the first joint in the spine, it would offset the next joint by t, the following joint by $2t$ etc. The succession traces down all branches in parallel, for instance, modifying the start time of both collar bones, then both shoulders and then both elbows etc.

4.3.2. Amplitude

Amplitude edits take a positive float a which specifies the degree of the amplitude adjustment. A value of one indicates no change, less than one a reduction and greater than one an increase. This adjustment must be done with respect to a reference pose, the semantics of which we now describe.

By default, an amplitude edit will calculate the inter-pose average between end values and vary the amplitude relative to this. Two deltas are calculated, one measuring the distance from the average to the end of the pose and the second measuring the distance from the average to the end state of the previous pose. The deltas are multiplied by the amplitude a and added to the average to determine new suggested end and start values; a start value simply being the end value of the previous pose. If the pose is among a sequence of poses, there will be a suggested new value calculated relative to the average on either side of it. These are averaged to generate the final value. Joint limits can also be enforced here. Whenever quaternion joints are used, spherical linear interpolation is used instead of regular linear interpolation to determine joint angles.

An amplitude edit can also take a reference pose. In this case the amplitude is varied relative to that pose rather than relative to the computed averages.

4.3.3. Extent

Two different extent edits are provided. The first examines how closely the arms are held to the body. It blends the poses in an action with a pose that has the arms held straight down, close to the torso, in order to vary the shoulder angle and either pull a movement closer to the body or move it out into space.

The second edit examines the distance of the hand from the shoulder and allows an animator to pull the hand closer to the shoulder or move it further out into space. A similar averaging process is used here as with the amplitude edit above. When multiple poses are edited, an average extent value is calculated over all the poses and then an offset from this average is calculated for each pose. The extent edit varies the distance of this average and maintains the same offsets. In this way, the extent of an action like a wave can be varied without needing to vary each individual pose in the wave.

4.4. Movement Generation

The lower portion of the architecture in Figure 2 involves the generation of the animation. Movement generation drives off the final, edited underlying motion representation. Kinematic motion is computed using the transition curves, end values and timing information contained in the transition elements.

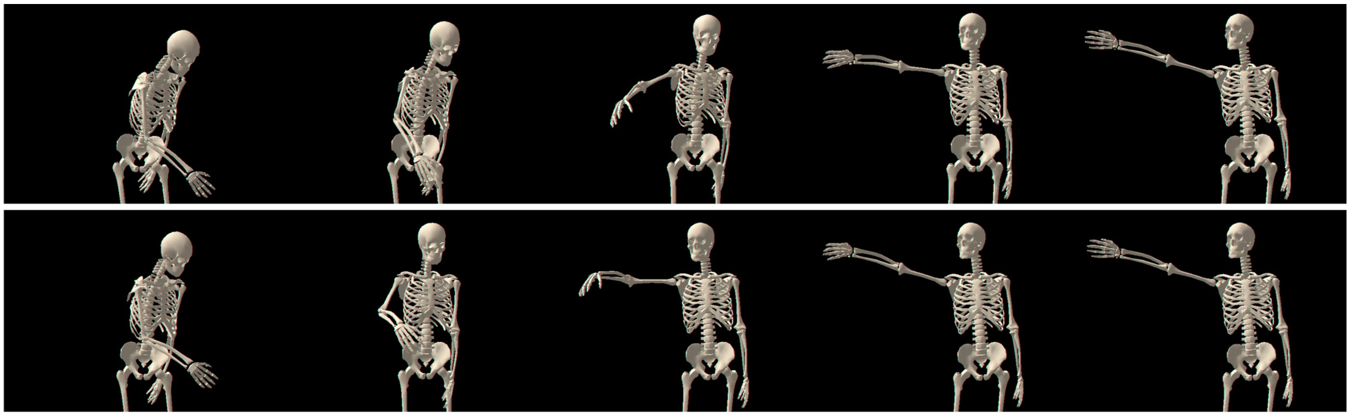

Figure 3: *Succession Edits: The top image sequence shows frames from an unedited animation. The bottom shows the same sequence after a succession edit has been applied. Note the greater sense of flow in the lower animation. Frames are evenly spaced within the transition.*

In order to determine dynamic motion, the necessary joint torques must be computed to achieve the specified motion. This is done in the control signal generator, which uses a simple antagonistic actuator that supports tension changes as described in [7]. The tension control formulation will not be repeated here, but the basic ideas is that the gains of the actuator are varied to track the transition curvers. Accurate joint positioning is achieved by determining the torques caused by gravity acting on the character and then adjusting muscle gains to compensate for these torques to achieve the desired position. For most upper body motions, the required gains for the end pose are calculated at the beginning of the motion and then the gain values are varied from the starting value to the final value. The system uses the underlying representation to estimate future states of the character in order to estimate the torques induced by gravity at the end of a pose and calculate appropriate muscle gains. The torque values are used as input to a physics simulator which generates the final motion. The simulation code is generated by a commercial package, SD/Fast[5]. For kinematic motion, the control signal generator simply passes information from the underlying representation to a kinematic "simulator" which generates the final motion.

5. Results

All animations discussed in this paper are available online at http://www.dgp.toronto.edu/˜neff.

A simple bowing animation shows the power of the succession edit. The animation consists of two poses on top of the rest pose, one of the character bowed forward and a second of the character gesturing off and up to the right. The basic animation generated from the poses has a stiff, robotic feel. The animator applies the succession edit with an offset of 0.2 sec for the first transition and 0.3 sec for the second transition. As can be seen in a side by side comparison of the animations, the application of the succession edit gives the movements a remarkable sense of flow. A few frames from the end of the animation are shown in Figure 3. The character's lower body is automatically controlled by the reactive balance controller.

A simple animation based on the sixties dance "The Twist" is generated by cycling two poses. The various edits are applied over multiple repetitions of the dance. Due to the programmatic representation used to define the edits, it is a straightforward task to vary the intensity of the edits over a movement sequence such as this, allowing the dance to be built up to a wild crescendo, or reduced to a shy bob.

When creating a realistic piece of acting, sometimes a subtle piece of motion is needed to colour a scene. What is called for is often not a broad gesture that would distract from the scene, but a small piece of motion that does not draw attention to itself, but helps to set a mood for a character. These very subtle gestures, while clear in intent in an animator's mind, are difficult to envision and animate. One effective way to generate them is to take a broad piece of motion and then apply edits to both adjust the flow through succession changes and to drastically scale down the motion using amplitude and extent. We illustrate this with a tilting dance motion that has been reduced to generate a subtle, but expressive "twitch" that can be applied to a character. A succession edit was also used.

We employ dynamic simulation in a manner analogous to a final rendering pass. The animation is first created kinematically as this offers a more efficient initial workflow. A final simulated version of the motion is then generated which incorporates additional nuances afforded by physics, such as pendular limb motion, force transference between joints, smoothing, and envelope shaping caused by tension changes.

6. Discussion, Conclusions and Future Work

We have presented a system in which aesthetic motion edits can be defined and applied. These edits target some important expressive aspects of motion. We argue that tools at this level of abstraction offer the potential for being particularly effective for character animators as such tools allow them to focus on expressive aspects of motion while at the same time providing an appropriate level of control. This approach has been demonstrated with three exemplars, succession, amplitude and extent. We also demonstrated how they allow an animator to quickly adjust the various expressive aspects of a motion.

In the language of animation, very different approaches may be taken by different animators to achieve a specific expressive effect. Our edits thus serve a pedagogic purpose, identifying for less experienced animators different ways to vary motion to achieve such an effect.

Much work remains on developing other more interesting edits, on refining existing edits, and on developing new user interface techniques for edit specification. In particular, aside from enforcing joint limits and ground contact, the system does not currently enforce constraints. It would be useful to allow an animator to specify say an end-effector contstraint that is maintained while an edit is applied. Rules for combining potentially conflicting edits would also be useful.

This work also suggests the fascinating and crucial problem of user validation. While it is difficult to develop and conduct user performance studies on complex software such as that used for authoring animation, it is important to develop methodologies that serve to validate experimentally or empirically the effects of what we claim to be improved animation workflow afforded by expressive edits. This topic will be the subject of considerable future work.

Acknowledgements

This work was financially supported by an NSERC Post Graduate Scholarship, an NSERC Research Grant and graduate student funding from the University of Toronto. The software in this paper is developed on top of the DANCE software framework, authored by Petros Faloutsos and Victor Ng-Thow-Hing. We would like to thank the anonymous reviewers for their useful comments.

References

1. AMAYA, K., BRUDERLIN, A., AND CALVERT, T. May 1996. Emotion from motion. *Graphics Interface '96*, 222–229.
2. BRAND, M., AND HERTZMANN, A. July 2000. Style machines. *Proceedings of SIGGRAPH 2000*, 183–192.
3. BRUDERLIN, A., AND WILLIAMS, L. August 1995. Motion signal processing. *Proceedings of SIGGRAPH 95*, 97–104.
4. CHI, D. M., COSTA, M., ZHAO, L., AND BADLER, N. I. July 2000. The emote model for effort and shape. *Proceedings of SIGGRAPH 2000*, 173–182.
5. Michael G. Hollars, Dan E. Rosenthal, and Michael A. Sherman. *SD/FAST User's Manual*. Symbolic Dynamics Inc., 1994.
6. LABAN, R. 1988. *The Mastery of Movement*, fourth ed. Northcote House, London. Revised by Lisa Ullman.
7. NEFF, M., AND FIUME, E. 2002. Modeling tension and relaxation for computer animation. In *ACM SIGGRAPH Symposium on Computer Animation*, 81–88.
8. PHILLIPS, C. B., AND BADLER, N. I. July 1991. Interactive behaviors for bipedal articulated figures. *Computer Graphics (Proceedings of SIGGRAPH 91) 25*, 4, 359–362.
9. PULLEN, K., AND BREGLER, C. 2002. Motion capture assisted animation: Texturing and synthesis. *ACM Transactions on Graphics 21*, 3 (July), 501–508.
10. ROSE, C., COHEN, M. F., AND BODENHEIMER, B. September - October 1998. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications 18*, 5, 32–40.
11. SHAWN, T. 1963. *Every Little Movement: A Book about Francois Delsarte*, second revised ed. Dance Horizons, Inc., New York.
12. THOMAS, F., AND JOHNSTON, O. 1981. *The Illusion of Life: Disney Animation*. Abbeville Press, New York.
13. UNUMA, M., ANJYO, K., AND TAKEUCHI, R. August 1995. Fourier principles for emotion-based human figure animation. *Proceedings of SIGGRAPH 95*, 91–96.