January 1999

A Motion Control Scheme for Animating Expressive Arm Movements

Diane M. Chi
University of Pennsylvania

Chi, Diane M., "A Motion Control Scheme for Animating Expressive Arm Movements" (1999). *IRCS Technical Reports Series*. 45.
http://repository.upenn.edu/ircs_reports/45

University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-99-06.

A Motion Control Scheme for Animating Expressive Arm Movements

Abstract
Current methods for figure animation involve a tradeoff between the level of realism captured in the movements and the ease of generating the animations. We introduce a motion control paradigm that circumvents this tradeoff-it provides the ability to generate a wide range of natural-looking movements with minimal user labor.

Effort, which is one part of Rudolf Laban's system for observing and analyzing movement, describes the qualitative aspects of movement. Our motion control paradigm simplifies the generation of expressive movements by proceduralizing these qualitative aspects to hide the non-intuitive, quantitative aspects of movement. We build a model of Effort using a set of kinematic movement parameters that defines how a figure moves between goal keypoints. Our motion control scheme provides control through Effort's four dimensional system of textual descriptors, providing a level of control thus far missing from behavioral animation systems and offering novel specification and editing capabilities on top of traditional keyframing and inverse kinematics methods. Since our Effort model is inexpensive computationally, Effort-based motion control systems can work in real-time.

We demonstrate our motion control scheme by implementing EMOTE (Expressive MOTion Engine), a character animation module for expressive arm movements. EMOTE works with inverse kinematics to control the qualitative aspects of end-effector specified movements. The user specifies general movements by entering a sequence of goal positions for each hand. The user then expresses the essence of the movement by adjusting sliders for the Effort motion factors: Space, Weight, Time, and Flow. EMOTE produces a wide range of expressive movements, provides an easy-to-use interface (that is more intuitive than joint angle interpolation curves or physical parameters), features interactive editing, and real-time motion generation.

Comments
University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-99-06.

A MOTION CONTROL SCHEME FOR ANIMATING EXPRESSIVE ARM MOVEMENTS

DIANE M. CHI

A DISSERTATION
in

COMPUTER AND INFORMATION SCIENCE

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy.

1999

Norman I. Badler
Superviser

Jean Gallier
Graduate Group Chair

ABSTRACT

A MOTION CONTROL SCHEME FOR ANIMATING EXPRESSIVE ARM MOVEMENTS

Diane M. Chi

Supervisor: Norman I. Badler

Current methods for figure animation involve a tradeoff between the level of realism captured in the movements and the ease of generating the animations. We introduce a motion control paradigm that circumvents this tradeoff—it provides the ability to generate a wide range of natural-looking movements with minimal user labor.

Effort, which is one part of Rudolf Laban's system for observing and analyzing movement, describes the qualitative aspects of movement. Our motion control paradigm simplifies the generation of expressive movements by proceduralizing these qualitative aspects to hide the non-intuitive, quantitative aspects of movement. We build a model of Effort using a set of kinematic movement parameters that defines how a figure moves between goal keypoints. Our motion control scheme provides control through Effort's four-dimensional system of textual descriptors, providing a level of control thus far missing from behavioral animation systems and offering novel specification and editing capabilities on top of traditional keyframing and inverse kinematics methods. Since our Effort model is inexpensive computationally, Effort-based motion control systems can work in real-time.

We demonstrate our motion control scheme by implementing EMOTE (Expressive MOTion Engine), a character animation module for expressive arm movements. EMOTE works with inverse kinematics to control the qualitative aspects of end-effector specified movements. The user specifies general movements by entering a sequence of goal positions for each hand. The user then expresses the essence of the movement by adjusting sliders for the Effort motion factors: Space, Weight, Time, and Flow. EMOTE produces a wide range of expressive movements, provides an easy-to-use interface (that is more intuitive than joint angle interpolation curves or physical parameters), features interactive editing, and real-time motion generation.

Acknowledgements

When I reflect on these past few years of graduate study, there are numerous people to whom I am grateful. Above all, I am deeply indebted to my advisor Professor Norm Badler. Without his vision and inspiration, this work would never have been imagined much less demonstrated. Norm has been an ideal advisor—sensing when I needed direction or motivation, yet trusting me with independence and responsibility. I admire his intelligence, managerial abilities, and selfless devotion to his students, and am grateful for his years of advice, guidance, and support.

I am also grateful to Janis Pforsich, who was instrumental in providing the LMA expertise for the project. Her enthusiasm and willingness to explore the potential of computer technology were true assets, and I appreciate her role as a teacher and a friend.

My committee, Dr. Armin Bruderlin, Dr. Martha Palmer, Professor CJ Taylor, and Professor Dimitri Metaxas, deserves special thanks for reading my papers and providing insightful comments. Their varied perspectives brought truly useful suggestions that strengthened the work and its presentation, and their interest in the project was encouraging.

Others that deserve thanks include: Deepak Tolani for providing the inverse kinematics code; Mike Pan for his modeling and video assistance, along with Mark Palatucci, for being guinea pigs for my user tests; Amy Matthews, Connie Cook, and Janet Hamburg for their Effort model evaluations; Karen Carter for making the lab a pleasant place to work; Rama Bindiganavale and John Granieri for their lab software support; Harold Sun and Christian Vogler for setting up the motion capture system; Professor Bonnie Webber, Dr. John Clarke, and Dr. Yumi Iwasaki for their guidance in my earlier research work; Professors

Mark Steedman and Jean Gallier for their work as graduate student advocates; and Mike Felker for keeping the department running smoothly.

I am also grateful to the National Physical Science Consortium and the Department of Defense at Fort Meade for supporting my graduate studies and encouraging women and minorities to enter underrepresented fields of study.

Thanks to my "dissertation support group" — Omolola Ijeoma Ogunyemi and Sonu Chopra for offering a listening ear, providing sound advice, and joining me on all those "stress-relieving" shopping trips and candy runs. I appreciate the many past and present members of the graphics lab who were valuable colleagues and friends, including Vangelis Kokkevis, Ken Noble, Roxana Cantarovici, Charles Erignac, Bond-Jay Ting, Jeff Nimeroff, Barry Reich, and Jonathan Kaye. I'm also grateful for my friends David Brogan, Chrissy Benson, Alan Kuo, Jerome Strach, and Margie Wendling.

I would like to thank David Jelinek for his input in various technical discussions, for laboring through my papers and providing (sometimes) helpful comments, and for being my best friend during the last few years.

I am constantly reminded of how lucky I am to have such a wonderful family. Marie and Mike have been the best siblings a little sister could ask for—from the childhood antics through the transition into adults, they have shared and shaped the important moments of my life. Their unconditional support means a lot to me. I also appreciate my brother-in-law Darrell for being my financial, hardware, and home repair advisor. My nephews Jeffrey and Kevin are constant sources of fun, humor, and entertainment, reminding me off the truly important things in life. Thanks are due to Auntie Alice, my California relatives, and the close family friends who "adopted" me while I was away from home.

I want to express my utmost respect and admiration for my father and my mother. They are my role models. My father built a successful career, starting out with little support and a lot of determination. He has also shown me that age is of no matter—one can always be young at heart. My mother is the most dedicated and generous person I know. I admire her uncomplaining sacrifices and her ability to provide practical advice even in the face of crisis. I thank them for their unflagging support and guidance, and I dedicate this work to them.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As a society, we are surrounded by other people. Consciously and subconsciously, we are constantly observing how people move. We often recognize others solely by catching a glimpse of them walking or moving. When people limp or make subtle compensations for an injury, we immediately recognize something unnatural about their movements. Through constant observation of everyday life, we become subconsciously aware of the subtleties of human movement. Animations of virtual characters must capture these subtleties in order to appear life-like and believable.

Current methods for human or character animation involve a tradeoff between the level of realism captured in the movements and the ease of generating the animations. At one end of the spectrum, computer-animated features and movie special effects display extremely realistic individuals with personalized movement characteristics; however, they require teams of animators using labor-intensive systems where every movement, from the flick of a finger to a full-body leap, must be explicitly specified. At the other end of the spectrum, behavioral animation systems can automatically generate multiple characters interacting with each other and their environments by merely specifying a set of initial conditions; however, the various characters often move in a fairly mechanical manner and share very similar motions. A method that combines the advantages found at these two extremes—the ability to generate a wide range of natural-looking movements with minimal user labor—can play an important role in circumventing this tradeoff.

1.1 Our Approach

We introduce a motion control paradigm that simplifies the generation of expressive movements by proceduralizing the qualitative aspects of movement to hide the non-intuitive quantitative aspects. To do this, we needed (1) a language for specifying expressive movements, and (2) a translation of this language into quantitative movement parameters.

We sought a language that would cover the complex description space of expressive movement while still providing an intuitive interface. We examined the nonverbal communication and dance notation literature for methods of describing, notating, and recording human movement. We found that the Effort component of Laban Movement Analysis [50, 51, 27, 7, 59] met our requirements, has a solid foundation based on observation and analysis of humans performing a wide range of movements, and is being used as a research tool in a growing number of disciplines. Effort is the part of Rudolf Laban's theories on movement that describes the qualitative aspects of movement using textual descriptors along four motion factors: Space, Weight, Time, and Flow (Chapter 3). The extremes of each motion factor give the eight Effort Elements: Indirect, Direct, Light, Strong, Sustained, Sudden, Free, and Bound.

We derived quantitative structures to model Effort. Through much trial and error, we established an empirical model of each Effort Element and designed techniques to generate ranges and combinations of Effort (Chapter 4).

We demonstrate the use of our motion control scheme by implementing EMOTE (Expressive MOTion Engine), a 3D animation control module for expressive arm movements. EMOTE lets users specify movements using end-effector positions and Effort settings. A sequence of end-effector positions provides a general spatial description of a movement, while the Effort settings define the desired qualitative nature of the movement. EMOTE uses inverse kinematics (IK) to compute postures for an articulated figure from the specification of end-effector locations [89, 76]. Inverse kinematics, however, does not specify how a figure achieves a computed posture or changes between a series of postures. Thus, IK output is usually used as input to a separate animation process, such as an end-effector linear interpolator or figure control equations. EMOTE lets a user express the essence of the movement by setting sliders for each of the four qualitative Effort motion factors: Space, Weight, Time, and Flow. EMOTE uses these Effort settings to compute the values for a set of low-level motion parameters that specify an animation that follows the defined position sequence and displays the selected Effort qualities. Since our translation process is computationally inexpensive, EMOTE provides interactive editing and real-time motion generation.

1.2 Motivation

We seek to provide a useful tool that enables further automation of expressive character animations. We believe such a system can play an important role in keyframe animation systems, virtual environments, games, and behavioral animation systems.

For novice keyframe animators, an Effort-based motion control system provides the ability to generate a broad range of motions with a short learning curve and an easy-to-use interface. For skilled animators, our system provides a blocking tool (for quickly sketching out movement) using a language that directly supports the desired intent of the animated character in a process that is repeatable and provides interactive editing at the parameter level. Traditional low-level editing methods can be provided to allow animators to further refine their animations.

Users of virtual environment systems and computer game players often interact with other individuals, both user-controlled actors (avatars) and autonomous synthetic characters. With current systems, the synthetic characters either have to be completely specified with a library of reactive movements, or the characters all have extremely similar actions and reactions. For realistic interactions, virtual characters must appear to be individuals—they must move naturally and with subtle "personality" traits. In fact, having them act independently is more important than having them look different physically, because we commonly use actions and other non-verbal communication to try to infer emotional state, attitude, and ultimately intent. Behavioral animation uses a hierarchical structure to define crowds, herds, and schools of characters [13, 65, 69, 77]. Such systems define low-level movements such as a character's basic means of locomotion, as well as higher level behaviors such as schooling or flocking, obstacle avoidance, pursuit and evasion, and aggressiveness. However, these systems frequently omit the "middle" layer – variations on low-level movements to create different expressions, personalities, and intents. Further, behavioral systems use characters represented by simple models which offer a very small range of movements and expression. Recent feature films have used crowd simulations to create armies in Disney's *Mulan*[36], worker ants in Pacific Data Images' *Antz*[67], and Tippett Studios' bugs in *Starship Troopers*[38]. Casts of characters were created by varying body shapes, props, and clothing; however, the movements of these characters were drawn from a small pre-defined library of motions generated by keyframing, motion capture, stop motion, and procedural animation.

Games, virtual environments, and computer-generated animations have obvious applications in entertainment; however, synthetic characters also enable a participant to experience a wide variety of scenarios with cognitive and decision-making challenges without the physical consequences or harm that could result in the real world. For instance, prototype systems already exist for training battlefield medics [2, 19, 20], firefighters [12], and surgeons [58, 28].

A system that allows users to customize basic movements based on a character's personality, mood, and attitudes is the first step towards simplifying the development of a repertoire of characters with a wide range of expressiveness. By selecting a general human movement description language to customize motions, such a system can also lead to the generation of virtual characters from different cultures. Although certain gestures are culture-specific, the descriptions of the movements of individuals with the same emotions and intent are often similar. A playful child moves with free, indirect abandon; an aggressor makes strong, direct, and sudden attacks; and a soldier marches in bound, sustained strides. Our motion control paradigm allows the user to customize movements through general qualitative descriptors. This is the first step towards enabling a system where a user creates character by specifying attitudes and intentions, which in turn may eventually lead to the automatic generation of appropriate movements from speech text, a storyboard script, or a behavioral simulation.

Chapter 2

Related Work

In this chapter, we examine the current approaches to motion control used in computer animation. Then, we discuss techniques that have been developed to specifically address emotion and expression in animated movement. Next, we briefly overview the biomechanics research and related work on human movement. We conclude with a survey of previous efforts to combine dance notation and computers.

2.1 Motion Control

The basic approaches to motion control for articulated figures include: keyframing, dynamic simulation, motion capture, and procedural methods. Each differs in the amount of user specification, the skill required to generate good-looking animations, the ease of editing, and the generality of its application.

Keyframing articulated figures uses kinematics, requiring the user to specify the positions and orientations of a figure and its limbs at major points in a movement. The computer automatically generates the frames in-between the specified keyframes. With keyframing, the animator has a lot of control over the final animation. To change the generated animation, the animator can add, remove, or change keyframes. The disadvantage of keyframing is that it requires a certain artistry and skill in order to capture life-like movements and personality. Current 3D keyframing systems provide tools so users can view and edit curves representing the changing parameter values over an animation.

For instance, users are often able to edit joint angle interpolation curves, a motion's velocity curve, or an object's trajectory in space. Another technique, inverse kinematics, reduces the amount of user input by allowing users to specify end-effector locations to determine the posture of a figure [89]. These tools aid the tedious specification tasks of the animator but fail to offer assistance in capturing expression. Some systems also provide shape blending tools that morph a source object into a target object over a specified amount of time. This technique is typically used for facial or soft object animation and not for movements of articulated figures.

Dynamic simulation (sometimes called physics-based modeling) uses the physics of bodies in motion to produce animations with a realistic impression of weight, friction, inertia, and other physical properties. However, dynamic simulation requires solving the equations of motion, which is computationally expensive and may result in unstable solutions. Also, users must provide a detailed physical description for all objects in the scene [84, 68]. This description must include object properties, such as mass along with its distribution over the object, moments of inertia, segment lengths, and joint limits, as well as any forces or torques acting on the body, any frictional or damping effects that might be triggered through collisions or other movements, and any effects due to energy expenditure and transfer. Since dynamic simulation generates motions entirely from the physical description, users have limited and indirect control over the final animation; and editing proves non-intuitive. Efforts to simplify control of dynamic systems while maintaining physically correct animations include blending kinematic and dynamic techniques [42, 8, 84, 14, 48], customizing controllers for specific tasks [68, 39], and automatically generating controllers using genetic algorithms or control theory [73, 72, 35, 62, 54]. While these methods capture the physical realism of bodies in motion, they do not address how to change the movements to display different expressions or intentions. Spacetime constraints methods attempt to provide the user with better control over generated animations and editing capabilities [85, 21, 55]. However, thus far, editing of *how* a motion is performed using spacetime constraint systems is limited to qualities that directly translate to physical criteria. For instance, "don't waste energy" minimizes kinetic energy, while "land hard at the end of a jump" maximizes the contact force on landing. On the other hand, achieving expressive qualities such as "careful" or "meandering" are not addressed. Also, spacetime constraints methods are non-interactive and have been limited to movements of simple creatures.

Motion capture uses electro-magnetic or optical technologies to collect position and orientation data of real human movement, which can then be used to animate articulated figures. Motion capture techniques can capture the small nuances of an individual's movement, enabling an animated character to display the intended expression of the original human performer; however, motion capture requires expensive equipment and quickly becomes impractical when an animation involves a large number of characters and/or movements. Further, motion capture offers only indirect control over produced animations—by requesting changes from the performer and re-capturing the subsequent motions. Gleicher introduced a spacetime constraints technique for adapting motions from one character to produce motion specifications for other differently sized characters [33, 34]. For instance, using motion capture data of two people swing dancing, he changed the sizes of the dancers, but was able to maintain their foot contact to the floor and their hand contact with each other using his retargetting technique. Bindiganavale and Badler present a method that *automatically* recognizes and maintains spatial, as well as visual constraints while mapping them to characters of different sizes[10]. For instance, the data for an adult grabbing a mug on a table and drinking from it is modified to animate a child drinking from the mug—they ensure that the child grasps the mug at the correct location and brings it to his lips. These methods however provide no direct means of editing the expressive qualities of captured movement.

Procedural animation methods define how a figure moves over time using a model (often mathematical) that responds to some external input, either from a human user, other procedures, or some means of sensing the current state[1]. Often procedural methods are geared towards specific applications. For instance, procedural methods have been used in animating a number of physical phenomena, such as cloth movements, waves, and particle phenomena [29]. Bruderlin and Calvert present a procedural system for animating

---

[1] We note that the broad definition given for procedural animation gives a disjoint set of techniques, some of which, by our categorization, use other fundamental approaches to animation, notably physically-based methods.

the running movements of an articulated human-like figure [15]. Their system, RUNNER, provides a user with high-level parameters and attributes to interactively alter an animation to display a wide variety of human running styles. Parameters determine the running stride and include velocity, step length, step frequency, flight height, and level of running expertise. Attributes allow the user to individualize a run by setting movement variables for the arms, torso, pelvis, and legs. RUNNER modifies a default running animation in real-time according to changes specified by its user.

Behavioral animation, a subset of procedural methods, uses rules to define virtual creatures that can sense and react to the environment. Various researchers have developed behavioral animation systems to generate animations of multiple creatures with varying personalities and/or goals [69, 77, 5, 60]. Tu and Terzopoulos create a "virtual marine world" filled with different fishes [77]. They define three types of fish – predators, prey, and pacifists–each with a different set of intentions and behaviors. Different fish types react to their environments in different ways; however, since they are all physically modeled using the same spring-mass model, their low-level movements are all the same. We note though that the basic movement of real fish is fairly unexpressive (at least to this casual observer!). Blumberg and Galyean present a method of directing autonomous creatures at multiple levels [13]. Their hierarchical organization of behaviors allow commands that reflect emotional state to initiate lower-level behaviors. In their example, a dog told to display a happy state, induces behaviors to set appropriate positions for his ears, tail, and mouth and may also issue a meta-command for the dog to use a more jovial gait. Although their system allows users to change the expression of characters, each expression and its manifestations must be defined separately. Badler and his collaborators have implemented behavioral systems with a more complex model—an articulated human figure. In the *Hide and Seek* project, characters are assigned different roles as hiders or as the seeker in on-the-fly animations of the children's game of the same name [60]. The role of the character influences its goals and subsequent behavior towards other characters in the scene, but the low-level movements (locomotion) of all characters is the same and fairly expressionless.

Procedural methods ease the work required of the animator by encoding some information on how things move. Our motion control paradigm takes this approach and applies it to a general-purpose application—the expressivity of movements. We parameterize the non-intuitive, quantitative aspects of movement and provide the user with more meaningful, interactive control through textual descriptors (Effort Elements). Since our motion control paradigm is based on a comprehensive language for describing movements, it can be applied to any type of movements and objects.

2.2 Expressive Movement

Several researchers have specifically addressed the generation of expressiveness in movement. Their approaches involve adding expressiveness to neutral motions [64, 79, 1] or providing editing tools to modify expression or fit different constraints [16, 86, 70]. Most of these techniques are not specific to any particular motion generation method and are valuable tools for making existing motions more usable; however, they may prove costly or difficult to use in generating the range of human expressivity. Other researchers have developed systems that generate animations from high-level specifications [61, 47].

Ken Perlin gives the "visual impression of personality" to animated puppets using stochastic noise functions [64]. The user controls the puppet through a panel of buttons representing a set of primitive actions. The system smoothly blends the selected actions into a coherent animation. The user can vary expression by adding a random component to joints, modifying the bias on joints, and varying the transition times for different primitive actions. These methods give characters a dynamic presence and a random sense of attentiveness, which play a role in responsive virtual agents, but do not necessarily present a natural, human-like demeanor. Also, varying expression by modifying joint angle frequency and amplitude functions is non-intuitive.

Unuma, Anjyo, and Takeuchi capture a wide variety of expression in human locomotion [80, 79]. They use motion capture to collect joint angle data of a human subject performing neutral and various emotion-influenced locomotion. From the discrete data, they approximate the original movement with a continuous *rescaled Fourier function model.* Their Fourier function model allows them to smoothly transition between two captured motions using interpolation, as well as to generate exaggerated motions using extrapolation.

For instance, they can interpolate between a “normal” walk and a “tired” walk to get various degrees of “tiredness”; more importantly, they can extrapolate to get a “brisk” and an exaggerated “tired” walk. Further, they generate *Fourier characteristic functions* for different emotions by taking the difference between the Fourier coefficients of a functional model for an emotion-influenced locomotion and those for a neutral locomotion. This allows them to superimpose emotions onto models for other motions. For instance, they can superimpose “briskness” or “tiredness” onto a “run” model, even if the characteristic functions for “brisk” and “tired” were defined from brisk and tired *walks*. In addition, they provide interactive controls for step, speed, and hip position. In [80], they show examples of walking and running with the following emotions: hollow, vacant, graceful, cold, brisk, happy, vivid, and hot. They seem to have captured a wide variation in movements, however, their methods work only on cyclic motions. Also, although users can interactively adjust values for the modeled emotions, the addition of other emotions requires repeating the lengthy process of motion capture on a human subject and generating its functional model and characteristic function.

Amaya, Bruderlin, and Calvert present a more general method for adding emotion to motions. They derive *emotional transforms* from motion capture data by quantifying the differences between neutral and emotion-driven actions using the speed of the end-effector and the spatial amplitude of joint angle signals [1]. They then use the emotional transforms to add emotion to neutral actions. Further, by dividing the joints of an articulated human figure into joint categories, they are able to apply emotional transforms derived from one part of the body to another part of the body. For instance, an emotional transform derived from angry and sad arm movements (drinking motions) was applied to the legs to generate angry and sad kicking motions. Further, their emotional transforms can be applied to simulated, keyframed, and procedurally generated motions. The authors use the technique to capture ten emotions or moods: neutral, angry, sad, happy, fearful, tired, strong, weak, excited, and relaxed. By separating the definition of basic movements from the transforms required to generate emotions, the authors are able to generate a broad range of movements with different types of expressivity. They note however that individuals differ in the ways they express themselves, requiring different transforms to represent different personalities, genders, cultures, and ages, which could result in a large database of movement transforms.

In [16], Bruderlin and Williams introduce a set of techniques to modify existing motion data generated from motion capture, keyframing, or procedural animation . By treating motion parameters (such as joint angles or coordinates) as sampled signals, they can apply techniques from image and signal processing to modify the animated motions. Multiresolution motion filtering passes a motion parameter signal through a series of filters, decomposing the signal into a set of bandpass filter bands. An animator can adjust the amplitudes of high, middle, or low frequency bands to add a nervous twitch, exaggerate the movement, or constrain joint ranges, respectively. Multitarget motion interpolation blends two different motions into a single motion. Dynamic timewarping resolves timing differences between two motions to be blended. Waveshaping modifies input motions using shaping functions and is useful for maintaining joint limits or adding subtle effects. Motion displacement mapping permits local shaping of a signal while maintaining the global shape of the signal. This allows an animator to change select keyframes (for instance, to satisfy certain constraints or change an end effector position) while maintaining the overall character of the original motion. The system fits a spline curve through the displacements in each degree of freedom and adds it to the original curve signal.

Witkin and Popović describe a technique, similar to motion displacement mapping, for editing captured or keyframed animation by warping motion parameter curves [86]. The animator modifies the pose at particular frames, which are used as constraints on a smooth deformation to be applied to the captured motion curves. The deformation satisfies the constraints, while maintaining the fine details of the captured motion. A large number of realistic motions may be created from a single prototype motion sequence using just a few keyframes to define the motion warp.

Rose, Cohen, and Bodenheimer present a method for leveraging existing motions—they interpolate between structurally similar example motions along multiple dimensions to create new motion [70]. Using an off-line authoring system, they parameterize the motion "verbs" with "adverbs" and create a verb graph to specify transitions between verbs. At runtime, these structures allow the user to modify animations in real-time by changing adverb settings.

Koga, Kondo, Kuffner, and Latombe present a task-level animation system that generates arm motions of a human figure moving an object to a goal location [47]. They introduce a planner to compute collision-free paths and an inverse kinematics algorithm for human arms based on neurophysiological studies. Though their system generates a complete animation with minimal user specification, their focus is on the "intention" of moving an object from one location to another and not on the underlying movement qualities of the character and their expressive manifestations.

Morawetz and Calvert introduce a framework for an animation system, which perhaps most closely matches our high-level goals, but takes a very different approach [61]. They implement a portion of this framework as the GESTURE system, which uses a mock expert system to add secondary motion to walking characters based on their user-specified "personality" and "mood". Secondary movements are the subtle movements that don't serve to satisfy a particular goal, but often reflect one's subconscious, inner attitudes and play an important role in nonverbal communication. For instance, reaching for a cup or opening a door are goal-directed primary movements, while tapping one's foot or scratching one's head are examples of secondary motion. Personality is specified through slider values for extrovert/introvert, cheerful/gloomy, assertive/passive, and domineering/submissive. Moods include boredom, nervousness, fatigue, impatience, and fear. For motion specification, GESTURE uses a graph to facilitate the sequencing of movements and to allow for one movement to interrupt another. GESTURE takes as input a high-level script, which chronologically lists actor movements and start times using textual descriptions (such as "walk forward" or "scratch head with left hand") and frame numbers. Each high-level movement must be defined using a pre-defined gesture specification function. The system generates a detailed animation script, specifying all joint angles for all characters in an animation.

2.3 Biomechanics

There has been a significant amount of research on the biomechanics of human motion; however, the field is still fairly young with no definitive models and several working hypotheses. For instance, the minimum-jerk hypothesis suggests that people performing skilled tasks move in a maximally smooth manner, minimizing the time rate of change of acceleration [31]. Another model, the minimum-torque change hypothesis, suggests that motions display minimized torque derivatives [78]. Zatsiorsky provides a good introductory text on elementary movement concepts as well as the biomechanical research literature [88]. As none of the biomechanical models can explain all the complexities of human movement, we take the artistic approach in developing our basic movement model (Chapter 4). We further justify this approach by the fact that actors (human as well as animated) tend to move differently from humans in everyday situations—they must exaggerate their movements, emotions, and reactions to capture an audience's attention.

2.4 Computers and Dance Notation

For the past twenty years, people have been exploring ways to combine computers and dance notation, both to simplify the process of notating dance and to facilitate computer animation by using notations already established to describe human movement. Several software programs exist for editing and printing dance notation scores. For instance, there is LabanWriter [52] and LED [40] for Labanotation, and MacBenesh [56] for Benesh Movement Notation. Others have used notation, Labanotation in particular, to animate human figures [3, 18]. These projects have focused on the structural aspects of movement, which are specified by Labanotation, but have not addressed the more qualitative aspects of movement provided by Effort. Bishko suggests analogies between the "Twelve Principles of Animation" [75] and Laban Movement Analysis (LMA) [11]. She shows that there is an abstract relationship between LMA and traditional animation techniques, but does not provide a practical means of exploiting this relationship. Badler was the first to propose the use of Effort as a higher level of control for human figure animation [4]. Our work here provides a method of implementing just such a system.

Since Effort provides a universal theory of human expression and a vocabulary for movement dynamics, our motion control paradigm eliminates the need to build a database of specific expressions and behaviors. Also, since Effort is expressed by combining and varying the magnitude along just four motion factors, we are able to re-use our Effort model to specify all types of expressive movements without requiring off-line modeling for new expressions. Further, our Effort model is computationally efficient, performing in real-time and allowing for interactive motion generation and editing.

Chapter 3

Background

Describing human motion is a formidable task. In dance alone, there are over a dozen significant notations, each varying in approach, aims, strengths, and weaknesses [37]. The Random House Word Menu has over five hundred verbs of motion [32]. Laban describes movement as "one of man's languages" [49]; unfortunately, translating movement into a non-visual, spoken or textual language is not straightforward.

To achieve our goals for a means of specifying expressive movements of computer-animated characters, we sought a language for describing the qualitative aspects of movement that was:

- systematic with a limited number of terms (the set of adverbs in the English language is unwieldy),
- intuitive, so that users do not have to learn a new notation or express how they want a movement to be performed in abstract, quantitative terms,
- objective enough that different people can agree on the meanings of the terms, and
- not limiting on the range of represented movements.

We examined the nonverbal communication and dance notation literature for methods of describing, notating, and recording human movement. We found that the Effort component of Laban Movement Analysis was most appropriate for our purposes. This chapter surveys the research in nonverbal communication and describes how other notations used to study human behavior are inadequate for our purposes. Then, we provide justifications for the use of Laban's notation as a research tool, give a brief history of Laban's work, and describe Effort as a method for qualitative movement description.

3.1 Nonverbal Communication

Nonverbal communication encompasses research in a variety of disciplines and seeks to answer a wide range of questions. Significant advances have been made in our understanding of nonverbal communication; however, the field still seems to suffer from overlapping terminology, categorizations, and ill-defined boundaries. Researchers in nonverbal behavior include specialists from psychology, psychiatry, anthropology, sociology, dance notation, ethology, education, and the performing arts. They seek theories on the expression of emotion and personality, psycho-pathological diagnoses and treatments, cultural characteristics of movement, developmental motor processes, psychological implications of gesture and posture, influences of affect and attitude, comparison to animal behavior, and a slew of other issues. Body movement is a key area of study, but facial expression, visual behavior, paralanguage, proxemics (use of space and distance), tactile behavior and multichannel communication are also considered major topics in nonverbal communication. We do not attempt to integrate or categorize the large body of research in nonverbal communication and body movement, since that is beyond the scope of this work. However, we point the interested reader to the readings and commentaries in [83], which offer some insight into the study of body movement and gesture, and several extensive bibliographies on nonverbal communication [23, 43, 44].

Our goal differs from much work in nonverbal communication in that we are not seeking universal theories on the communicative implications of movements or any underlying psychological meaning; nor are we seeking comparisons or generalizations between individuals of different mental states, ages, cultures, or even species. Instead, we seek a comprehensive *description* system of human movements that may provide a useful interface for controlling figure animation. Thus, we are most interested in description and notation systems for coding body movements. Wallbott surveys nonverbal behavior measurement and observation systems, noting that they tend to form a continuum between objective systems making physical measurements and subjective notations that require human observers making inferences and interpretations [81, 82]. We seek a notation somewhere between these two extremes. We desire a system that describes qualitative not quantitative aspects of movement, yet is objective enough to prove intuitive to laypersons and to ensure inter-observer agreement. Many of the systems for measuring body movement do not meet our needs, because they are cumbersome and non-intuitive, or focus solely on spatial aspects of movement. Similarly, many dance notations are either particular to a specific style of dance, lack methods of describing movements not used in dance, or have no means of describing the expressive nature of movement. There are three major dance notation systems currently in use: Benesh Movement Notation[9], Eshkol-Wachmann [30], and Labanotation [41]. Benesh notation has been used primarily to notate ballets and was adopted by the Royal Ballet in London. Eshkol-Wachmann notation uses a numerical system for recording movement, focusing on joint angles and spatial patterns. Only notations based on the work of Rudolf Laban seem to have the potential to satisfy our needs.

The use of Labanotation and its derivative notations has extended beyond the dance community to become a valuable tool for nonverbal communication research. Davis evaluates the "logic and consistency" of Effort-Shape Analysis[1] (an offshoot of Labanotation) promoting its use as a research tool [22]. In [6], Bartenieff and Davis justify the use of Laban's notation for the study of behavior. Their justifications match our requirements for a language to describe and control computer figure animation. They point out that many behavioral studies use subjective, detailed descriptions of movement and postures, and argue for the need for a *systematic*, *objective* notation with a *limited* number of descriptive terms. They demonstrate that Effort-Shape provides such a system, and further, they discuss the hypothesis that there may be neurophysiological support for the selection of its basic variables. In [24], Davis further details the evolution of Laban's work, describing his students' extension and application of his ideas, and surveys the early use of Laban analysis in behavior research.

---

[1] Effort-Shape later evolved into Laban Movement Analysis.

3.2 Laban Movement Analysis

Rudolf Laban (1879-1958) made significant contributions to the study of movement, bringing together his experiences as a dancer, choreographer, architect, painter, scientist, notator, philosopher, and educator. He observed the movement of people performing all types of tasks: from dancers to factory workers, fencers to people performing cultural ceremonies, mental patients to managers and company executives. His theories on movement and its extensions by his students and colleagues have resulted in a rich vocabulary for describing and analyzing movement. He also developed a movement notation system, which has evolved and expanded into a number of related and overlapping variations, including Labanotation, Kinetography Laban, Effort-Shape, and Laban Movement Analysis. The International Council on Kinetography Laban (ICKL) was established in 1959 to standardize the notation and to unify some of the differences between the various forms. Labanotation, which was adopted by the Dance Notation Bureau [17], focuses on the structural aspects of movement and provides a very exact notation that allows dancers to reproduce a dance solely from its score [41]. Kinetography Laban is essentially the same as Labanotation except for minor differences in notation usage and rules [45]. Effort-Shape developed somewhat independently and with an emphasis on the qualitative, dynamic aspects of movement [6]. Effort-Shape spawned the development of Laban Movement Analysis (LMA) [7, 59, 27, 57], which is promoted by the Laban/Bartenieff Institute for Movement Studies [63]. LMA has evolved into a more comprehensive system and has been used in dance, drama, nonverbal research, psychology, anthropology, ergonomics, physical therapy, and many movement-related fields [22, 6, 24]. The variance between the systems is partially due to historical reasons and all remain consistent with Laban's original philosophies.

Moore and Yamamoto enumerate five principles that are fundamental to Laban's theories ([59], Chapter 9):

1. Movement is a process of change.
2. The change is patterned and orderly.
3. Human movement is intentional.

4. The basic elements of human movement may be articulated and studied.

5. Movement must be approached at multiple levels if it is to be properly understood.

These basic principles form the theoretical foundation of LMA.

LMA is divided into four major components: Body, Space, Shape, and Effort[2]. Together these components constitute a language for describing movement. Body deals with the parts of the body that are used and the initiation and sequencing of a motion. Space describes the locale, directions, and paths of a movement. Shape involves the changing forms that the body makes in space. Effort describes the qualitative aspects of movement and is often compared to dynamic musical terms such as legato, staccato, forte, dolce, etc., which give information on how a piece of music should be performed.

Movement is often described in terms of actions or *what* one does. However, we are interested in *how* one moves, which is precisely what is provided by Effort. Moore explains that, "While the uses of [S]pace and of the [B]ody reveal the mover's purposes, Laban believed that the uses of energy, or the dynamics of an action, [Effort] were particularly evocative of intentions" ([59], pp 185). Maletic goes further to say that "Laban sees Effort as the inner impulse—a movement sensation, a thought, a feeling or emotion—from which movement originates; it constitutes the link between mental and physical components of movement"([57], pp 179). She continues, saying that "one may conclude that the concept of Effort unifies the actual, physical, quantitative and measurable properties of movement with the virtual, perceivable, qualitative, and classifiable qualities of movement and dance"([57], pp 101). We note that neither Laban nor anyone else has ever specified these "quantitative ... properties of movement"; this is precisely the contribution of our Effort model (Chapter 4).

Effort comprises four motion factors: Space, Weight, Time, and Flow. Each motion factor is a continuum between two extremes: (1) indulging in the quality and (2) fighting against the quality[3]. These extreme Effort Elements are seen as basic, "irreducible"

---

[2] Throughout this document, we capitalize key terms defined by LMA to distinguish them from their common English language usage.

[3] Some meaning is lost in the translation of Laban's original German texts into English. The German word "antrieb" has been translated into "effort". "Antrieb" has an implication of the existence of a force that drives or propels something. The German term "ballung" has markedly different connotations from its translation of "fighting". Ballung connotes a coming together or clustering; it is a condensing

qualities—they are the smallest units of change in an observed movement. The eight Effort Elements are: Indirect, Direct, Light, Strong, Sustained, Sudden, Free, and Bound. Table 3.1 illustrates the motion factors, listing their opposing Effort Elements with textual descriptions and examples.

We note that we are not trying to provide a tool to facilitate the recording of Effort, but rather we use the language provided by Effort as an interface for controlling expressive figure movements. Aside from handling the individual Effort Elements, there are several other issues that must be handled in order to capture the full extent of expressivity afforded by Effort. First, human movements span the range along each motion factor continuum. Also, movements rarely involve only one motion factor; more often they display combinations of Effort Elements from different motion factors and at varying intensities. Another issue is phrasing. Certified Movement Analysts (CMAs) trained in Laban Movement Analysis observe movements as a subtle sequence of Effort changes or phrases. For instance, in general terms one might say that a movement was "light". However, a skilled Effort observer might observe that, "The movement began with a quickness, then became light, and ended with a sustained indirectness." To exploit the richness of Effort descriptions, one must allow users to use both general Effort descriptions as well as detailed Effort phrasing. Finally, we note that individuals tend to display particular Effort patterns, although they may consciously learn to expand their Effort repertoire. An Effort-based system should support individualized expression by allowing users to customize Effort Element settings for different characters. We address these issues in Chapter 4, where we describe our Effort model.

We believe that the Effort descriptions of Laban Movement Analysis provide an adequate interface to controlling the expressiveness of computer animated figures. Effort seems to generate the space of human movements, with its use justified by numerous researchers in a variety disciplines. Further, Effort proves intuitive by providing a small number of textual descriptors as compared to detailed, cumbersome notations or mathematical and physics-based parameters for describing expression.

---

or agglomeration. For instance, the term ballung might be used to describe the coming together and condensing of particles to form storm clouds [66].

Space – attention to the surroundings	
Indirect	flexible, meandering, wandering, multi-focus
examples:	waving away bugs, slashing through plant growth surveying a crowd of people, scanning a room for misplaced keys
Direct	single focus, channeled, undeviating
examples:	pointing to a particular spot, threading a needle, describing the exact outline of an object
Weight – attitude towards the impact of one's movement	
Light	buoyant, delicate, easily overcoming gravity, marked by decreasing pressure
examples:	dabbing paint on a canvas, pulling out a splinter, describing the movement of a feather
Strong	powerful, having an impact, increasing pressure into the movement
examples:	punching, pushing a heavy object, wringing a towel, expressing a firmly held opinion
Time – lack or sense of urgency	
Sustained	lingering, leisurely, indulging in time
examples:	stretching to yawn, stroking a pet
Sudden	hurried, urgent
examples	swatting a fly, lunging to catch a ball, grabbing a child from the path of danger, making a snap decision
Flow – amount of control and bodily tension	
Free	uncontrolled, abandoned, unable to stop in the course of the movement
examples:	waving wildly, shaking off water, flinging a rock into a pond
Bound	controlled, restrained
examples:	moving in slow motion, tai chi, fighting back tears, carefully carrying a cup of hot liquid

Table 3.1: Motion Factors and Effort Elements

Chapter 4

Effort Model

In order to use Effort for computer animation, we needed a quantitative Effort model. This chapter discusses the method we used to build an empirical model of Effort using quantitative, low-level movement parameters. We describe the set of low-level movement parameters and show how Effort settings are used to compute movement parameter values.

Our current Effort model has been developed with the intent of creating expressive arm movements, although many aspects of our model are applicable to expressive movements displayed in other body parts. We elected to focus on arm movements because the arms are a primary means of bodily expression and communication. Also, many task-oriented movements involve the use of the arms. Further, the extensive reach and joint angle ranges in the arms allow for a wide variation in movements (as compared, for instance, to head nods or torso bends). Finally, although a number of researchers have addressed issues of locomotion and its influence under various emotions, *expressive* arm movements have been essentially neglected.

4.1 Translating Effort into Movement Parameters

The translation of the qualitative Effort Elements into quantitative, low-level movement parameters was the key task in implementing a system using Effort for motion control. Initially, we tried to deduce movement characteristics from motion capture data. We collected 3D motion capture data and made a video recording of a Certified Movement

Analyst (CMA) trained in Laban Movement Analysis performing several examples of each combination of two and three Effort Elements[1]. We used 12 electromagnetic sensors: one each on the head, sternum, stomach, and back; and one on each shoulder, elbow, wrist, and knee. Analysis of the motion capture data led to only the most obvious conclusions; i.e.: Sudden is short in duration, Sustained is longer in duration, and Strong tends to have large accelerations. The inability to deduce the more subtle characteristic qualities of Effort arose from several factors. First, Effort reflects complex inner physiological processes that are related to a being's inner drive to respond to the physical forces in nature. Thus, Effort is embodied in the whole person and manifested in *all* body parts, whereas we were interested solely in the physical embodiment and visual result of inner attitudes on *movement*, particularly that of the arms. Furthermore, numerous other movements such as visual attention, changes in muscular tension, facial expressions, and breath patterns are not adequately captured by current motion capture technology.

As a result, we turned to other methods for developing an empirical model of Effort. Initially, we used visual analysis of the playback of the motion capture data of a CMA performing Effort combinations as a stick figure formed by connecting the blocks representing the sensors (Fig. 4.1). This allowed us to focus our attention solely on the influence of Effort on one's gross body *movement* by extracting out other more subtle manifestations of Effort (facial expression, visual attention, etc.). We then made repeated and careful analyses of the video of the Effort performance and used descriptions of Effort from the literature [7, 27, 57, 59] to deduce some of the tangible qualities of Effort that we needed to capture. In determining the underlying motion parameters to use to model the Effort Elements, we experimented with and extended computer animation methods (such as velocity curves and interpolation), as well as traditional animation principles (such as anticipation, overshoot, squash and stretch) [53, 75]. Finally, we performed numerous experiments by having a CMA [66] mark the Efforts present in video segments and having her experiment with low-level parameters in our system EMOTE (Chapter 5). This also induced more suggestions on qualities of Effort that would add to our portrayal of the Effort Elements.

---

[1] Movements displaying a single Effort element or a combination of four Effort Elements are rare, and thus, were not included in the recording.

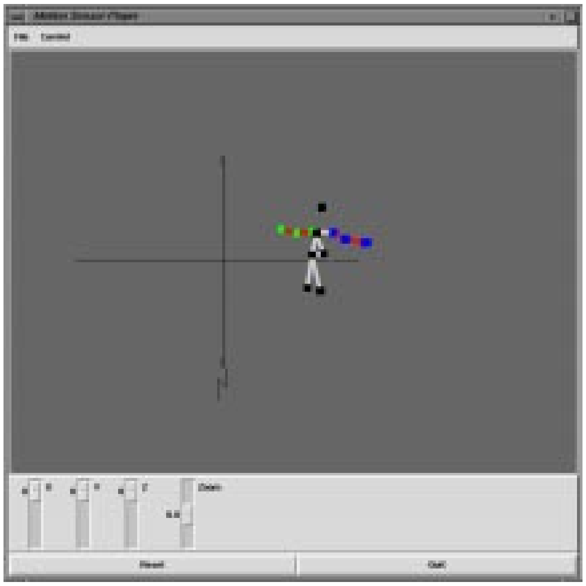


Figure 4.1: Stick Figure Formed By Motion Capture Sensors

4.2 Low-level Movement Parameter Definitions

In selecting the set of low-level movement parameters, we chose a kinematic model over a dynamics-based or hybrid implementation of Effort. Dynamics-based techniques require computationally costly calculations, limiting our desire for interactivity. Also, although dynamic simulations generate physically accurate motions, these motions often lack the nuances and flourishes of expressive human movements. Further, kinematic models have proven reasonable at portraying physics-based models [87], and our models for Flow and Weight combinations adequately generate the impression of mass, force, inertia, and other physical phenomena.

In the following section, we define the set of low-level movement parameters in our model. These are divided into three categories: those that affect the arm trajectory, those that affect timing, and flourishes that add to the expressiveness of the movement.

4.2.1 Trajectory Definition

We define the arm trajectory for a given animation with two parameters:

- **Path curvature** determines the straightness or roundness of the path segments between keypoints. We control the path curvature using the tension parameter introduced by Kochanek and Bartels for interpolating splines [46]. The tension parameter ranges from -1 to $+1$. Decreasing the tension value gives rounder bends at the keypoints, while increasing the value results in a tighter curve with straighter segments between points.
- The **interpolation space** defines the space in which the interpolation is performed: end-effector position, joint angle, or elbow position.

For end-effector interpolation, we use the end-effector position and swivel angle stored for each keypoint. We define an interpolating spline between the positions at keypoints using the tension parameter to determine the curvature of the path. We also interpolate between swivel angle values with an interpolating spline. For joint angle interpolation, we compute and store the shoulder and elbow rotations at keypoints. We then generate an interpolating spline between the elbow angle values at keypoints and perform spherical linear interpolation to determine the shoulder rotations. For interpolation in elbow position space, we compute and store the elbow position at keypoints using the posture defined by the end-effector position and swivel angle. We then define an interpolating spline between these positions, which are later used to set the shoulder rotations. The elbow rotations for elbow position interpolation are the same as those computed for end-effector interpolation. Interpolation in elbow position space gives smooth elbow motions with a less path-driven movement than interpolation in end-effector position space.

The Effort settings determine which interpolation space is used. The default interpolation space is end-effector position. Free movements use angular interpolation to achieve a less path-driven and less controlled movement. Our empirical studies show that Indirect movements tend to be driven by the elbow, and thus we interpolate them in elbow position space.

4.2.2 Parameterized Timing Control

We separate timing control from trajectory definition by using a variation of the double interpolant method introduced by Steketee and Badler [74]. The interpolating splines that define the trajectory (described in the preceding section) compute values between keypoints using an interpolation parameter s that varies from 0 to 1 over the interval from keypoint i to keypoint $i+1$ [46]. Let the trajectory be defined by some function $P(s,i)$. To obtain points on the trajectory, we need a method for translating (in-between) frame numbers into s and i. At each keypoint, $s=0$ and i is the number of the current keypoint. For in-between frames, we define a variable $t' \epsilon [0,1]$ to represent a frame's relative time between the previous and following keypoints. Let $prev$ equal the frame number of the previous keypoint, $next$ equal the frame number of the next keypoint, and $curr$ equal the current frame number. Then,

$$t' = \frac{curr - prev}{next - prev}. \tag{4.1}$$

We define a timing control function

$$Q(t', \vec{I}) = s, \tag{4.2}$$

where $\vec{I}$ is a four-dimensional vector whose components specify various timing effects (described further below).

For each in-between frame, we

1. compute t', the frame's normalized time between keypoints, using Equation 4.1,
2. compute the interpolation parameter s using function Q (Equation 4.2), and then
3. input s and the corresponding keypoint number i into function P to compute the position values (or joint angle values for angular interpolation) for the given frame.

We provide several parameters for timing control:

- The **number of frames between keypoints** is initially set according to the user's specified key times, but these values get adjusted according to the Effort settings. The $nfmult$ parameter is a multiplier that increases ($nfmult > 1$) or decreases ($0 > nfmult > 1$) the number of frames between two keypoints.
- **The components of $\vec{I}$** include inflection time t_i, time exponent $texp$, start velocity v_0, and end velocity v_1.

Our parameterized timing control function Q assumes every movement (from one goal keypoint to the next) starts and ends at rest. Also, every movement has a constant acceleration until time t_i, followed by a constant deceleration[2]. We introduce velocities v_0 at time t_0 and v_1 at time t_1 to achieve the traditional animation effects of anticipation and overshoot. This model gives us the following velocity function (Fig. 4.2):

$$v(t'') = \begin{cases} \frac{-v_0}{t_0} t'' & [0, t_0) \\ \frac{-(v_0+t_i)t''+v_0 t_i+t_0 t_i}{t_0-t_i} & [t_0, t_i) \\ \frac{-(v_1+t_i)t''+v_1 t_i+t_1 t_i}{t_1-t_i} & [t_i, t_1) \\ \frac{-v_1 t''+v_1}{t_1-1} & [t_1, 1] \end{cases} \tag{4.3}$$

where

$$t'' = (t')^{texp}. \tag{4.4}$$

The function Q is the integral of Equation (4.3).

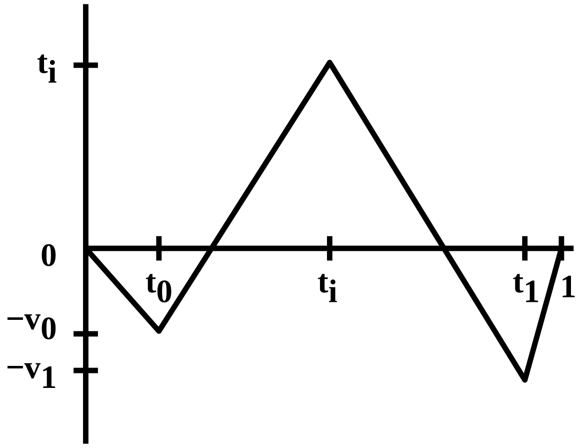


Figure 4.2: Velocity Function

The components of $\vec{I}$ to the timing control function Q provide control to the acceleration/deceleration pattern of the movement, as well as allowing for anticipation and overshoot. The inflection point $t_i \epsilon [0, 1]$ represents the point between two goal keypoints where the movement changes from accelerating to decelerating. A value of 0.5 gives a basic

---

[2] As none of the existing models for human motor control explain all the characteristics of human movement, we use an ease-in/ease-out function, popular in current animation technology, as an approximation for an expressionless movement profile.

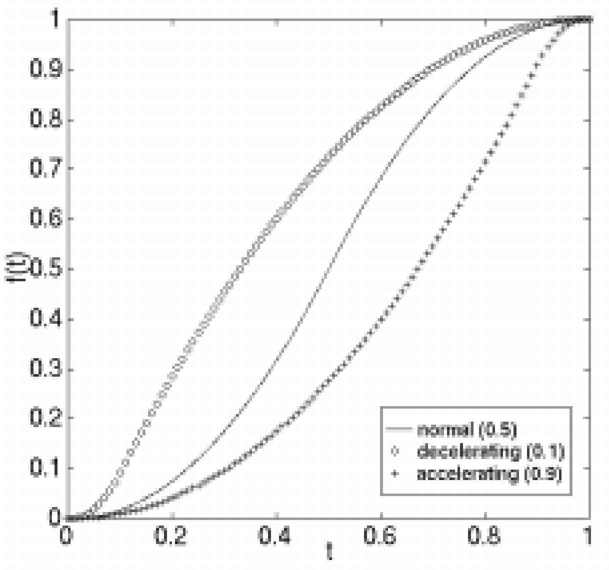


Figure 4.3: Varying Inflection Point to Obtain Acceleration and Deceleration (Inflection Point Value Given in Parentheses)

ease-in/ease-out curve. A value greater than 0.5 corresponds to a primarily accelerating motion, while a value less than 0.5 gives a decelerating motion (Fig. 4.3). The default time exponent ($texp$) value of 1.0 does not affect the velocity curve; however, values greater than 1.0 magnify an acceleration, while values less than 1.0 exaggerate a deceleration (Fig. 4.4). The start (v_0) and end (v_1) velocities[3] default to 0. Increasing v_0 generates movements with anticipation, where the hand pulls back before extending, such as in preparation for a Strong movement. Increasing v_1 generates movements with overshoot, such as in Free movements where an indulgence in flow causes one to swing out past a target before hitting it (Fig. 4.5). We set t_0 to 0.01 and t_1 to 0.99, which gives us natural-looking anticipation and overshoot effects; however, these values can easily be included in $\vec{I}$ as variable low-level movement parameters.

---

[3] As mentioned, each movement begins and ends at rest. The start and end velocities represent times shortly after the beginning or shortly before the end of a movement, respectively. They are so named to emphasize that they are not initial and final velocities, which remain 0.

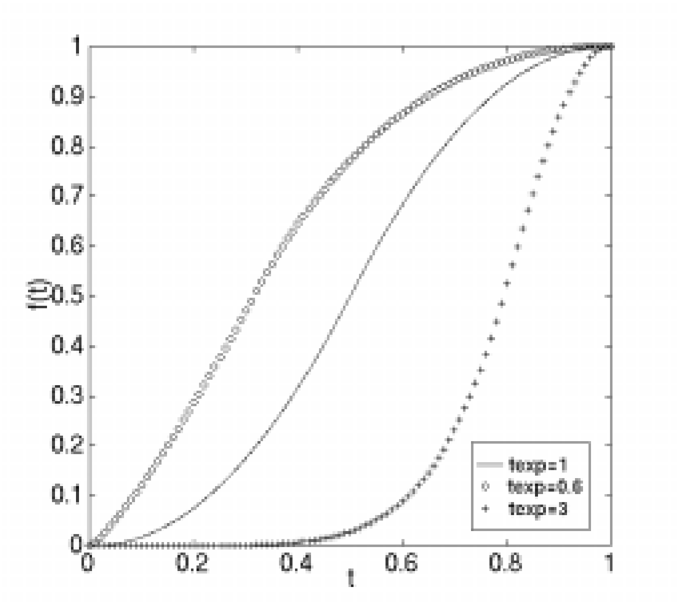


Figure 4.4: Varying Time Exponent to Magnify Acceleration and Deceleration

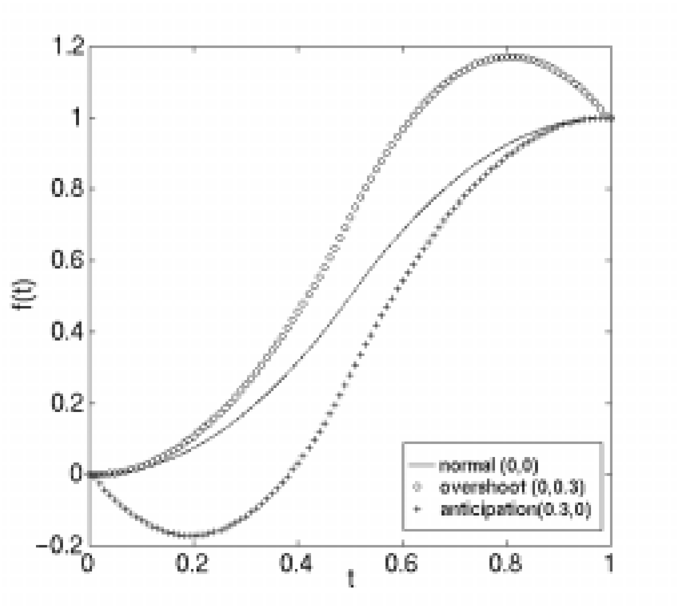


Figure 4.5: Varying Initial and Final Velocities to Obtain Anticipation and Overshoot (Initial and Final Velocities Given in Parentheses)

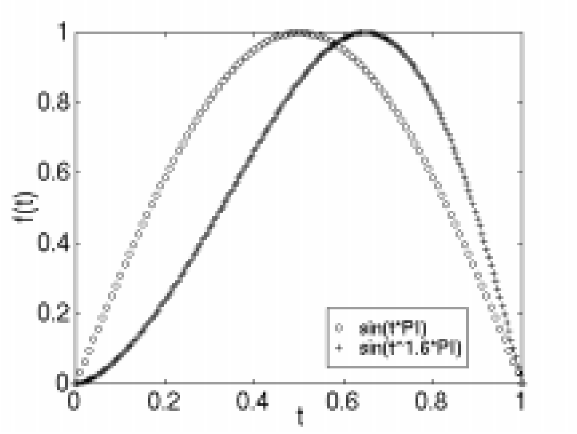


Figure 4.6: Sine Factor in Squash Equation

4.2.3 Flourishes

Flourishes are miscellaneous parameters that add to the expressiveness of the movements. They are listed below:

- **Squash and stretch** is a traditional animation technique where non-rigid objects deform, giving a dynamic, life-like quality [53, 75]. We employ squash and stretch by scaling the body of simple characters to simulate the expansion and contraction of the human torso. We parameterize squash and stretch with the variable *squashmag*. The squash is computed as:

  $$squash = 1 + (squashmag * sin(t'^{1.6} * \pi)), \tag{4.5}$$

  where t' is the normalized time between two keypoints. The *squash* value is then used to scale the body by $1.0/sqrt(squash)$ in x (width) and z (depth), and by *squash* in y (height); thus, maintaining the original body volume. Fig. 4.6 illustrates the motivation for the sine factor. The **o** line illustrates the normal sine curve; the + line shows that our sine factor skews the curve slightly towards 1. This squash computation gives a normal-sized torso which gradually elongates, then more quickly returns to its normal size between each keypoint.

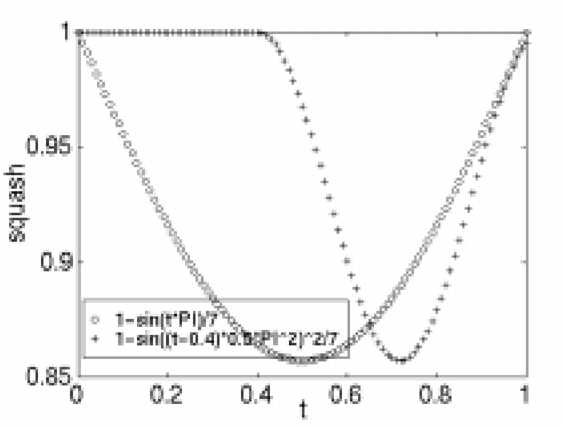


Figure 4.7: Sine Factor in Equation for Breath

- **Breath** refers to the noticeable exhale one makes when executing powerful movements. To simulate breath, we re-use the *squash* variable, setting it with a computation based on t', a frame's relative time between the previous and following keypoints. An exhale begins with an unsquashed torso, where *squash* is 1.0. When t' is greater than 0.4, we set the squash using the sinusoidal function (Fig. 4.7):

  $$squash = 1 - \frac{sin^2(\frac{\pi^2}{2} * (t' - 0.4))}{7} \tag{4.6}$$

  This results in a normal-sized torso that squashes (shortens and widens) and returns to normal towards the end of each movement.

- **Wrist bend** is determined by the wrist bend multiplier *wbmag* and the wrist extension magnitude *wxmag*. The *wbmag* parameter is a multiplier that represents the magnitude of the wrist bend. If the *wbmag* is set for a flexed wrist, the wrist bend is set to 0.6 radians about the y-axis. Otherwise, the wrist bend is set using

  $$wrist\_bend = wbmag * sin(2\pi(t' + 0.75)) + 1 - wxmag), \tag{4.7}$$

  where $t' \epsilon [0, 1]$ and represents the normalized time between two keypoints. The shifted sine factor (Fig. 4.8) results in a wrist that gradually bends inwards and back out. The value of *wxmag* shifts the sinusoidal graph up or down, setting the beginning wrist extension to be positive (outward) or negative (inward).

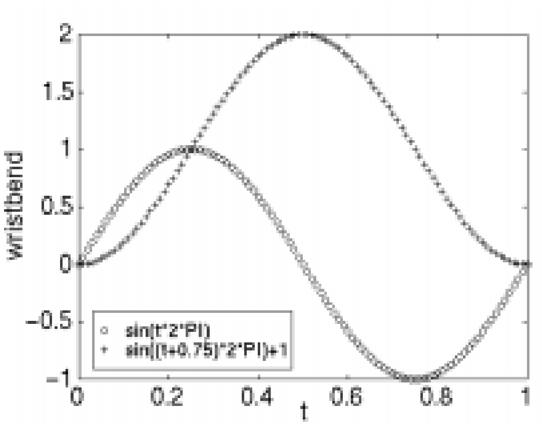


Figure 4.8: Multiplier in Wrist Bend Equation

- **Arm twist** is parameterized by wrist twist magnitude $wtmag$, wrist frequency $wfmag$, elbow twist magnitude $etmag$, and elbow frequency $efmag$. The wrist twist is measured in radians about the z-axis and is determined by:

$$wrist\_twist = wtmag * sin(wfmag * \pi t'). \tag{4.8}$$

  Elbow twist is set using a similar equation, replacing $wtmag$ and $wfmag$ with $etmag$ and $efmag$, respectively.

- **Displacement magnitude** is a multiplier $dmag$ that adds a sinusoidal displacement to the elbow angle

$$elbow\_angle = elbow\_angle * (1 + dmag * sin(2\pi t')) \tag{4.9}$$

  where t' is the normalized time between two keypoints.

- **Limb volume** refers to the swelling of the biceps muscles that is particularly noticeable in powerful movements. We simulate biceps flexion by scaling the width and depth of the upper arm by $1 + 0.7 * (elbow\_angle * awmag)$, where $elbow\_angle$ is the current elbow angle and $awmag$ is a low-level movement parameter determined by the Effort settings.

4.3 Parameter Settings

To determine the mapping of the four Effort motion factors into our low-level movement parameters, we first determined the default settings for each of the eight Effort Elements. Next, we devised a scheme for generating the range between opposing Effort Elements and for combining Elements from different motion factors. Finally, we expressed our Effort model as a set of mathematical equations.

4.3.1 Parameter Settings for Individual Effort Elements

Table 4.1 presents the parameter settings for the individual Effort Elements. We started by defining a generic, expressionless motion similar to what would be generated using traditional keyframe animation systems. The generic motion has normal path curvature. There is no squash and stretch of the body, and no wrist or elbow twists. Also, there is no wrist bend so the wrist stays aligned to the forearm throughout the motion, and the arm maintains normal limb volume. The movement is interpolated in end-effector space using a standard ease-in/ease-out velocity curve with no anticipation or overshoot. Now, we give some explanation behind the settings for the individual Effort Elements. We note that the exact values were determined by a significant amount of trial and error using visual analysis and testing by a CMA.

A video of a CMA performing various Effort combinations shows that the elbow often leads in Indirect arm movements and are characterized by arm twists and curved paths. To capture these qualities, we interpolate in elbow position space, adding a sinusoidal displacement for extra variation in space. Also, we add elbow and wrist twists and use a significant amount of wrist bend.

Direct motions are focused and path-oriented, so we interpolate in end-effector space. The default motion path for Direct motions generates straight (rather than curved) line segments between keypoints.

Movements associated with weight (Strong and Light) usually have marked breath patterns; we approximate these by varying the squash and stretch of the body. Strong movements contract slightly near the end, much like a person exerting a forceful exhale.

Parameter	**Neutral**	**Indirect**	**Direct**	**Light**	**Strong**	**Sustained**	**Sudden**	**Free**	**Bound**
Path Curvature ($Tval$)	normal 0	curved -1	straight +1	normal 0	normal 0	none 0	none 0	none 0	none 0
Interp. Space	end-eff	elbow posn	end-eff	end-eff	end-eff	end-eff	end-eff	angular	end-eff
Num Frames Multiplier ($nfmult$)	normal 1	normal 1	normal 1	normal 1	normal 1	lengthen 3	shorten 0.33	normal 1	lengthen 3
Acceleration (t_i,$texp$)	ease 0.5,1	ease 0.5,1	ease 0.5,1	decl 0.1,1	accel 0.9,0.8	decl 0.1,1	accel 0.9,3	ease 0.5,0.6	ease 0.5,1
Antic./Oversh. (v_0,v_1)	none 0,0	none 0,0	none 0,0	oversh 0,0.03	antic 0.1,0	oversh 0,0.03	none 0,0	oversh 0,0.20	none 0,0
Squash & Stretch ($squashmag$)	none 0	none 0	none 0	yes 0.20	breath	none 0	none 0	yes 0.15	none 0
Wrist Bend ($wbmag$,$wxmag$)	none 0,0	yes 0.6,0.0	none 0,0	yes 0.5,-0.3	flexed	none 0,0	none 0,0	yes 1.0,0.2	flexed
Wrist Twist ($wtmag$,$wfmag$)	none 0,0	yes 0.4,2	none 0,0	none 0,0	none 0,0	none 0,0	none 0,0	none 0,0	none 0,0
Elbow Twist ($etmag$,$efmag$)	none 0,0	yes 0.4,2	none 0,0	none 0,0	none 0,0	none 0,0	none 0,0	none 0,0	none 0,0
Displ. Mag.	0	0.4	0	0	0	0	0	0	0
Limb Volume ($awmag$)	normal 0	normal 0	normal 0	normal 0	magnified 0.25	normal 0	normal 0	normal 0	normal 0

Table 4.1: Low-level Parameter Settings for Effort Elements

For Light movements, the squash and stretch effects vary depending on the end-effector locations. When the hand is high above the figure, the body is stretched vertically; for positions horizontally distant from the figure, we widen the body.

Light movements use a significant amount of wrist bend to achieve an airy quality and enhance the lightness. To achieve the buoyant quality of light movements, we use a slight overshoot and decelerating velocity curve.

Strong motions tend to display a marked acceleration corresponding to the force exerted. Strong movements also use anticipation to emphasize the preparation required to exert that force. The limb volume is magnified to simulate muscle flexion, and the wrist remains in a flexed position throughout the movement.

Sustained movements indulge in time, displaying a marked deceleration [66]. A small amount of overshoot captures the slight rebound at the end of a sustained “sigh”. Sustained movements usually have a longer duration than other movements, and thus, we increase the number of frames between keypoints. On the other hand, Sudden movements are short in duration and tend to accelerate.

Free motions are unstoppable and tend to overshoot their goal points. To achieve an abandoned, uncontrolled quality, we use wrist bend values which give significant inward and outward bends. Since end-effector interpolation produces movements that are too rigid and controlled for Free movements, we use angular interpolation of shoulder and elbow angles. Squash and stretch of the body adds to the feeling of freedom and flexibility.

Bound motions tend to occur in slow motion, so we extend the number of frames between keypoints. Also, since Bound movements tend to be stiff and rigid, we keep the wrist in a flexed position.

4.3.2 Generating Effort Ranges and Combinations

Motion factor ranges are represented by a sliding value between -1 for the indulging extreme and $+1$ for the fighting extreme. To generate the range of each motion factor, we interpolate between the settings at the two extremes. For discrete variables (interpolation space and whether to display breath), we use the value of the nearest extreme. For instance, a Flow slider value of -0.3 generates an animation using angular interpolation because it is nearer to the Free extreme value than that of Bound. We note that this may lead to discontinuities in an animation generated in phrase mode when Space, Weight, or Flow cross zero at points that are *not* goal keypoints. Such occurrences are fairly uncommon, as it is rare to change from one Effort extreme to its opposite within a movement. Our Effort model is designed such that the posture computed at any goal keypoint is the same regardless of Effort settings (in other words, editing of the movements based on Effort settings occurs only *between* goal keypoints); thus, discontinuities do not occur at zero crossings at goal keypoints.

In general, combinations of Effort Elements are achieved in a straightforward manner. The magnitude of an Effort Element is used to weight its contribution for a parameter setting. If more than one Effort Element contributes to a parameter setting (such as for parameters *squashmag* and *wbmag*), we take the maximum value of the weighted contributions. Several parameters are tweaked when combining Effort Elements from different motion factors; the specific cases of these are described below.

Finally, we express our Effort model as a set of equations. Let the variables ind, dir, lgt, str, sus, sud, fre, and bnd represent the magnitudes for Indirect, Direct, Light, Strong, Sustained, Sudden, Free, and Bound, respectively. Each of these variables is in the range $[0,1]$. Variables within the same motion factor are related as such: if one Effort Element variable is positive, then its opposing Effort Element variable is zero. To tweak parameters for combined Effort settings, we use the function f, which is defined as follows:

$$f(a,b) = \begin{cases} a & a <= b \\ b & a > b \end{cases} \tag{4.10}$$

Our model for translating Effort into low-level movement parameters is given by the following equations:

$$Tval = (-1 * ind + 1 * f(fre, ind)) + dir \tag{4.11}$$

$$t_i = 0.5 + 0.4 * max(str, sud) - 0.4 * max(lgt, sus) + 0.8 * f(bnd, lgt) \tag{4.12}$$

$$v_0 = 0.1 * str - max(0.06 * f(sus, str), 0.1 * f(fre, str)) \tag{4.13}$$

$$v_1 = max(0.03 * max(lgt, sus), 0.2 * fre - 0.1 * f(ind, fre)) \tag{4.14}$$

$$
\begin{aligned}
texp &= 1 + 2 * sud + (0.2 * f(str, sud) - 0.6 * f(fre, sud)) \\
&\quad -0.2 * max(str, 0.5 * (dir + sus)) - 0.4 * fre - 0.1 * f(ind, fre)
\end{aligned} \tag{4.15}
$$

$$squashmag = max(0.20 * lgt, 0.15 * fre) \tag{4.16}$$

$$wbmag = max(0.6 * ind, 0.5 * lgt, 1.0 * fre) \tag{4.17}$$

$$wxmag = -0.3 * lgt + (0.2 * fre - 0.8 * f(str, fre)) \tag{4.18}$$

$$wtmag = 0.4 * ind \tag{4.19}$$

$$wfmag = 2 * ind \tag{4.20}$$

$$etmag = 0.4 * ind \tag{4.21}$$

$$efmag = 2 * ind \tag{4.22}$$

$$dmag = 0.4 * ind \tag{4.23}$$

$$awmag = 0.25 * str \tag{4.24}$$

When Effort Elements from different motion factors are combined, characteristics of one Effort Element may mask or conflict with characteristics of another. We defined our Effort model to adjust for these situations. When a movement is both Indirect and Free, we reduce the path curvature ($Tval$) to avoid overly uncontrolled movements. When a movement is Free and Strong, we reduce the wrist extension ($wxmag$) so the wrist bends are more inward, giving a greater impression of strength. By default, Light movements are defined to decelerate (t_i value of 0.1); while Bound movements use a standard ease-in/ease-out velocity curve (t_i value of 0.5). Since the primary characteristic of Bound movements is their evenness, movements that are both Light and Bound use a standard ease-in/ease-out velocity curve. Strong movements show anticipation (v_0 value of 0.1); since this contradicts certain characteristics of Sustained and Free elements, we reduce this value when Strong movements are combined with Sustained or Free. We reduce the overshoot (v_1) value for Free and Indirect movements to avoid uncontrollably wild movements. Sudden movements use a large $texp$ value to create a magnified acceleration. The $(0.2 * f(str, sud) - 0.6 * f(fre, sud))$ factor in the time exponent parameter ($texp$) calculation offsets the factors that represent the individual contributions for Strong and

Free. Strong movements use a slightly decreased *texp* value, as does the combination of Direct and Sustained. Free also uses a decreased *texp* value, which is increased slightly when displayed in combination with Indirect.

Our Effort-based motion control paradigm works directly with end-effector specified movements; however, it could be used with keyframed, motion captured, and procedurally generated movements as well. End-effector specification easily facilitates interpolation in end-effector, joint angle, and elbow position space. With motions specified using other methods, one could select keypoints in the movement, compute their corresponding end-effector positions, and use that as input into an Effort-based system for qualitative editing. Also, our Effort model is not specific to a particular character, nor to characters with particular arm and body dimensions, although we do assume human-like arm structures.

The next chapter discusses an implementation of our Effort model and shows how the methods and equations defined in this chapter are applied to generate animations in real-time.

Chapter 5

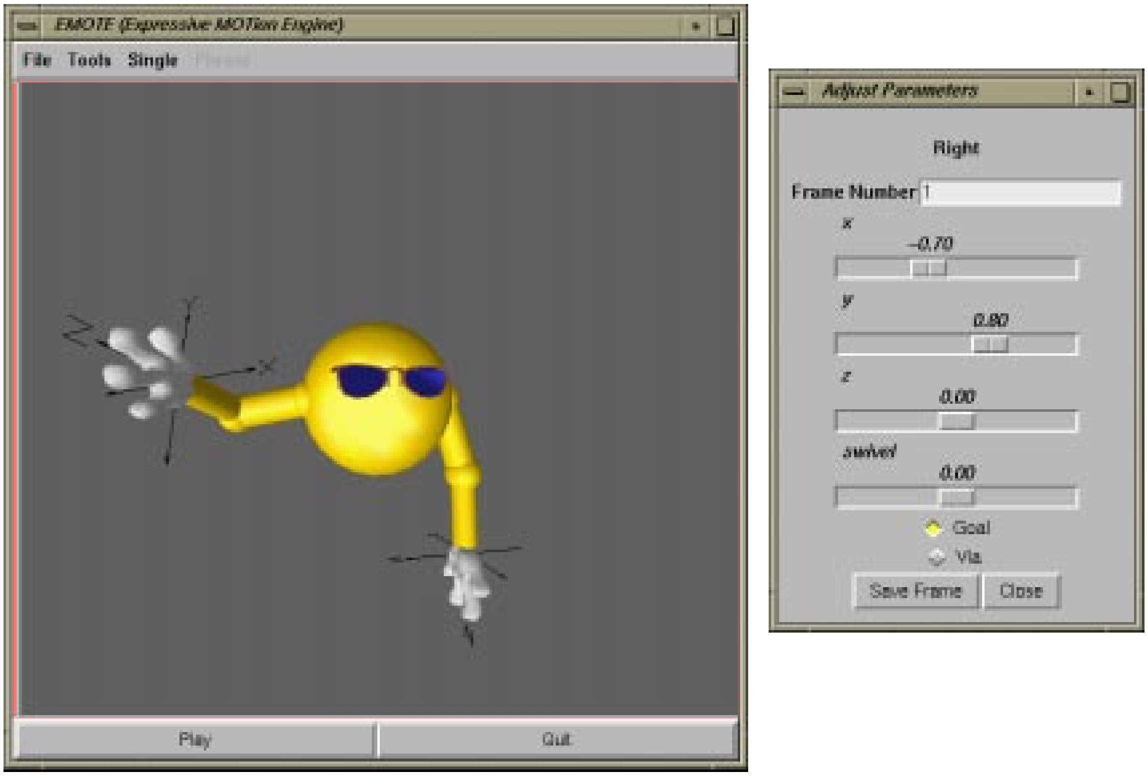
Implementation: EMOTE

We have implemented our motion control scheme in an animation module called EMOTE (Expressive MOTion Engine). EMOTE uses inverse kinematics, where spatial movement requirements are specified through end-effector positions. This chapter describes the generation of animations in EMOTE, the arm model, use of the Effort model, and example animations.

5.1 User Interaction

EMOTE users generate animations by (1) specifying general movement sequences as a series of keypoints, and then (2) interactively editing qualitative parameters until the desired animation is achieved.

EMOTE uses two types of keypoints: goal keypoints and via keypoints. Goal keypoints define a general movement path; the hand follows a path which stops at each goal keypoint. Via keypoints direct the motion between goal keypoints without pausing. For instance, a via keypoint might be used to generate a semi-circular path between two goal keypoints. EMOTE provides two methods for users to specify a series of keypoints. One method allows users to specify a keypoint by using sliders (Fig. 5.1) to interactively set the figure in the desired position. After specifying the frame number and selecting the keypoint type, the user can save the keypoint into the Key Editor (Fig. 5.2). The other method reads in an ascii text file containing the frame number, x, y, z position values, swivel angle, and



Frame Number 1

x −0.70

y 0.80

z 0.00

swivel 0.00

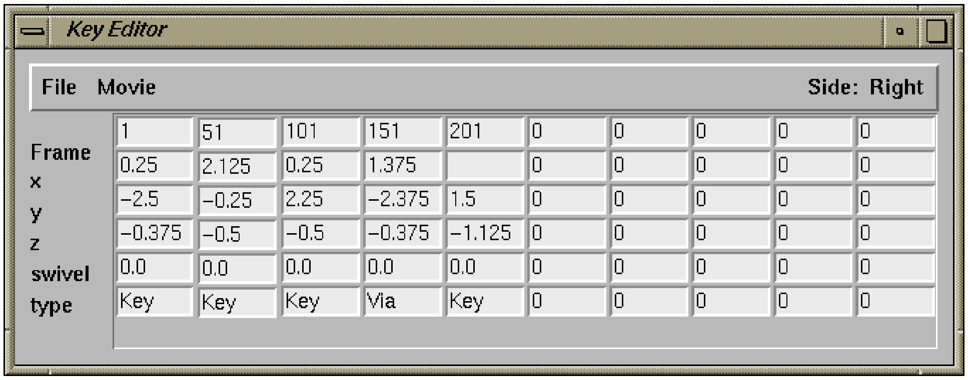
● Goal

○ Via

Figure 5.1: Interface for Adjusting End-Effector Positions

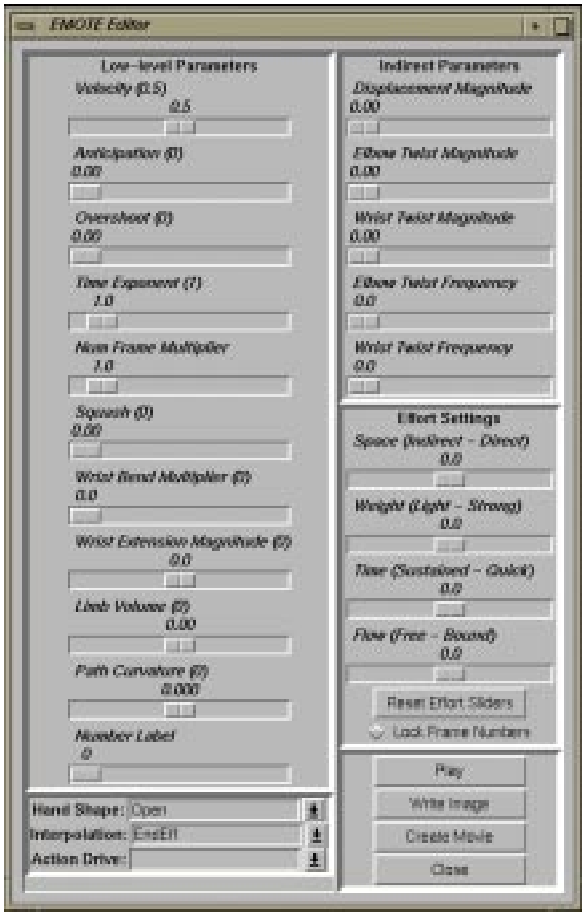
type of each keypoint.

EMOTE has two modes for specifying Effort settings: single mode and phrase mode. In single mode, the user sets Effort values using four sliders—Space, Weight, Time, and Flow—on the Effort Editor (Fig. 5.3). These automatically set a number of low-level parameters, which the user may also edit. Pressing the **Play** button generates an animation following the path specified by the end-effector keypoints and displaying the specified Effort qualities. In phrase mode, the user can specify changes in Effort over time. An Effort Graph Editor (Fig. 5.4) displays the Effort settings over an animation, allowing the user to interactively edit the points and play the corresponding animation. Both modes provide a radio button that allows the user to lock or unlock the frame numbers for keypoints. Locking the frame numbers prevents the Effort settings from changing the frame numbers of keypoints.



Frame	1	51	101	151	201	0	0	0	0	0
x	0.25	2.125	0.25	1.375		0	0	0	0	0
y	−2.5	−0.25	2.25	−2.375	1.5	0	0	0	0	0
z	−0.375	−0.5	−0.5	−0.375	−1.125	0	0	0	0	0
swivel	0.0	0.0	0.0	0.0	0.0	0	0	0	0	0
type	Key	Key	Key	Via	Key	0	0	0	0	0

Figure 5.2: Key Editor



Velocity (0.5) 0.5
Anticipation (0) 0.00
Overshoot (0) 0.00
Time Exponent (1) 1.0
Num Frame Multiplier 1.0
Squash (0) 0.00
Wrist Bend Multiplier (0) 0.0
Wrist Extension Magnitude (0) 0.0
Limb Volume (0) 0.00
Path Curvature (0) 0.000
Number Label 0

Hand Shape: Open
Interpolation: EndEff
Action Drive:

Displacement Magnitude 0.00
Elbow Twist Magnitude 0.00
Wrist Twist Magnitude 0.00
Elbow Twist Frequency 0.0
Wrist Twist Frequency 0.0

Space (Indirect – Direct) 0.0
Weight (Light – Strong) 0.0
Time (Sustained – Quick) 0.0
Flow (Free – Bound) 0.0

Figure 5.3: Effort Editor

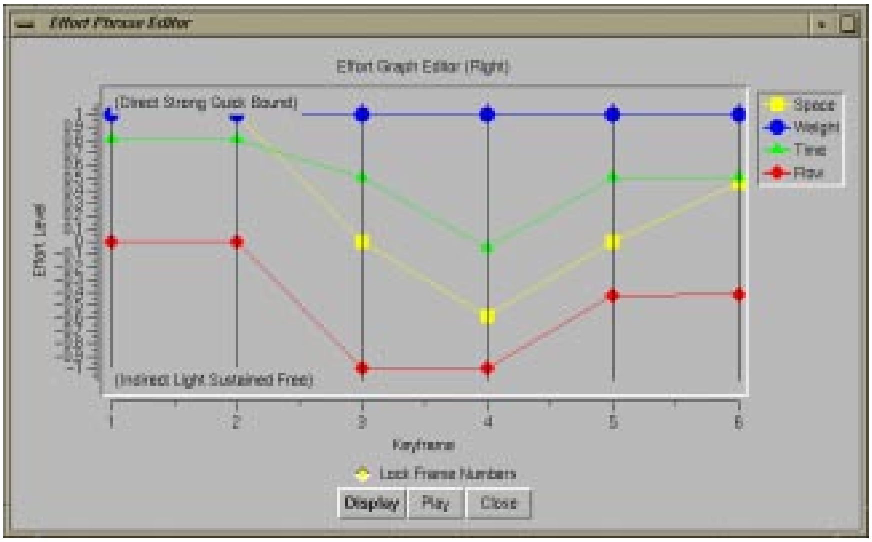


Figure 5.4: Effort Graph Editor

5.2 Arm Model

EMOTE models the arm as a kinematic chain with two spherical joints, the shoulder and wrist, connected by a flexion joint at the elbow. The shoulder and wrist have three degrees-of-freedom (DOF), which represent rotations about the x, y, and z axes. The elbow has 1 DOF, rotation about a single axis. An analytical inverse kinematics algorithm computes the shoulder and elbow rotations, given a goal specified by three-dimensional position coordinates and an elbow swivel angle [76]. The base of the wrist acts as the end-effector indicating the goal position. Since the determination of arm posture given 3D position is under-specified, Tolani uses the *swivel angle*, which specifies the location along the circular arc swept out by the elbow swiveling around the axis between the shoulder and the wrist ($\hat{n}$) and lying on the plane normal to $\hat{n}$ (Fig. 5.5). Wrist rotations are determined according to Effort settings (as described in Chapter 4).

For demonstration, we have created the character Mo. Mo has a spherical body and articulated arms. The current version of Mo does not have an articulated hand, although EMOTE provides a selection of hand shapes: open, fist, pointing, closed, fingers together with thumb out, cupped, and claw.

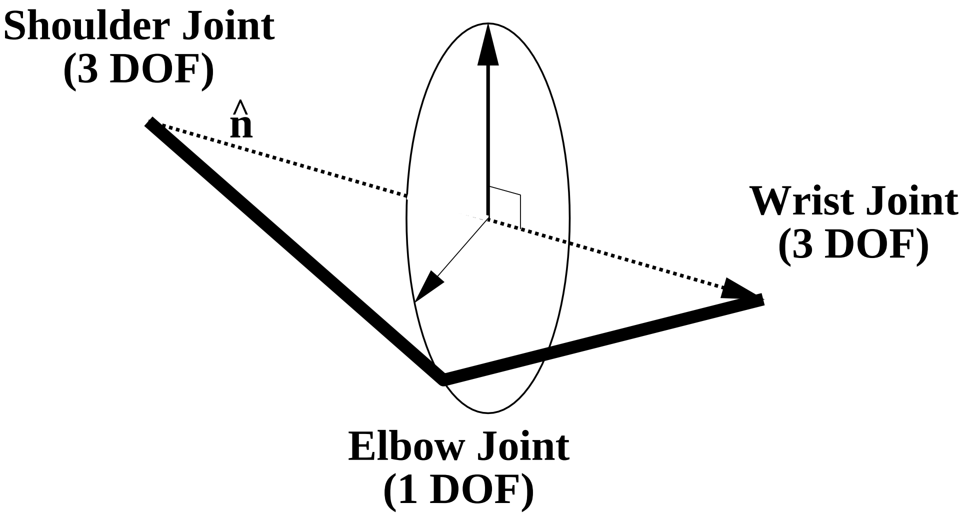


Figure 5.5: Arm Model

5.3 Method for Using Effort Model

When a user requests an animation, we plug the Effort settings into Equations 4.11 to 4.24 to compute the low-level movement parameters. For each frame, we use the method described in Section 4.2.2 to compute end-effector positions for the given frame. Then, we call the IK module to compute and set the matrices defining the arm postures. Mo is re-drawn using the computed arm matrices and computation begins on the next frame.

5.4 Examples

5.4.1 Individual Effort Elements

To demonstrate the utility of our motion control scheme and the wide range of possible movements generated, we include thumbnail images from a series of animations that were generated from the same five goal keypoints (Fig. 5.6). We generated the animations using the default settings for each individual Effort Element. Opposing Effort Elements for each motion factor are shown together for easy comparison. Frames are displayed in order from right to left, top to bottom. The Space and Weight animations show every fifth frame of the animation, while the Time and Flow animations show every tenth frame.

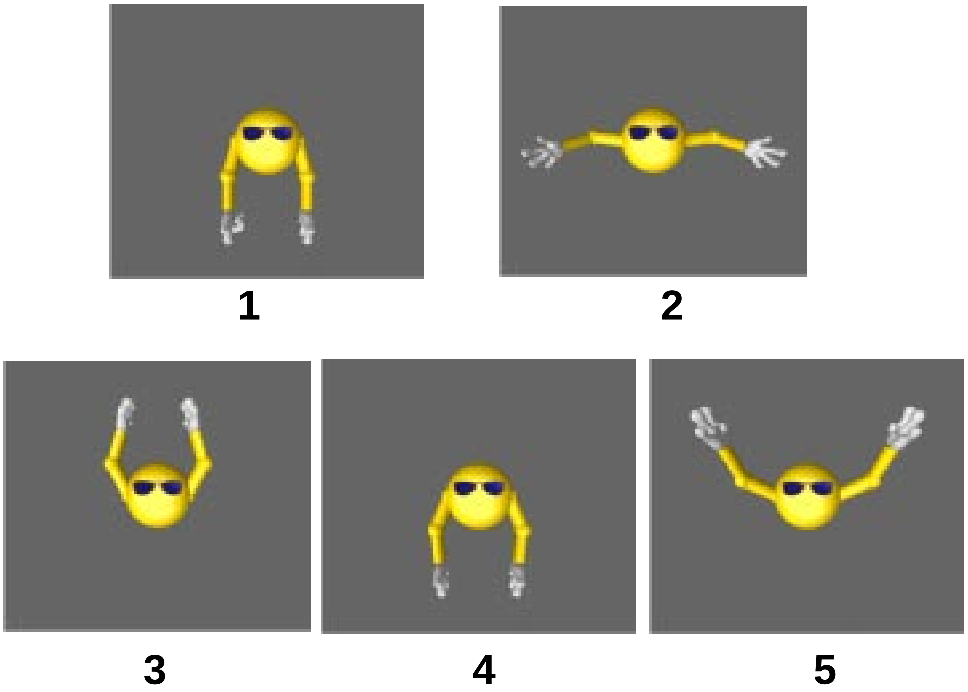


Figure 5.6: End-Effector Keys for an Example Movement Sequence

Fig. 5.7 shows frames from the animations of Indirect and Direct. The Direct animation is more focused on the path of the hands, displaying stiffer, more angular arm angles than the rounded, all-encompassing arm postures of the Indirect animation (as seen in frames A7, C3, and E7). The Indirect animation also displays more arm twists and a flexible wrist; whereas Mo maintains a stiff, rigid wrist position in the Direct animation (A7, C1, C5, F2).

Fig. 5.8 shows several notable differences between the Strong and Light animations. The Strong animation displays marked acceleration, apparent by the large movement changes observed just before each keypoint (A6-B3, C1-C6, D4-E2,F1-F7), while the beginning of each movement remains relatively unchanged (A2-A5, B4-B7, C7-D3, E3-E7). In contrast, the Light animation uses a decelerating velocity curve, which starts with significant amounts of movements (A2-A6,B6-C2, D2-D5, E6-F1) and ends with slight movements (A7-B5, C3-D1, D6-E5, F2-F7). To emphasize the impact (or lack thereof) of the movements, we selected a fist shape for the Strong movement and an open hand for the Light movement. The Strong animation has a stiff, but flexed wrist, while the Light

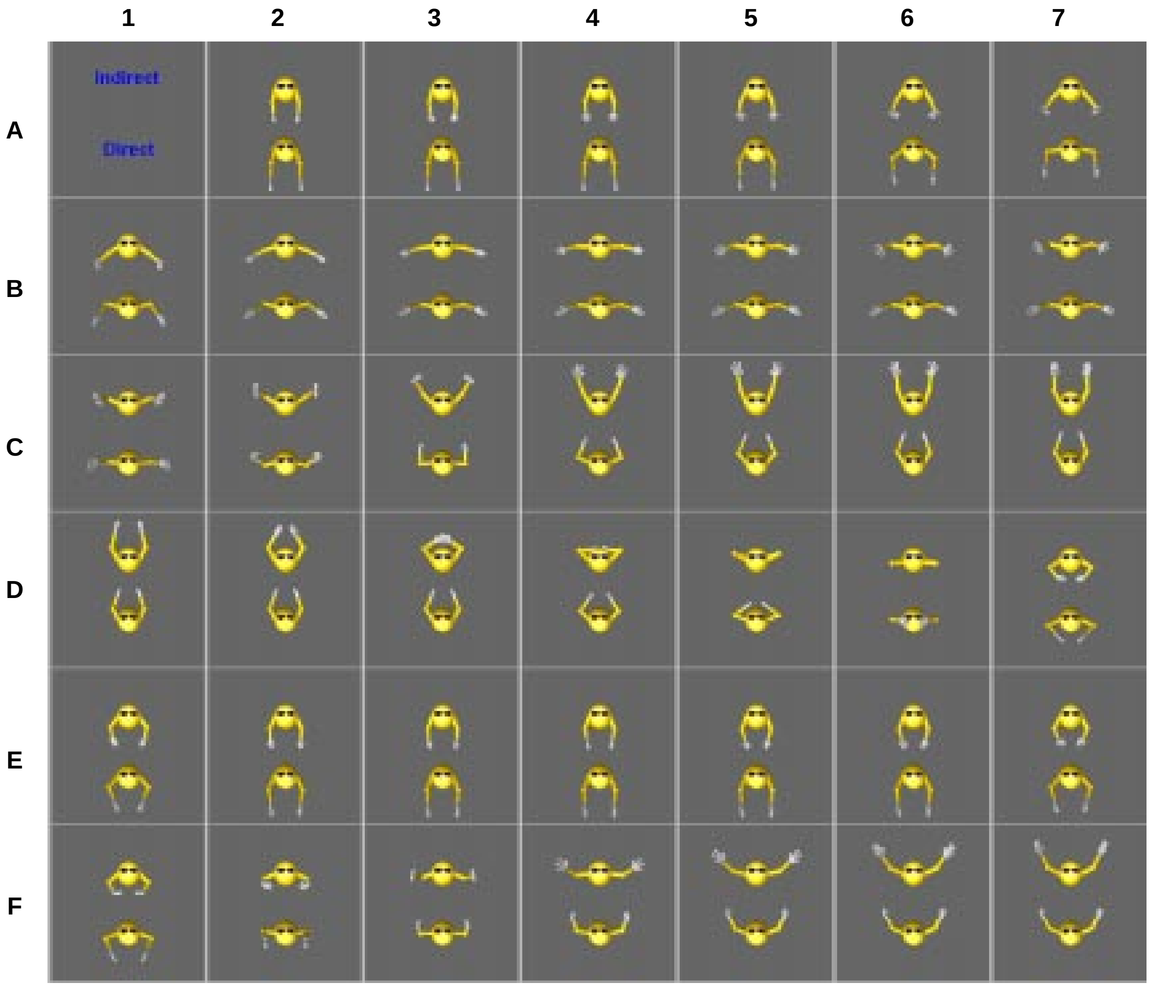


Figure 5.7: Every Fifth Frame from Animation of Indirect and Direct Efforts (ordered left to right, top to bottom)

animation uses a significant amount of wrist bend (A3-A7, B6-C2, E6-F2). Although the Strong movements shows anticipation and the Light movements overshoot the keypoints, these effects are subtle and difficult to observe from the selected thumbnail images. The Light animation displays squash and stretch of the torso, while the Strong animation shows a slight torso contraction (breath) and increased limb volume (biceps flexion) before reaching each keypoint.

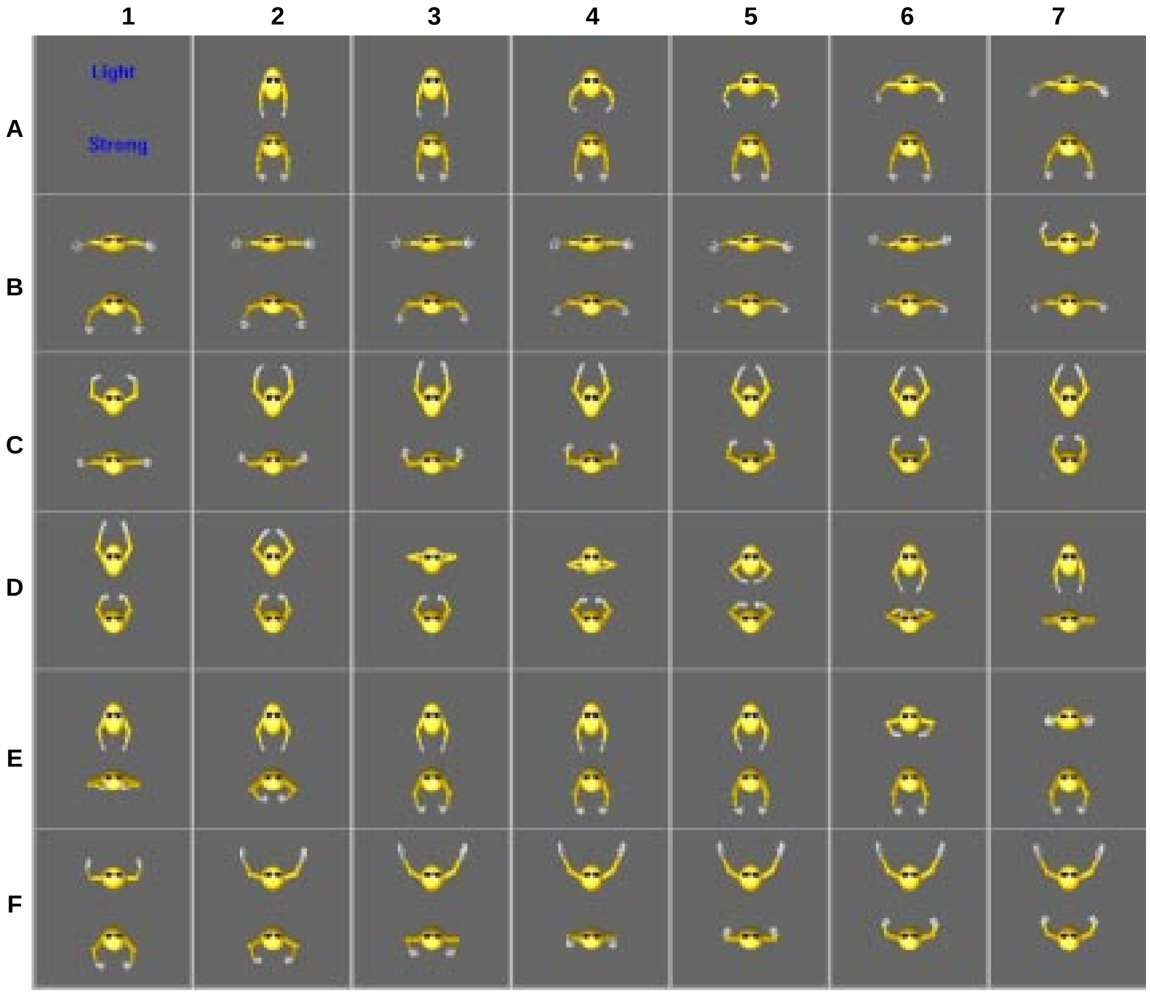


Figure 5.8: Every Fifth Frame from Animation of Light and Strong Efforts (ordered left to right, top to bottom)

The most notable difference between the Sudden and Sustained animations (Fig. 5.9) is the difference in duration. The Sudden animation is completed by frame B1, while the Sustained animation continues through to frame I7. The Sudden animation displays marked acceleration, but this element is not evident in the figure since only every 10th frame is displayed. The Sustained movements are decelerating, as displayed by the large movements (A2-B2, C5-C7, E7-F4, G7-H5) followed by the insignificant movements (B3-C4, D1-E6, F5-G6, H6-I7) before each keypoint.

The duration of the Bound animation is significantly longer than that of the Free animation (Fig. 5.10). The Free animation is complete by frame C6, while the Bound animation lasts until I7. Free movements show a significant amount of wrist bend (A4, B6, C4), while Bound movements maintain a flexed wrist. The Free movement also displays squash and stretch of the torso.

5.4.2 Gestures Accompanying Speech

We have found that our motion control scheme is useful for generating the hand gestures accompanying speech, because the Effort settings for the animation can be set to match the intent of the speaker. For instance, a powerful denial is accompanied by Strong and Direct hand gestures (Fig. 5.11). The speech that accompanies the animation states, "I did not have sexual relations with that woman." Fig 5.12 displays every fourth frame from an animation of Mo saying, "Hey, look! I made it. I made it." The animation shows a gleeful exclamation that ends with Mo throwing up his arms in a Free movement.

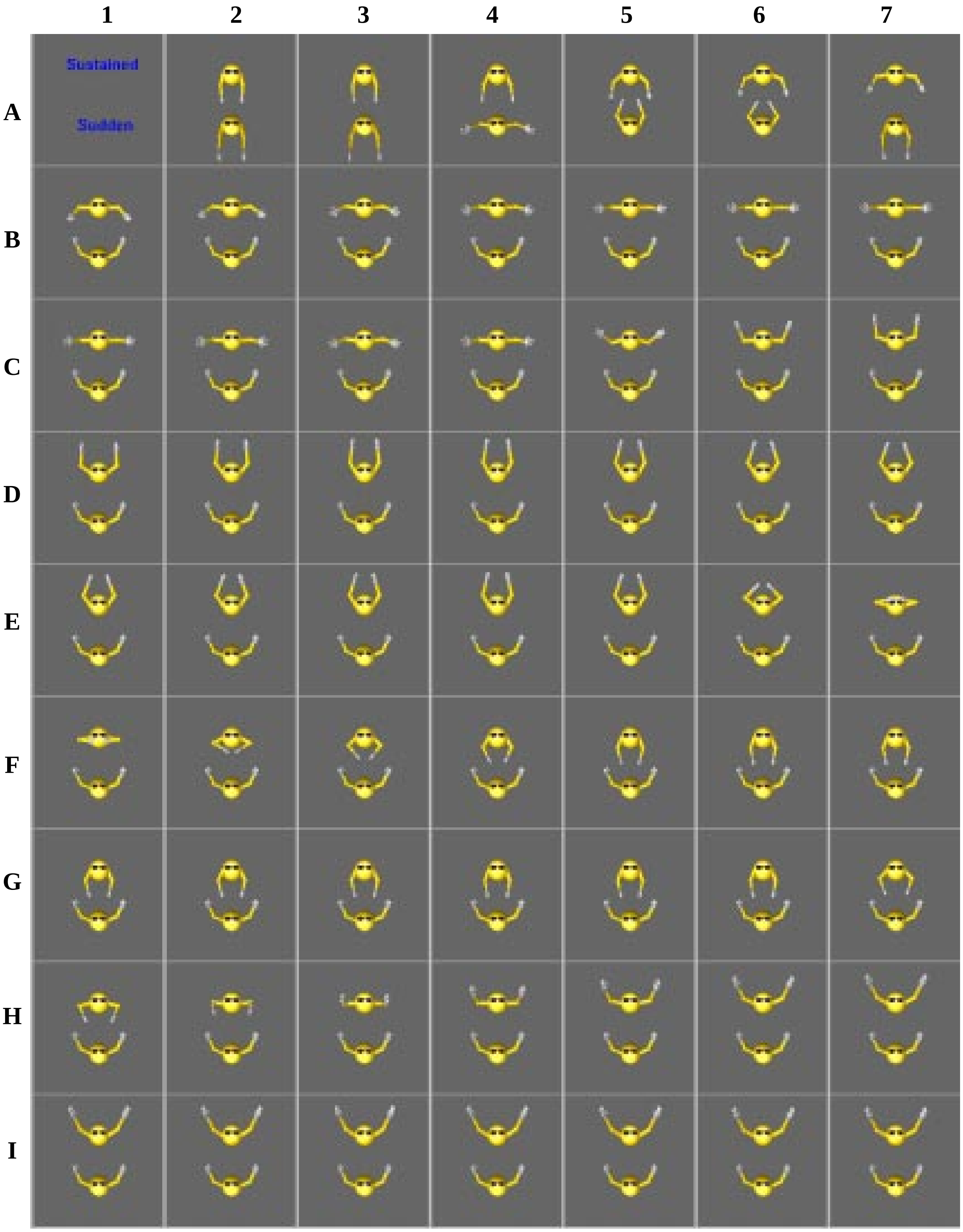


Figure 5.9: Every Tenth Frame from Animation of Sustained and Sudden Efforts (ordered left to right, top to bottom)

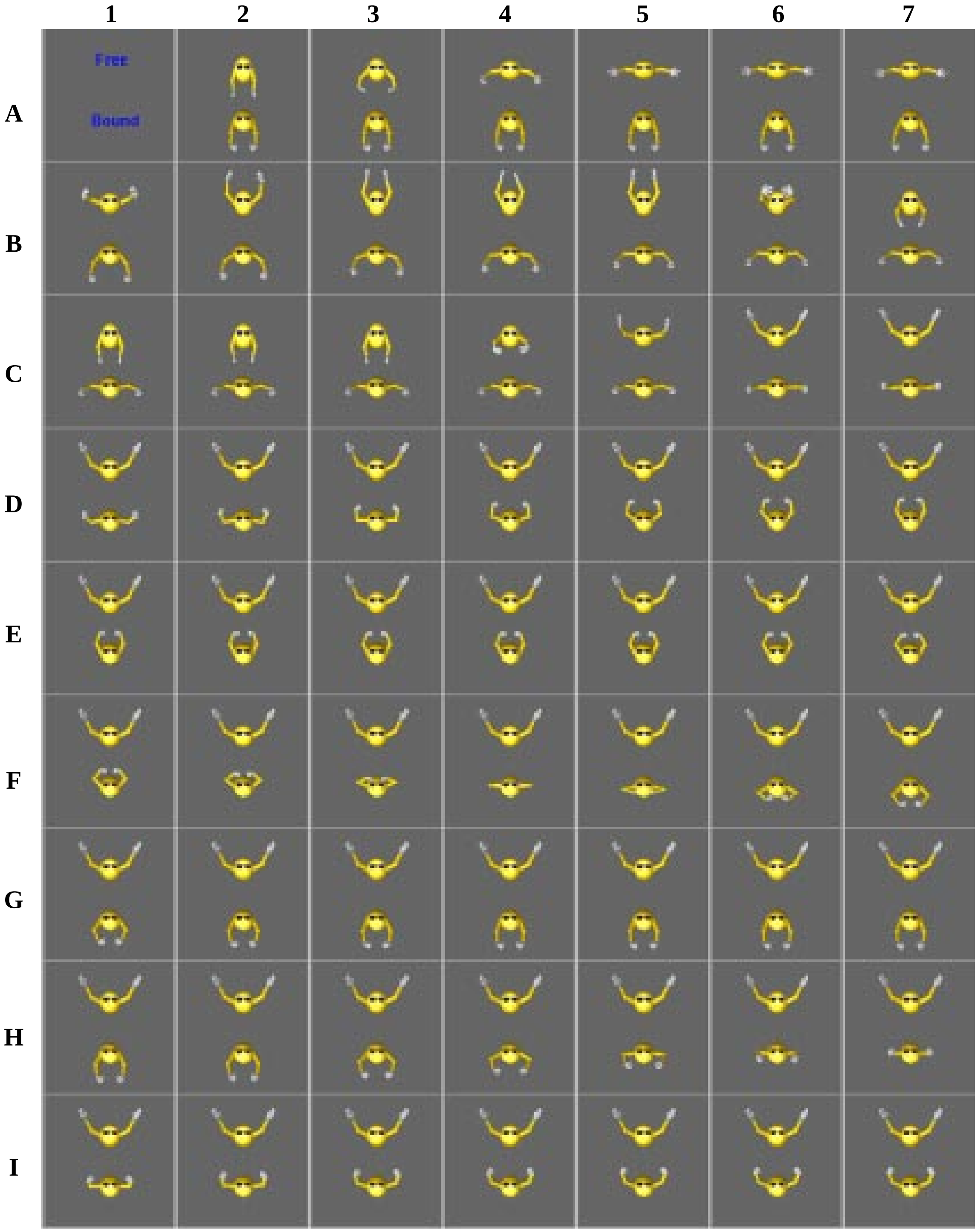


Figure 5.10: Every Tenth Frame from Animation of Free and Bound Efforts (ordered left to right, top to bottom)

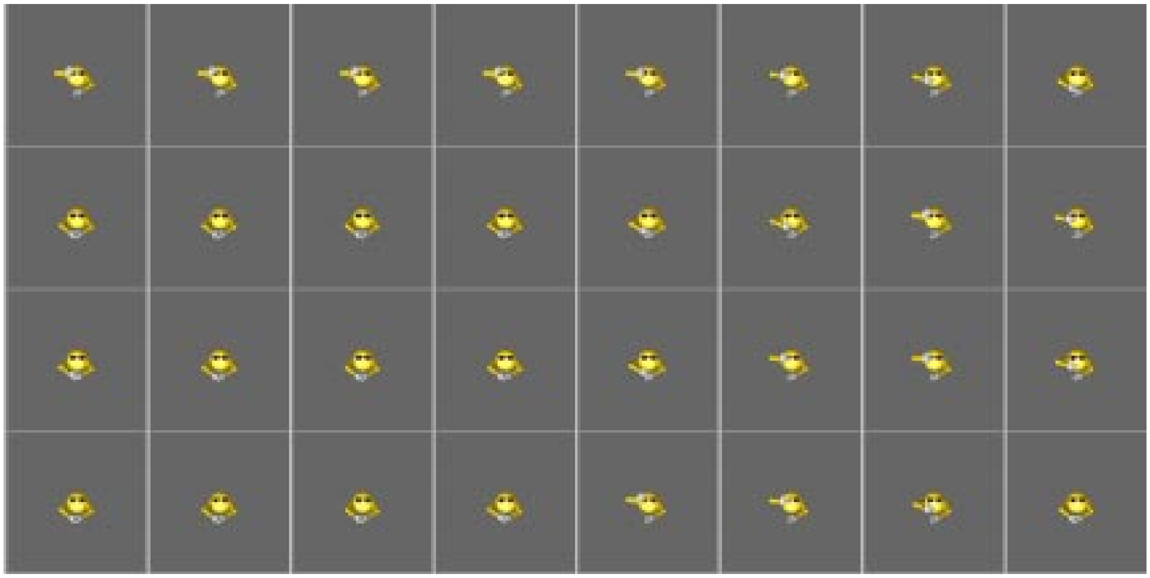


Figure 5.11: Every Fourth Frame from Animation of a Denial (ordered left to right, top to bottom)

Figure 5.12: Every Fourth Frame from Animation of a Gleeful Exclamation (ordered left to right, top to bottom)

Chapter 6

Conclusions

This chapter discusses an evaluation of our Effort model and the usability of EMOTE. We conclude with extensions and contributions of the work.

6.1 Evaluation

6.1.1 Effort Model Evaluation

To evaluate our motion control scheme, we used EMOTE to create a 16-minute video containing a series of animation segments. The animation segments displayed randomly selected Effort Elements, both individual and in combination. The video was divided into two parts. The first part showed a neutral movement (no Effort Elements present) before displaying the animation segment of the movement to be coded. This part consisted of 16 short animations (2 keypoints), followed by 16 long animation segments (5 keypoints). The second part showed only animations of the movement to be coded, and consisted of 30 long animation segments (5 keypoints). The video was given to 3 CMAs and our consultant CMA. They were asked to view the video once to get a feel for the movements of the character Mo, and then to watch it again while marking a coding sheet. For each animation segment, they were asked to "mark the *main, overall* Effort(s) present in the segment" on a chart as follows:

Indirect	-1	0	1	Direct
Light	-1	0	1	Strong
Sustained	-1	0	1	Quick
Free	-1	0	1	Bound.

The -1 value represents the indulging extreme, while the $+1$ value represents the fighting extreme. The 0 value indicates neutrality along the given motion factor.

Table 6.1 summarizes the overall results of the evaluation of our Effort model. The first row indicates the percentage of correct responses—where the CMA either marked the Effort that we were trying to display in the animation or marked neutral when we were trying to display neutrality along a given motion factor. The second row indicates the percentage of neutral responses—where the CMA marked neutral when we were trying to display an Effort or where the CMA marked an Effort when we were trying to display neutral along a given motion factor range. The third row indicates the percentage of opposite responses—where the CMA marked the Effort opposite from the one we were trying to portray. Fortunately, the correct responses display significantly greater than chance agreement, while the percentage of opposite responses is low. The low but significant percentage of neutral responses is partially attributed to the fact that most of the animation segments on our video showed *combinations* of the Effort Elements—thus, a more prominent Effort may have masked other displayed Effort Elements.

There are several reasons that we did not achieve perfect agreement between the Efforts we were trying to portray and the qualities marked by the CMA observers. First of all, many of the manifestations of Effort are subtle and ephemeral; notators watching the same live human performers must work together through repeated viewings and discussions in order to achieve inter-observer agreement. Even CMAs notating video several months after their initial observation don't agree completely with their previous notations [26]. Another factor affecting the scores is that the manifestations of Effort on our character Mo affected only his arm movements with limited limb volume and torso support; with a human performer, facial expression, eye gaze, muscle flexion, environmental context, sounds and spoken words all give additional clues to the Effort being portrayed. Further, observing and analyzing a stylized cartoon character was a novel experience for the CMA

	Consultant	CMA 1	CMA 2	CMA3
Correct	76.6	55.6	53.2	60.1
Neutral	22.6	38.7	39.1	37.1
Opposite	0.81	5.6	7.7	2.8

Table 6.1: Overall Percentages for Effort Model Evaluation

observers. Also, we asked the CMAs to make judgments based on only two viewings in one sitting, rather than allowing for repeated, careful analyses over a period of time. The evaluations generally represent only their initial impressions. Finally, even though CMAs are extensively trained in observing the qualitative aspects of movements, the subjective nature of Effort inherently leads to somewhat personalized notions of Effort.

The LIMS Reliability Project is the most comprehensive study of observer agreement on LMA thus far [26]. The LIMS project involved CMAs watching 45-second videotape segments of dance solos and segments of conversation with one speaker off camera. Observers were asked to press keys on a computer when they observed certain features; the computer recorded the time (in seconds) of the key presses. Effort Elements of the same motion factor were observed simultaneously. For various reasons, it is difficult to conclusively assess the results of the project. The low number of occurrences of certain Effort Elements (Light, Sustained, and Indirect) on the videotape segments meant that statistical analysis was infeasible for those features. Also, assessments of Flow were unreliable because observers were not used to specifying instances of Free and Bound; rather they view Flow as a continuous fluctuation between the two extremes. In general terms, agreement for Strong and Light was "relatively high, if spotty". Observers "could agree on the perception of [S]udden" for the dance segments, though agreement was "low but notable" on the conversation segments. Direct gave "consistently positive if sometimes low" agreement. The results of the project suggest that Indirect, Light, Bound and Free "require more rigorous training, practice and, in some cases, more dynamic videotape examples" in order to achieve high levels of agreement. Because the nature of their experiments were fundamentally different from ours, it was not possible to quantitatively compare their results with our results. Our experiments had "right" answers, where

	Consultant	CMA 1	CMA 2	CMA3	Average (non-consultant CMAs)
Indirect	84.6	92.3	76.9	69.2	79.5
Direct	60.0	80.0	40.0	26.7	48.9
Light	90.5	38.1	23.8	42.9	34.9
Strong	82.3	5.9	29.4	0.0	11.8
Sustained	79.0	47.4	100	42.1	63.2
Sudden	100	100	100	100	100
Free	76.2	57.1	33.3	38.1	42.9
Bound	75.0	25.0	75.0	37.5	45.8

Table 6.2: Percentage Correct for Individual Effort Elements

marking the Effort we were trying to display constituted a correct response. Their experiment had no right answer, and instead measured the level of agreement between two viewers. Further, their analysis measured agreement in the total number of Effort Elements viewed in a segment, rather than the overall Effort displayed.

Table 6.2 shows the percentage of correct responses in our Effort Model Evaluation for each Effort Element. The results are interesting compared to the results from the LIMS Reliability Project, which gave high levels of inter-observer agreement for Strong, Direct, Sudden [26]. Our results indicate a high percentage of correct responses for Sudden, Indirect, and Sustained, while Direct, Bound, and Free gave moderate results. The percentages for Strong and Light are surprisingly low compared to the performance of our consultant on those qualities. There are several possible reasons for this mismatch. First of all, our consultant is familiar with the subtle cues we used to portray Weight characteristics, and thus, could probably more readily identify the Weight Efforts. Also, Strong qualities are often revealed by body and limb tension, and the throwing of one's weight into a movement—qualities that are difficult to display in our simple Mo character. Light qualities are often revealed by delicate hand and finger gestures—which are limited by EMOTE's requirement that movements must use a preset hand shape.

More details on the LIMS Reliability Project along with several other articles on observer agreement using Laban-based notations are provided in [25]. Overall, we believe that our Effort model gives more than adequate results given the subjectivity of Effort, the nature (and novelty) of Effort analysis in our experiment, and the results of previous studies on observer agreement.

6.1.2 User Evaluations

The EMOTE interface was not the focus of this work; however, we performed some preliminary user evaluations to demonstrate the ease of generating expressive movements using our Effort model. We had two users (User A and User B) evaluate EMOTE and compare it to a commercial animation product *3D Studio MAX*®. User A had no experience with EMOTE or LMA, but did have some experience with various commercial animation packages. User B had some experience in both. We had both users select two Basic Effort Actions[1]: float, punch, glide, slash, wring, dab, flick, or press. Each user was told to animate the selected actions using EMOTE and *3DS MAX*, filling out a brief questionnaire comparing the two. Both users took less time generating their animations while using EMOTE (10-20 minutes for User A, and 5-10 minutes for User B) compared to *3DS MAX* (20-30 minutes for both). Also, both rated the quality of their animations as the same or better using EMOTE. We also asked users what percentage of time they spent setting up the animation and what percentage of time they spent tweaking to get the desired animation. User A spent $40 - 50\%$ of the time tweaking the animation when using EMOTE and $70 - 80\%$ of the time tweaking the animation when using *3DS MAX*. The user explained that once he "got a rough approximation in EMOTE, [he] was able to get the motion to look great with a small amount of time". He did find that setting up an animation in *3DS MAX* was easier because it provided a more extensive set of tools. We note that this portion of the animation task was not the one we were seeking to improve. User B spent 85% of the time tweaking animations in EMOTE and only 50% tweaking animations in *3D Studio MAX*. He explained that he was "trying too hard in [EMOTE] to get the gestures just right" and appreciated that one doesn't have to worry as much about timing issues in EMOTE. Overall, we believe the results suggest that our motion control paradigm simplifies and speeds up the task of generating expressive movements, and provides a capability not available in current animation packages.

---

[1] The Basic Effort Actions were defined by Laban as the eight combinations of Effort Elements for Space, Weight, and Time. He used a textual "action word" to describe each.

6.2 Extensions

In order to extend our motion control paradigm to handle leg movements, one would need to account for some physical factors that don't affect the arms, such as balance and bearing weight. On the other hand, the legs are used significantly less than the arms in human expression. The trajectory definition and timing control aspects of our Effort model are directly applicable to leg movements. Some of the flourishes (namely, torso support and limb volume) would also be manifested in the legs depending on Effort settings, although the equations defining them may have to be altered slightly. The other flourishes we defined—wrist bend and arm twist—are specific to the arms, although the legs would probably have similar, and possibly additional, flourishes (i.e.: ankle bend and leg twist) that would need to be defined.

We believe that an important area of the body that has essentially been ignored in animating human-like characters is the torso. The torso plays an important role in shaping. For instance, a tired or discouraged person might reveal these inner turmoils with a concave shape and slouched shoulders. A proud, boastful person might strut with chest out and head high. Also, characteristics of one's breathing provide a number of clues into the situation or one's attitude towards it. Someone who is expending a lot of energy with powerful gestures probably feels strongly about the ideas they are conveying. A person who is fearful or anxious will likely display quickened breathing. We have simulated some torso changes using squash and stretch on Mo's body, but a more detailed model is needed in order to capture all the subtleties of human torso shaping. Francois Delsarte (1811-1871) dedicated his life to discovering the *laws of expression* [71]. One portion of his work is presented as a set of charts showing basic positions of various body parts and their respective meanings. Zhao and Badler implemented a system that uses Delsarte's charts for the head, arm, and hand as a baseline for gestures accompanying speech [90]. The first step towards expressive torso movements would be to extend the work of Zhao and Badler to include Delarte's interpretations of torso postures and movements.

We suggest a plan for utilizing an Effort-based motion control scheme in a behavioral animation system. Using natural language for the high-level control (i.e.: in the form of speech text or a storyboard script), one could develop a translator that would take adverbs and convert them into Effort settings. "Carefully" might translate into Light and slightly Sustained (a Weight value of -1.0 and Time value of -0.3); "haphazardly" might translate into Indirect and somewhat Free (a Space value of -1.0 and Flow value of -0.8). Then, one could tweak these settings based on a character's personality and/or mood. An introverted person might tend towards Bound and Indirect movements, which could be achieved by adding a small value to the Flow and Space settings. Finally, one could add a small degree of randomness to the Effort settings to account for individual differences among characters with matching personalities and moods.

6.3 Contributions

We have defined the quantitative structures necessary to model qualitative aspects of movement. Using these structures, we have built an empirical model of Effort. We introduce a novel motion control paradigm that uses our Effort model to facilitate the specification of human expressive movements and provides real-time, interactive control. While skilled animators can achieve life-like, expressive characters by controlling the individual lowest level degrees-of-freedom via keyframing or inverse kinematics methods, Effort provides a more systematic and meaningful way of describing qualitative aspects of movement. Varying the intensities of the Effort Elements and combining different elements produce a rich language for expressive movements. A system using Effort-based motion control could provide low-level control by outputting joint angle (or other motion parameter) curves for editing with traditional methods. More importantly, by allowing users to customize basic movements with general terms, such a system supports individualized expression as well as enabling users to create high-level movement controllers. This paradigm is an essential step towards a system where a user creates character by specifying personality, attitudes, and intentions, which in turn may eventually lead to the automatic generation of appropriate movements from speech text, a storyboard script, or a behavioral simulation.

Bibliography

[1] Kenji Amaya, Armin Bruderlin, and Tom Calvert. Emotion from motion. In Wayne A. Davis and Richard Bartels, editors, *Graphics Interface '96*, pages 222–229. Canadian Information Processing Society, Canadian Human-Computer Communications Society, May 1996.

[2] N. Badler, B. Webber, J. R. Clarke, D. Chi, M. Hollick, N. Foster, E. Kokkevis, D. Metaxas, O. Ogunyemi, J. Kaye, and R. Bindiganavale. Medisim: Simulated medical corpsmen and casualties for medical forces planning and training. In *The National Forum: Military Telemedicine On-Line Today. Research, Practice and Opportunities.* IEEE Computer Society Press, 1995.

[3] N. I. Badler and S. W. Smoliar. Digital representation of human movement. *ACM Computing Surveys*, 11:19–38, March 1979.

[4] Norman I. Badler. A computational alternative to effort notation. In Judith A. Gray, editor, *Dance Technology: Current Applications and Future Trends.* National Dance Association, Reston, VA, 1989.

[5] Norman I. Badler, Cary B. Phillips, and Bonnie Lynn Webber. *Simulating Humans: Computer Graphics Animation and Control.* Oxford University Press, New York, 1993.

[6] Irmgard Bartenieff and Martha Davis. Effort-shape analysis of movement: The unity of expression and function. In Martha Davis, editor, *Research Approaches to Movement and Personality.* Arno Press Inc., New York, 1972.

[7] Irmgard Bartenieff and Dori Lewis. *Body Movement: Coping with the Environment.* Gordon and Breach Science Publishers, New York, 1980.

[8] Ronen Barzel and Alan H. Barr. A modeling system based on dynamic constraints. In *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 179–188, August 1988.

[9] R. Benesh and J. Benesh. *An Introduction to Benesh Dance Notation.* A. and C. Black, London, 1956.

[10] Rama Bindiganavale and Norman Badler. Motion abstraction and mapping with spatial constraints. In *Modelling and Motion Capture Techniques for Virtual Environments, International Workshop (CAPTECH'98)*, pages 70–82. Springer-Verlag, 1998.

[11] Leslie Bishko. Relationships between laban movement analysis and computer animation. In *Dance and Technology I: Moving Toward the Future*, pages 1–9, Ohio, 1992. Fullhouse Publishing.

[12] James P. Bliss, Philip D. Tidwell, and Michasel A. Guest. The effectiveness of virtual reality for administering spatial navigation training to firefighters. *PRESENCE: Teleoperators and Virtual Environments*, 6(1):73–86, February 1997.

[13] Bruce M. Blumberg and Tinsley A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In *SIGGRAPH 95 Conference Proceedings*, pages 47–54, August 1995.

[14] Armin Bruderlin and Thomas W. Calvert. Goal-directed, dynamic animation of human walking. In *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 233–242, July 1989.

[15] Armin Bruderlin and Tom Calvert. Knowledge-driven, interactive animation of human running. In Wayne A. Davis and Richard Bartels, editors, *Graphics Interface '96*, pages 213–221. Canadian Information Processing Society, Canadian Human-Computer Communications Society, May 1996.

[16] Armin Bruderlin and Lance Williams. Motion signal processing. In *SIGGRAPH 95 Conference Proceedings*, pages 97–104, August 1995.

[17] Dance Notation Bureau. 31 West 21st Street, New York, NY, 10010; http://www.dancenotation.org.

[18] T. W. Calvert, J. Chapman, and A. Patla. Aspects of the kinematic simulation of human movement. *IEEE Computer Graphics and Applications*, 2:41–48, November 1982.

[19] Diane Chi, John Clarke, Bonnie Webber, and Norman Badler. Casualty modeling for real-time medical training. *PRESENCE: Teleoperators and Virtual Environments*, 5(4):359–366, December 1996.

[20] Diane Chi, Evangelos Kokkevis, Omolola Ogunyemi, Rama Bindiganavale, Mike Hollick, John Clarke, Bonnie Webber, and Norman Badler. Simulated casualties and medics for emergency training. In K.S. Morgan, H. Hoffman, D. Stredney, and S.J. Weghorst, editors, *Medicine Meets Virtual Reality*, volume 39, pages 486–494, 1997.

[21] Michael F. Cohen. Interactive spacetime control for animation. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 293–302, July 1992.

[22] Martha Davis. Effort-shape analysis: Evaluation of its logic and consistency and its systematic use in research. In Irmgard Bartenieff, Martha Davis, and Forrestine Paula, editors, *Four Adaptations of Effort Theory in Research and Teaching*, New York, 1970. Dance Notation Bureau, Inc.

[23] Martha Davis, editor. *Understanding Body Movement: An Annotated Bibliography.* Arno Press, New York, 1972.

[24] Martha Davis. Laban analysis of nonverbal communication. In Shirley Weitz, editor, *Nonverbal Communication: Readings with Commentary*, pages 182–206. Oxford University Press, New York, 2 edition, 1979.

[25] Martha Davis, editor. *Movement Studies: A Journal of the Laban/Bartenieff Institute*

of Movement Studies. Laban/Bartenieff Institute of Movement Studies, Inc., New York, 1987.

[26] Martha Davis. Steps to achieving observer agreement: The lims reliability project. *Movement Studies: A Journal of the Laban/Bartenieff Institute of Movement Studies*, 2:7–20, 1987.

[27] Cecily Dell. *A Primer for Movement Description: Using Effort-shape and Supplementary Concepts.* Dance Notation Bureau, Inc., New York, 1970.

[28] Scott L. Delp, Peter Loan, Cagatay Basdogan, and Joseph M. Rosen. Surgical simulation: An emerging technology for training in emergency medicine. *PRESENCE: Teleoperators and Virtual Environments*, 6(2):147–159, April 1997.

[29] David Ebert, Kent Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling: A Procedural Approach.* Academic Press, second edition, July 1998.

[30] N. Eshkol and A. Wachmann. *Movement Notation.* Weidenfeld and Nicholson, London, 1958.

[31] T. Flash and N. Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience*, 5:1688–1703, 1985.

[32] Stephen Glazier. *Random House Word Menu.* Random House, New York, 1992.

[33] Michael Gleicher. Retargeting motion to new characters. In *SIGGRAPH 98 Conference Proceedings*, pages 33–42, July 1998.

[34] Michael Gleicher and Peter Litwinowicz. Constraint-based motion adaptation. *The Journal of Visualization and Computer Animation*, 9(2):65–94, 1998.

[35] Larry Gritz and James K. Hahn. Genetic programming evolution of controllers for 3-D character animation. In *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 139–146, Stanford University, CA, USA, 13-16 July 1997. Morgan Kaufmann.

[36] Eric Guaglione. The art of disney's 'mulan'. In *Siggraph 1998 Course*, 1998.

[37] Ann Hutchinson Guest. *Choreo-graphics : a comparison of dance notation systems from the fifteenth century to the present.* Gordon and Breach, New York, 1989.

[38] Craig Hayes. Starship troopers. In Rick Parent, editor, *Conference Abstracts and Applications (Siggraph 1998)*, page 311, 1998. Animation sketch.

[39] Jessica K. Hodgins, Wayne L. Wooten, David C. Brogan, and James F. O'Brien. Animating human athletics. In *SIGGRAPH 95 Conference Proceedings*, pages 71–78, August 1995.

[40] Francis Edward Simon Hunt, George Politis, and Don Herbison-Evans. Led: An interactive graphical editor for labanotation. Technical Report 343, Basser Department of Computer Science, University of Sydney, 1989.

[41] Ann Hutchinson. *Labanotation.* Theatre Arts Books, New York, 3 edition, 1977.

[42] Paul M. Isaacs and Michael F. Cohen. Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. In *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 215–224, July 1987.

[43] Mary Ritchie Key. *Nonverbal communication : a research guide and bibliography.* The Scarecrow Press, Inc., Metuchen, N.J., 1977.

[44] Mary Ritchie Key and Janet Skupien. *Body movement and nonverbal communication : an annotated bibliography.* Indiana University Press, Bloomington, Indiana, 1982.

[45] Albrecht Knust. *Dictionary on Kinetography Laban (Labanotation).* MacDonald and Evans, Estover, Plymouth, 1979.

[46] Doris H. U. Kochanek and Richard H. Bartels. Interpolating splines with local tension, continuity, and bias control. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 33–41, July 1984.

[47] Yoshihito Koga, Koichi Kondo, James Kuffner, and Jean-Claude Latombe. Planning motions with intentions. In *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, pages 395–408, July 1994.

[48] Evangelos Kokkevis, Dimitri Metaxas, and Norman I. Badler. User-controlled physics-based animation for articulated figures. In *Computer Animation '96*, Geneva, Switzerland, June 1996.

[49] Rudolf Laban. *Choreutics.* Macdonald & Evans, London, 1966. Annotated and edited by Lisa Ullmann.

[50] Rudolf Laban. *The Mastery of Movement.* Macdonald and Evans, Plymouth, 4th edition, 1980. Revised and enlarged by Lisa Ullmann.

[51] Rudolf Laban and F. C. Lawrence. *Effort: Economy in Body Movement.* Plays, Inc., Boston, 1974.

[52] Labanwriter. Dance Notation Bureau Extension, The Ohio State University, Department of Dance, 1813 N. High Street, Columbus, Ohio 43210.

[53] John Lasseter. Principles of traditional animation applied to 3D computer animation. In *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 35–44, July 1987.

[54] Joseph F. Laszlo, Michiel van de Panne, and Eugene Fiume. Limit cycle control and its application to the animation of balancing and walking. In *SIGGRAPH 96 Conference Proceedings*, pages 155–162, August 1996.

[55] Zicheng Liu, Steven J. Gortler, and Michael F. Cohen. Hierarchical spacetime control. In *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, pages 35–42, July 1994.

[56] Macbenesh. DanceWrite, 119 Glen Cresent, Thornhill L4J 4W3, Ontario, Canada, (905)764-0046, DanceWrite@shaw.wave.ca.

[57] Vera Maletic. *Body, space, expression : the development of Rudolf Laban's movement and dance concepts.* Mouton de Gruyte, New York, 1987.

[58] Kevin Montgomery, Michael Stephanides, Joel Brown, Jean-Claude Latombe, and Stephen Schendel. Virtual reality in microsurgery training. In *Medicine Meets Virtual Reality*, volume 41, 1999.

[59] Carol-Lynne Moore and Kaoru Yamamoto. *Beyond Words: Movement Observation and Analysis.* Gordon and Breach Science Publishers, New York, 1988.

[60] Michael Moore, Christopher Geib, and Barry Reich. Planning for reactive behaviors in hide and seek. In *Proc. 5th Conference on Computer Generated Forces and Behavioral Representation*, 1995.

[61] Claudia L. Morawetz and Thomas W. Calvert. Goal-directed human animation of multiple movements. In *Proceedings of Graphics Interface '90*, pages 60–67, May 1990.

[62] J. Thomas Ngo and Joe Marks. Spacetime constraints revisited. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 343–350, August 1993.

[63] Laban/Bartenieff Institute of Movement Studies (LIMS). 234 Fifth Avenue, Room 203, New York, NY, 10001; (212)477-4299.

[64] Ken Perlin. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):5–15, March 1995.

[65] Ken Perlin and Athomas Goldberg. IMPROV: A system for scripting interactive actors in virtual worlds. In *SIGGRAPH 96 Conference Proceedings*, pages 205–216, August 1996.

[66] Janis Pforsich. Personal communication.

[67] Luca Prasso, Juan Bahler, and Jonathan Gibbs. The pdi crowd system for antz. In Rick Parent, editor, *Conference Abstracts and Applications (Siggraph 1998)*, page 313, 1998. Animation sketch.

[68] Marc H. Raibert and Jessica K. Hodgins. Animation of dynamic legged locomotion. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 349–358, July 1991.

[69] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 25–34, July 1987.

[70] Charles Rose, Michael F. Cohen, and Bobby Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics & Applications*, 18(5), September – October 1998.

[71] Ted Shawn. *Every Little Movement.* M. Witmark & Sons, New York, 1954.

[72] Karl Sims. Evolving 3d morphology and behavior by competition. In R. Brooks and P. Maes, editors, *Artificial Life IV Proceedings*, pages 28–39. MIT Press, 1994.

[73] Karl Sims. Evolving virtual creatures. In *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, pages 15–22, July 1994.

[74] Scott Steketee and Norman Badler. Parametric keyframe interpolation incorporating kinetic adjustment and phasing control. In *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 255–262, July 1985.

[75] Frank Thomas and Ollie Johnston. *Illusion of Life: Disney Animation.* Hyperion, New York, 1995.

[76] Deepak Tolani. *Inverse Kinematics Methods for Human Modeling and Simulation.* PhD thesis, University of Pennsylvania, 1998.

[77] Xiaoyuan Tu and Demetri Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, pages 43–50, July 1994.

[78] Y. Uno, M. Kawato, and R. Suzuki. Formation and control of optimum trajectory in human multijoint arm movement. *Biological Cybernetics*, 61:89–101, 1989.

[79] Munetoshi Unuma, Ken Anjyo, and Ryozo Takeuchi. Fourier principles for emotion-based human figure animation. In *SIGGRAPH 95 Conference Proceedings*, pages 91–96, August 1995.

[80] Munetoshi Unuma and Ryozo Takeuchi. Generation of human motion with emotion. In Nadia Magnenat Thalmann and Daniel Thalmann, editors, *Computer Animation '91*, pages 77–88. Springer-Verlag, New York, NY, 1991.

[81] Harald G. Wallbott. The measurement of human expression. In Walburga von Raffler-Engel, editor, *Aspects of Nonverbal Communication*, pages 203–228. Swets and Zeitlinger, Lisse, Netherlands, 1980.

[82] Harald G. Wallbott. Analysis of nonverbal communication. In Uta M. Quasthoff, editor, *Aspects of Oral Communication*, pages 480–488. Walter de Gruyter, New York, 1995.

[83] Shirley Weitz, editor. *Nonverbal Communication: Readings with Commentary.* Oxford University Press, New York, 2 edition, 1979.

[84] Jane Wilhelms. Dynamic experiences. In Norman I. Badler, Brian A. Barsky, and David Zeltzer, editors, *Making them move: mechanics, control, and animation of articulated figures*, pages 265–279. Morgan Kaufmann, 1991.

[85] Andrew Witkin and Michael Kass. Spacetime constraints. In *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 159–168, August 1988.

[86] Andrew Witkin and Zoran Popović. Motion warping. In *SIGGRAPH 95 Conference Proceedings*, pages 105–108, August 1995.

[87] QinXin Yu and Demetri Terzopoulos. Synthetic motion capture for interactive virtual worlds. In *Computer Animation 1998*, pages 2–10. IEEE Computer Society, June 1998.

[88] Vladimir M. Zatsiorsky. *Kinematics of Human Motion.* Human Kinetics, Champaign, IL, 1998.

[89] Jianmin Zhao and Norman Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics*, 13(4):313–336, October 1994.

[90] Liwei Zhao and Norman Badler. Gesticulation behaviors for virtual humans. *Sixth Pacific Conference on Computer Graphics and Applications*, pages 161–168, October 1998.