1-1-2001

Synthesis and Acquisition of Laban Movement Analysis Qualitative Parameters for Communicative Gestures

Liwei Zhao
University of Pennsylvania

Norman I. Badler
University of Pennsylvania, badler@seas.upenn.edu

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-01-24.

Synthesis and Acquisition of Laban Movement Analysis Qualitative Parameters for Communicative Gestures

Abstract

Humans use gestures in most communicative acts. How are these gestures initiated and performed? What kinds of communicative roles do they play and what kinds of meanings do they convey? How do listeners extract and understand these meanings? Will it be possible to build computerized communicating agents that can extract and understand the meanings and accordingly simulate and display expressive gestures on the computer in such a way that they can be effective conversational partners? All these questions are easy to ask, but far more difficult to answer. In this thesis we try to address these questions regarding the synthesis and acquisition of communicative gestures.

Our approach to gesture is based on the principles of movement observation science, specifically Laban Movement Analysis (LMA) and its Effort and Shape components. LMA, developed in the dance community over the past seventy years, is an effective method for observing, describing, notating, and interpreting human movement to enhance communication and expression in everyday and professional life. Its Effort and Shape component provide us with a comprehensive and valuable set of parameters to characterize gesture formation. The computational model (the EMOTE system) we have built offers power and flexibility to procedurally synthesize gestures based on predefined key pose and time information plus Effort and Shape qualities.

To provide real quantitative foundations for a complete communicative gesture model, we have built a computational framework where the observable characteristics of gestures - not only key pose and timing but also the underlying motion qualitites - can be extracted from live performance, either in 3D motion capture data or in 2D video data, and correlated with observations validated by LMA notators. Experiments of this sort have not been conducted before and should be of interest not only to the computer animation and computer vision community but would be a powerful and valuable methodological tool for creating personalized, communicating agents.

Disciplines

Computer Sciences

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-01-24.

SYNTHESIS AND ACQUISITION OF LABAN MOVEMENT ANALYSIS QUALITATIVE PARAMETERS FOR COMMUNICATIVE GESTURES

LIWEI ZHAO

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.

2001

Dr. NORMAN I. BADLER
Superviser

Dr. VAL TANNEN
Graduate Group Chair

ABSTRACT

SYNTHESIS AND AQUISITION OF LABAN MOVEMENT ANALYSIS QUALITATIVE PARAMETERS FOR COMMUNICATIVE GESTURES

Liwei Zhao

Supervisor: Norman I. Badler

Humans use gestures in most communicative acts. How are these gestures initiated and performed? What kinds of communicative roles do they play and what kinds of meanings do they convey? How do listeners extract and understand these meanings? Will it be possible to build computerized communicating agents that can extract and understand the meanings and accordingly simulate and display expressive gestures on the computer in such a way that they can be effective conversational partners? All these questions are easy to ask, but far more difficult to answer. In the thesis we try to address these questions regarding the synthesis and acquisition of communicative gestures.

Our approach to gesture is based on the principles of movement observation science, specifically Laban Movement Analysis (LMA) and its Effort and Shape components. LMA, developed in the dance community over the past seventy years, is an effective method for observing, describing, notating, and interpreting human movement to enhance communication and expression in everyday and professional life. Its Effort and Shape component provide us with a comprehensive and valuable set of parameters to characterize gesture formation. The computational model (the EMOTE system) we have built offers power and flexibility to procedurally synthesize gestures based on predefined key pose and time information plus Effort and Shape qualities.

To provide real quantitative foundations for a complete communicative gesture model, we have built a computational framework where the observable characteristics of gestures—not only key pose and timing but also the underlying motion qualities—can be extracted from live performance, either in 3D motion capture data or in 2D video data, and correlated with observations validated by LMA notators. Experiments of this sort have not been conducted before and should be of interest not only to the computer animation and computer vision community but would be a powerful and valuable methodological tool for creating personalized, communicating agents.

Acknowledgements

These pages mark the culmination of a journey years in the making. When I trace back the footprints of my graduate study, there are numerous people and guides who have been graciously helping me in many aspects of my life and to whom I am grateful.

First, and foremost, I am deeply indebted to my advisor Professor Norm Badler. Without his years of support and guidance, I cannot complete my Ph.D study at Penn. Without his vision and inspiration, this work could not be possibly done. As a researcher, his great asset is his intuitive instinct. His comments are always to the point. As an advisor, he does not tell you what you want to hear but always tells you what you should hear. I admire his vision, broad knowledge, managerial abilities and selfless devotion to his students. He will be my lifelong lighthouse, navigating me no matter where I am going.

I am also grateful to my thesis committee members: Professor Martha Palmer, Professor Justine Cassell, Professor Jean Gallier, Professor CJ Taylor, and Professor Dimitris Metaxas. Their varied perspectives brought truly useful suggestions and insightful comments that strengthened the work and its presentation.

Others that deserve special thanks include: Diane Chi for being the first to implement a kinematic Effort model; Monica Costa for helping build the Shape models; Janis Pforsich and John Chanik for providing their LMA expertise and performing numerous motion samples for the project; Deepak Tolani for providing the inverse kinematics code; Bjoern Hartmann for making a considerable contribution to the design and implementation of the Maya EMOTE plug-in; Professor DeLiang Wang at the Ohio State University for teaching me the neural networks; Harold Sun and Christian Vogler for setting up the motion capture system; Shan Lu for contributing his expertise on computer vision and developing the video capture system; Tsukasa Noma for implementing the C++ version of PatNets and

the collaboration on the gesticulation project; Jan Allbeck for providing the agent model and painstakingly reading through my thesis and providing sound advice; Koji Ashida for spending many hours on video recording and editing; Fred Azar for being an "actual" actor in many scenarios; Karin Kipper and William Schuler for helping implement a natural language interface; Seung-Joo Lee for building the balloon model; Christian Vogler for sharing his expertise on ASL and performing the ASL signs; Charles Erignac for many beneficial discussions on geometric curves and interpolation methods; Rama Bindiganavale, Jan Allbeck, and Matt Beitler for their lab software support; Adam Kendon and Craig Martell for providing the tour guide video and the collaboration and evaluation on the tour guide project; Mike Felker and Gail Shannon for graduate program coordination and administration; Karen Carter for complaining about the air condition system and helping make the lab a cool place to work.

I also appreciate the many past and present members of the graphics lab who are valuable colleagues and friends, including Meeran Byun, Ting Chen, Sonu Chopra, John Graneiri, Gang Huang, Suejung Huh, Pei-Hwa Ho, Laiyuan Liu, Thomas Jurgensohn, Ying Liu, Jianping Shi, Hogeun Shin, Bond-Jay Ting, and Daniel Widyono.

I wish to express my deepest gratitude to my parents for giving me the strength, courage, and independence to pursue my dreams, and always standing by my side no matter what happened. I still remember vividly what they said to me when I left for college when I was fifteen years old. I know by heart that my attainment of this highest academic degree is the dream come true for them. I want to make them happy, forever.

Finally, my heartfelt thanks, gratitude, and love go to my dearest wife, Nan Ya, for her love, patience, and encouragement. She put up with the late nights and the long hours of absence and preoccupation required by this project. I owe her countless dinners, laundry, and shoppings. I dedicate this work to her.

This research is partially supported by NSF, NASA, ONR, and NSF Center for Sign Language and Gesture Resources.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Human movement ranges from voluntary, goal-oriented movements to involuntary, subconscious movements. Voluntary movements include task-driven actions, such as walking to get somewhere or speaking. Involuntary movements occur for physiological or biological purposes; for instance, blinking, balancing, and breathing. A wide class of movement falls in between these two. In general, this class is characterized by movements which occur in concert and perhaps unconsciously with other activities. We note two interesting subclasses of this class of movements. One subclass consists of low-level motor controls that accomplish a larger coordinated task. For instance, unconscious finger controls form grasps, leg and foot coordination enable walking or running, and lip movements generate speech. Another important subclass are communicative acts: facial expressions, limb gestures, and postural attitude. While computer animation researchers have actively studied all these classes of human movements [3, 30, 16, 27, 130, 138, 5, 26, 28, 97], it remains difficult to procedurally generate convincing communicative "natural" limb and postural movements.

McNeill and Cassell [90, 30, 27] approach communicative gestures through several categories:

- **Iconics** represent some feature of the subject, such as the shape or spatial extent of an object.
- **Metaphorics** represent an abstract feature of the subject, such as exchange, emergence, or use.

- **Deictics** indicate a point in space that may refer to people or spatializable things.
- **Beats** are hand movements that occur with accented spoken words and speaker turn-taking.
- **Emblems** are stereotypical patterns with understood semantics, such as a good-bye wave, or the OK-sign.

Such an approach has served to make conversational characters appear to gesture more-or-less appropriately while they speak and interact with each other or real people. The impression that one gets when watching even the most recent efforts in making convincing conversational characters is that the synthetic movements still lack some qualities that make them look "right". Indeed, the characters seem to be doing the right things, but with a kind of robotic awkwardness that quickly marks the performance as synthetic. It is not a computer animation problem *per se* — conventional but skilled key-pose animators are able to produce excellent gestures in three dimensional (3D) characters by careful application of classic rules for conventional animation [124, 81]. But there is a considerable gap between what an animator intuits in a character (and is therefore able to animate) and what happens in a procedurally synthesized movement.

The McNeill/Cassell approach to gesture is rooted in psychology and experimental procedures that use human observers to manually note and characterize a subject's gestures during a story-telling or conversational situation. The difficulty in this approach is *hidden within the decision to call something a gesture.* That is, the observer notes the occurrence of a gesture and then records its type. This kind of recording fails to capture the parameters of movement that makes one particular gesture appear over another (its movement qualities), as well as what makes the gesture appear at all. This issue is crucial in the studies of Kendon [65], who tries to understand the deeper question: What makes a movement a gesture or not? In his work, a gesture is a particular act that appears in the arms or body during discourse. There may be movements that are not gestures and there may be movements that are perceived as gestures in some cultures but not in others. So clearly, the notion of "gesture" as a driver for computer-generated characters cannot be—in itself—the primary motivator of natural movements.

1.1 Our Approach

To address this apparent dilemma, we argue that looking only at the psychological or descriptive notion of gestures is insufficient to capture motion qualities needed by animated characters. We need to look toward movement representations outside the constraints of communicative acts. We find that Laban Movement Analysis (LMA) [35, 76, 39, 11, 77, 60, 12, 88, 93] and its Effort and Shape components provide us with the most comprehensive and valuable set of parameters for describing the form and execution of the qualitative aspects of movements. LMA is not the same as Labanotation [60]. The former addresses movement qualities while the latter addresses places and positions. We have created and implemented prototype computational models of Effort and Shape to apply qualitative parameters to generate expressive movements on the torso and the limbs of an articulated human figure [5, 31, 32]. We call this system EMOTE (Expressive MOTion Engine).

Our EMOTE approach to gesture augments the McNeill/Cassell approach by addressing a missing dimension: gesture exists not just because it has underlying linguistic relationships but also because *it has some distinctiveness in its Effort and Shape parameters.* Our approach meshes perfectly with the perspective offered by the LMA proponents: "Gesture ... is any movement of any body part in which Effort or Shape elements or combinations can be observed [12]." Our EMOTE approach to gesture also complies with two other important LMA concepts. The first one is synthesized by Bartenieff when she observes that it is not just the main movement actions that let us identify behavior but it is the sequence and phrasing of Effort and Shape parameters that express and reinforce content [12]. The other concept is best expressed by Lamb: a gesture localized in the limbs alone lacks impact, but when its Effort and Shape characteristics spread to the whole body, a person appears to project full involvement, conviction, and sincerity [78]. In the animated Gilbert and George characters produced for [30], torso involvement was precluded. The characters appear to nod and move their arms in a vaguely disturbing, disembodied fashion. When the rest of the body is moved along with limb gestures, the greater weight of the torso naturally reacts to and absorbs limb forces.

Effort and Shape qualities provide us with a comprehensive and valuable set of parameters to characterize gesture formation. The EMOTE model offers power and

flexibility to procedurally synthesize gestures based on predefined key pose and time information plus Effort and Shape qualities. To provide real quantitative foundations for a complete communicative gesture model, we have elaborated the EMOTE system in several new ways:

- Bypass manual key pose specification by connecting a motion capture system with the EMOTE system and automatically extracting the key point definitions from live performance.
- Experiment with porting EMOTE to a deformable human model in a commercially available visualization package (Alias|Wavefront's Maya 3.0).
- Connect EMOTE with an agent model so that agent motion manners can set appropriate EMOTE parameters for gesture performance.
- Investigate motion analysis techniques for extracting EMOTE Effort parameters from live inputs, both in 3D motion capture data and in 2D video data.
- Validate the automated acquisition of EMOTE Effort parameters by experiments using professional LMA notators for ground truth.

It is also very important to distinguish between motion quality and expressivity and communicativity. The difference lies in that motion quality emphasizes how movement is performed and how stability, mobility, exertion, and recuperation are dynamically interleaved, while expressivity and communicativity stress more the degrees to which linguistic or psychological information are effectively conveyed through the motion channel. The association of motion qualities with the underlying gestural movement facilitates but does not necessarily determine the expression and communication of individual predispositions and characteristics. Different motion qualities distributed over the same underlying motion may produce dramatically different gestures and hence may effect an observer's interpretation of the internal state of the performer. On the other hand, motion qualities plus the underlying movement are not necessarily sufficient to determine the linguistic or psychological meaning of a gestural movement. We believe there are other factors including contextual variables at work determining the real meaning of gesture.

Nonetheless, by building computational models of motion qualities we open the door to later research that might rigorously study the effect of these qualities on expressivity and communication.

This work does not attempt to address the problem of gesture recognition, nor does it intend to build a model for expressing or communicating the linguistic or psychological meaning of gesture. Instead, our current approach is focused on gesture analysis and synthesis — we first convert the gestural movements from observation into *a computational representation*, which comprises not merely motion forms but also Effort and Shape motion qualities. We are then able to use the computational representation to generate a variety of gestures by adjusting its motion quality parameters. We believe it is the computational representation that forms the basis for further quantification of more complicated gesture models and ultimately for gesture understanding and recognition.

1.2 Overview

The remainder of this document presents our implementation of a bi-directional gesture framework where Effort and Shape parameters are used both to synthesize expressive limb and torso movements, and in reverse, the Effort and Shape parameters, as well as the key pose and timing information, are extracted from live performance. We shall review the related work and scope of the gesture research in Chapter 2. Chapter 3 introduces the basic concepts and components of Laban Movement Analysis (LMA) theory. Gesture synthesis is presented in Chapter 4, while gesture acquisition is elaborated in Chapter 5 and Chapter 6, dealing with acquisition from 3D and 2D data, respectively. We conclude with our contributions and future work in Chapter 7.

Chapter 2

Related Work and Scope

Broadly speaking, there are two separate threads running through the gesture research field. In one thread, there is work by linguists, psychologists, neurologists, choreographers, physical therapists, and others. Basically, they are not committed to building a computational gesture model to verify their theories, and are rarely concerned with any computer implementation implications of their work. Their concern is largely with a conceptual understanding of gesture and its function. Although their work often involves some deep analysis, most of their models are *qualitative* and *theoretical*, making it very difficult to justify their correctness, generality, and appropriateness. The other thread of research on gesture operates in areas such as computer vision, human-computer interaction (HCI), human motor control, and computer graphics and animation. Most of these approaches are in a system-oriented context. Various computerized systems have been built to recognize, analyze, and/or synthesize gestures for control, modeling, or animation purposes. While these approaches explore different areas of research, some fundamental questions remain unanswered, such as whether or not gesture really serves any measurable function and utility, how gesture and speech are correlated and how gesture reveals affect.

We shall investigate all the important approaches taken within each thread to give a complete overview about the state-of-the-art in gesture and carefully position our approach. The remainder of this chapter is organized as following: we embark on our investigation by presenting several possible definitions of gesture, followed by a taxonomy and classification of gestures done by several major researchers. Then we move on to qualitative gesture

models and computational gesture models. In the section on qualitative models, we summarize the experiments, hypotheses, and theories about the fundamental questions of gesture that have intrigued researchers for years in disciplines such as psychology, linguistics, theater, dance, and cognitive science. Their approaches provide valuable input to building a computational gesture model. Some system builders are connected to and cognizant of the work being done in some of the areas. For example, Cassell's work [30] has psychological/linguistic roots. Our focus is to build a computational gesture model.

2.1 Definition of Gesture

There is no single universally accepted definition of what a gesture actually is. Kendon, one of the few people who presented a definition, defines a gesture thus: "... for an action to be treated as a *gesture* it must have features which make it stand out as such." [65] Clearly, this is not really a definition though it suggests the use of features as the distinguishing characteristics. McNeill [90] defines a gesture as "movements of the arms and hands which are closely synchronized with the flow of speech." This explicitly excludes the involvement of the body or gestures without speech. Some researchers have a narrower focus, for example, Cassell [27] focuses on hand gestures that co-occur with spoken language. *American Heritage Electronic Dictionary* gives a broader definition: "a motion of the limbs or body made to express or help express thought or to emphasize speech." While all are useful descriptions of gestures, none really gives a generative or analytical view suitable for computational implementations.

2.2 Taxonomy and Classification

The lack of a clear definition of gestures in general raises another issue: the taxonomy of gestures. Over the years a number of gesture classification schemes have been proposed. Table 2.1 summarizes six major taxonomies of gestures, starting with Efron's work in the 1940's [41] and most recently that of McNeill in 1995 [90], as well as Koons and Wexelblat's classification scheme with focus on computer interpretation [134]. The summary provides only a rough comparison, which omits some of the details of each scheme. For example,

Efron	**Rimé & Schiaratura**	**Freedman & Hoffman**	**Kendon**	**McNeill & Levy**	**Koons & Wexelblat**	**Identifying Characteristics**
kinetographics/ physiographics	physiographics	literal-reproductive	physiographics	iconics	iconics	picture the semantic content of speech
ideographics	iconics	concretization minor and major qualifying	ideographics	metaphorics	pantomimics	picture an abstract idea rather than a concrete object or event
batonlike	speech-marking	punctuating	gesticulation	beats	beats	mark the rhythm and pace of speech
symbolics/ emblematics	symbolics	symbolics	autonomous gestures	symbolics/ emblematics	symbolics/ modalizing	standardized gestures, complete within themselves without speech
deictics	deictics	– none –	– none –	deictics	deictics/ Lakoffs[a]	point at people or spatializable things
– none –	– none –	– none –	– none –	cohesives	– none –	emphasize continuities can consist of iconics metaphorics, even beats
– none –	– none –	speech failures	– none –	Butterworths[b]	Butterworths/ self-adjusters	arise in response to speech failure, help to recall words

Table 2.1: Gesture classification

[a]Lakoffs, named after philosopher George Lakoff, are gestures used to show directionality of the metaphoric utterances. For example, as pointed out in [134], we tend to use such a gesture to spatialize verbal metaphors (such as "I'm feeling down" to indicate an unhappy mood).

[b]This category is named after Brian Butterworth, a scholar in Britain who has argued that many gestures arise in response to speech failures, but McNeill [90] thinks they are only a small fraction of all gestures.

McNeill distinguishes two kinds of iconic gestures ([90], pp.12-13). It is hard to compare these taxonomies; difficulties, as noted by Wexelblat [134], include:

1. A lack of a guided, systematic, and disciplined classifying method behind them: their authors observe and describe identifying characteristics for each category of gestures but provide no rule base in making decisions of classification.

2. Each taxonomy uses different terminology, including using the same term to mean different gestures.

3. Some taxonomies are incomplete – two do not include deictics, one of the most basic forms of gesture;

4. Categories in each taxonomy are not exclusive but instead there might be overlaps.

Although each scheme has its special usefulness due to the different historical backgrounds of its development, we are prone to agree with that of McNeill because of its comprehensive coverage and its focus on narrative and conversational gestures.

2.3 Qualitative Gesture Models

2.3.1 Psychological and Linguistic Gesture Models

Modern psychological and linguistic research on gesture based on systematic observations or experiments started with Efron's work [41] in 1940's[1]. In Efron's gestural theory, three phases in each gesture are identified: *preparation*, *stroke*, and *retraction*. In the preparation phase, the hands are raised to the location where the gesture begins. In the stroke phase, the actual gesture is performed, and the hands relax and fall back to the resting places in the retraction phase.

Following Efron's seminal work, three main researchers – Kendon [64, 65, 66, 67], McNeill & Levy [91, 89, 90],and Rimé & Schiaratura [112, 113] – have made significant contributions to the gesture research in the psychological/linguistic domain[2].

[1]The work by F. Descartes (1839), C. Darwin (1872), W. Wundt (1900) and K. Buhler (1933) are also creditable, but more comprehensive and therefore useful research has been carried on contemporarily, in a new interdisciplinary field which spans psychology, linguistics, and semiotics.

[2]There are also a number of researchers whose work made some contributions to the field. For example,

- **Kendon**

  Kendon began his research by attempting to determine what people saw when they watched gestures [64]. His experiments involved having subjects view videotapes of people speaking in a foreign language that the viewers did not understand. Kendon reported that the viewers had no trouble picking out gestures.

  Through investigating the relationship between a *gesture phrase* and a *tone unit* of speech, he proposed his gesticulation theory. A *gesture phrase* is the "nucleus of movement with definite form and enhanced dynamic qualities ... preceded by a preparatory movement and succeeded by a movement which either moves the limb back to its rest position or repositions it for the beginning of a new gesture phrase" ([65], pp. 34). A *tone unit* is a "phonologically defined syllabic grouping with a single intonation tune" ([65], pp. 34). He finds the stroke of the gesture phrase occurs simultaneously with (or slightly preceding) the nucleus of the tone unit. Also, Kendon notes that modes of expression are not equivalent. First, they are used in different contexts. For example, gestures might be produced more often when the conditions of speech reception are impaired by a noisy environment or by limited knowledge of a foreign language. Second, gestures and speech do not obey similar constraints in the turn-taking system. Finally, what is difficult to express in speech may be conveyed by gesture, including spatial information such as distance, orientation, and trajectory that are elusive to speech [67].

  Kendon [66] orders gestures of varying natures along a continuum of "linguisticity:" Gesticulation $\Rightarrow$ Language-like Gestures $\Rightarrow$ Pantomimes $\Rightarrow$ Emblems $\Rightarrow$ Sign Languages. As we move from left to right: (1) the obligatory presence of speech declines, (2) the presence of language properties increases, and (3) idiosyncratic gestures are replaced by socially regulated signs [90]. In other words, the formalized, linguistic component of the expression present in speech is replaced by signs going from gesticulation to sign languages. This is supportive of the idea that gesture

the classification and explanation by Ekman and Friesen [43] is quite meticulous and credible. But their work overlaps considerably with what we will cover and therefore is not explicitly listed. Many more specialized researchers investigate some specific gesture related areas. For example, Klima and Bellugi [68] (1979), Stokoe (1960, 1972), Friedman (1977) and Liddell (1980) have done some linguistically oriented studies on gesture and ASL (American Sign Language). This specialized research is not reviewed in this thesis.

and speech are one integrated system. Also, the continuum sorts out gestures of fundamentally different kinds. Many researchers refer to all forms of nonverbal behavior as "gesture," failing to distinguish among different categories, with the result that behaviors that differ fundamentally are confused or conflated.

- **McNeill & Levy**

  McNeill and Levy conducted experiments that involved having subjects watch a cartoon and then narrate the action of the cartoon to other subjects who have not seen it. They made the same discovery as Kendon: speech and gesture are part of a coherent whole [91]. In his most recent work, McNeill has elaborated this idea by providing a conceptual framework that includes both gesture and language [90]. According to McNeill, gestures present meaning in a form fundamentally different from that of speech: (1) gestures are *noncombinatoric* – two gestures produced together do not combine to form a larger, more complex gesture; (2) there is no hierarchical structure of gestures made out of other gestures, which contrasts with the hierarchical structure of language; (3) gestures do not share such linguistic properties as standard forms and duality of patterning. Despite these differences, McNeill argues that gestures are so closely linked to speech that both should be viewed within a unified conceptual framework. In support of his claim, McNeill enumerates five reasons ([90], pp. 23-25): (*i*) gestures occur only during speech; (*ii*) they are semantically and pragmatically coexpressive; (*iii*) they are synchronous; (*iv*) they develop together in children; (*v*) there is a simultaneous breakdown of gestural and linguistic abilities in aphasia.

  McNeill also addresses timing related issues [90]. He hypothesizes three "rules" that govern how gesture and speech synchronize: *phonological*, *semantic*, and *pragmatic rules*. The phonological rule means that the stroke of the gesture precedes or ends at, but does not follow the phonological peak syllable of speech, which complies with Kendon's observations. The semantic rule is that gesture and speech must cover the same idea if they co-occur. This rule is even applicable in cases of multiple gestures and multiple clauses. The pragmatic rule says that gestures and speech serve the same pragmatic functions if they co-occur. Although theoretically it is possible to

violate these rules, McNeil claims that no exception has been caught in a wide variety of observations and experiments [90].

- **Rimé & Schiaratura**

  Rimé & Schiaratura grounded their research by conducting experiments that involved putting speakers in conditions where the speakers could see their listeners and where there was no listener. They found the gesture frequency was not seriously *decreased* when the mutual visibility of partners was experimentally suppressed or while speaking by telephone [112]. Thus, they deduced that gesture must serve some function or purpose for the speaker more than just communicative. However, Krauss and Hadar [72] argue that this experimental result is not necessarily in conflict with the view that gestures are generally intended to be communicative because people always gesture when they speak spontaneously – they simply cannot suppress it when they are on the telephone.

  Another important experiment conducted by Rimé & Schiaratura involved restricting speakers from using gestures during their speech. They found that speakers tended to give poorer descriptions and induce more compensatory motor activity of eyebrows and fingers[3]. Furthermore, careful analysis of the semantic content of the speech showed that the speakers used *more* words but the speech was *less* clear and *less* fluid [113][4]. Again, this empirical evidence can be interpreted to support theories like McNeill's that gesture and speech are elements of a single integrated system. On the other hand, these experimental data sets can also be interpreted to support the hypothesis of Krauss and Hada that gestures facilitate access to lexical memory because the effects of restricting gesturing on speech were found to be similar to those of making word retrieval difficult by other means such as requiring subjects to use rare or unusual words [72]. We discuss Krauss and Hadar's approach in the following because their approach offers another psychological and cognitive dimension.

[3] D.M. Dobrogaev (1929) did some similar experiments and reported that speakers instructed to curb facial expressions, head movements, and gestural movements found it difficult to produce articulate speech, but the experiment lacked necessary controls and the results were presented mostly in impressionistic terms.

[4] Graham and Heywood did some similar experiments [51] but reported contradictory findings – they asked six speakers to describe abstract line drawings to a small audience of listeners, and found the elimination of gesture had no particularly marked effects on speech performance, however, their studies were criticized for some methodological problems [72].

- **Krauss and Hadar**

  Krauss and Hadar based their theory on a subset of speech-related gestures – "lexical gestures," which are relatively "long, broad, and complex arm-hand movements that often incorporate shapes or dynamics related to the content of the accompanying speech" ([72], pp. 99). In their view, lexical gestures facilitate lexical retrieval. As shown in Figure 2.1, the gesturally-represented spatiodynamic features are fed via the kinesic monitor to the formulator, where they participate in lexical search and facilitate the retrieval. Some findings also support the idea of semantic facilitation (Hadar 1998), suggesting entry via the lemma system, or the idea of word form facilitation (McNeill 1966), suggesting entry via the word form system[5]. The link from the speech production system to the gesture production system is not shown, even though such a path is necessary to tell when to terminate a gesture. In a mechanism proposed by Krauss *et al.* [73] to explain the tendency of gestures that are associated with hesitations, they suggest such a link: lexical selection switches off the gesture production system. On this account, if the set of spatiodynamic features is realized successfully in the lexical selection, the gesture production system is aborted. Consequently, many gestures are activated but may not actually get executed; difficulties encountered in lexical selection may simply allow sufficient time for the gesture to reach execution, or expedite such an execution. Alternatively, a gesture simply may be terminated when a new gesture is initiated. A similar but comparatively more complex dual mechanism is proposed by Butterworth and Hadar [25, 53]. In their view, some gestures are activated directly from short-term memory while others are initiated by failures of lexical retrieval. They assert that retrieval failures often result in a re-run of lexical selection, and during such re-runs, the formulator attempts to gather more cues for lexical selection by activating non-propositional representations. It is these non-propositional representations that actually initiate a gesture. Some pathological data has been reported [72] supporting of the hypothesis, but further investigations should be carried out to attest to it.

[5]Hadar and Butterworth [25] also suggest a link to the conceptualizer, which implies that the spatial/dynamic features would directly contribute to the construction of the speaker's communicative intention and only affect lexical retrieval indirectly. But the available experimental data so far [46] is not supportive of this hypothesis.

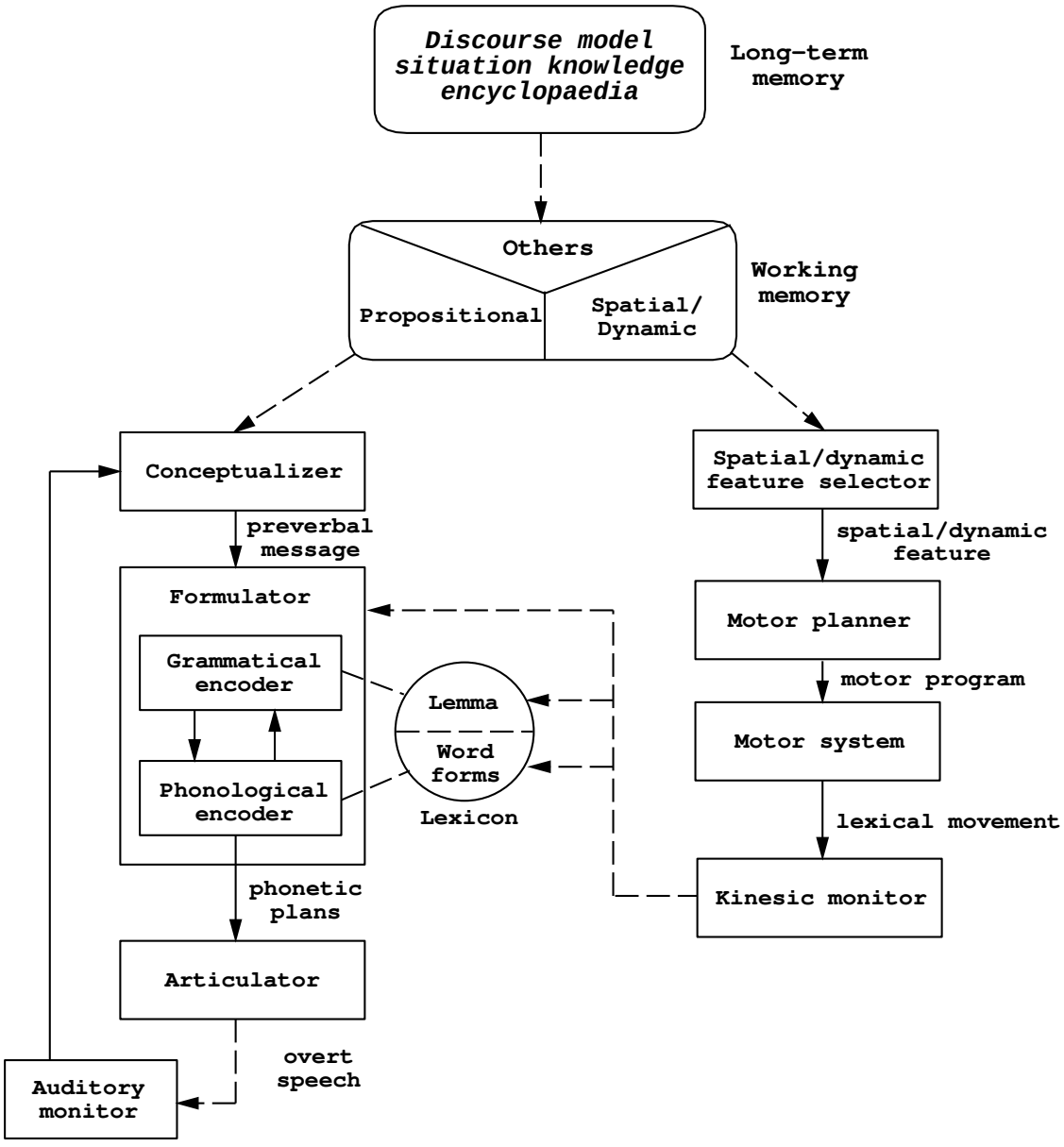

Figure 2.1: Krauss and Hadar's gestural facilitation model [72]

Moreover, both mechanisms apply only to those gestures that are associated with hesitations. These gestures usually amount to about 30% of total lexical gestures in a normal subject. Either the mechanism has to be revised or a different mechanism must be hypothesized to account for gestures that are not associated with hesitations.

Topics	**Assertions**	**Divergences**
Gesture and Speech	Gesture is closely linked to speech; gestures occur when there is some discrepancy between the units of thought and the units of speech	The full sentence is planned in advance during the gesture preparation phase, or there is a single process underlying gesture and speech
	Gesture and speech are not equivalent. Gesture has some nonlinguistic properties, more freedom	– none –
Gesture and Time	Gesture phrase occurs simultaneously with (or slightly proceeding) the relevant speech units	The amount of time precedence is variable, but no one has data showing gestures occur later than the related speech units
Gesture and Communication	Symbolic/emblematic, deictic gestures are generally communicatively intended and communicatively effective	The question is whether there is adequate justification for assuming that all or most co-speech gestures are so intended

Table 2.2: Assertions and divergences in psycholinguistic approaches

Although there are no definitive psycholinguistic models so far to explain all the functionalities of communicative gestures accurately and convincingly, models proposed by various researchers provide a convenient way of systemizing available data. They also compel theorists to make explicit the assumptions that underlie their formulations, thus making it easier to assess in what ways, and to what extent, different theories differ. We conclude this section by providing a chart (see Table 2.2) that summarizes several major assertions that are generally accepted by psycholinguistic researchers, as well as conceptions that diverge among them[6].

[6]There are also a number of other assertions and disagreements that are not listed in the chart.

2.3.2 Gesture Models in Cognitive Science

Building cognitive models is very much a research area at the forefront of psychological and AI research. In the previous section, we presented some psychological models, but with a focus on aspects that are associated with linguistics. In this section our investigation is more from a perspective of cognitive science.

Generally speaking, a cognitive model employed in psychological research serves as a vehicle for understanding human behavior. If the model is successful at producing human-like behavior under certain assumptions, a hypothesis can be formulated that different behavior will emerge under different assumptions. Change those assumptions in the model and see how it behaves. Explorations with models in such a way can then be used to design experimental conditions that are likely to show measurable effects. Speech-accompanying gesture, as an important human behavior, has been extensively studied within a broadly cognitive context. Numerous studies have yielded contradictory hypotheses, theories, and empirical evidence [46]. We list two representative hypotheses and their corresponding cognitive models in the following.

- **Coactivation Models**

  Some researchers have assumed inevitable activation of the gestural system during speech production and gestures is visible manifestation of the speaker's ongoing thinking process. In this conception, gesture and speech share origins and are triggered simultaneously, then separate into two different output channels. However, if the interaction occurs at the initial phase of the speech process, as McNeill assumes ([89], pp. 367), the model is not without problems – it is not clear how to identify the common stage, where the interaction occurs, and the output stage, where dissociation may be observed. On the other hand, to explain situations in which some gestures relate to prosodic features (such as stress, melodic contour) or syllabic structure of the verbal utterance, some researchers assume a collaborative model in which the interactions between gesture and speech happen at several different levels [25]. Nevertheless, this requires that the gesture depends not only on the expressed content

For example, investigators generally agree that the type of information communicated is an important determinant of gestural behavior but diverge with regard to other important factors such as speech connectivity, speech tempo, and familiarity of the spoken language, etc.

but also on the surface characteristics of the sentence, and thus, on the motor planning of the utterance. To make things more complicated, coactivation models imply a direct relationship between speech fluency and gesture production: the more one speaks, the more gestures are performed. In some circumstances, this may be true, but in other conditions (i.e., for bilingual subjects), gestural rate and speech fluency can be inversely related [46].

- **Competition Models**

  Empirical evidence shows that the gestural stroke phases alternate with rest phases and gesture production is sometimes prevented or delayed instead of simultaneously activated during the process of expression of thought [46]. According to this evidence, some researchers hypothesize that gesture and speech are two rival tasks – they compete and inhibit each other through their connections. Under a resource-sharing view, they assume that resources are limited and, thus, the attention load required for one task reduces the amount that can be allocated to the other concurrent task. From such a perspective, gesturing while speaking constitutes a particular dual-task diagram in which the speakers are required to divide their attention between concurrent processes. When the attention load reaches its maximum, hesitation pauses occur. A qualification that emerged from research on attentional mechanisms shows that fully automated tasks no longer require attention, which might be the case with some of the gestures performed while speaking when made without attentional control or in coordination with speech movements. Moreover, during silence, gesture production could be inhibited by the processing load required by speech planning. Although this model can be supported by some observations, the level at which inhibitions occur is not clearly specified yet.

In summary, the cognitive study of the interactions between gesture and speech from a cognitive perspective does not provide us with a consistent answer. Relationships between gesture and speech are found to be sometimes facilitative and sometimes competitive. The experimental literature has to be carefully reviewed in relation to these different hypotheses. Together with experimental analysis, observations from pathology, developmental psychology and neuropsychology may be used to delineate the functioning

of communicative gestures.

2.3.3 Performative Gesture Models in Theater and Dance

For years, in theater and dance, gestures have been seen as the most appropriate means of expression. People use gestures to enhance the emotional content of their characters and stories. Gestures communicate to the audience whether or not they should like or hate a character, or whether the story is a tragedy or comedy. For the avant-garde theater gesture is not simply a decorative addition but rather the source, cause, and director of language. Gestures can be very culturally-based. In ballet, movement takes as its base the Greco-Roman ideals of posture and movement. Erect, open posture and slow, expansive gestures are seen as elegant and graceful, while narrow, cramped and jerky movements are seen as ugly and poor [115]. Appropriately planned and selected, gestures can create a mood and arouse an emotional response in the audience [40]. In a play, the director looks at the combined movement of the cast and treats movement as an extension of line, mass, and form. The actors themselves must keep in mind the amount of movement in a gesture and the amount of space covered whether they are conveying power or weakness on stage. The length of the gesture, whether long or short, the intensity of the gesture, whether strong or light, will add to the emotional content. Motion is an important cue toward helping the audience to understand a character. The wrong movement or motion qualities can ruin a character or even the whole dynamic of the stage.

Two gesture models in theater and dance have long been recognized and analyzed: one is the ballet model and the other is the mime model. Both are highly stylized and codified. Gestures in ballet are based on the movement potential of the human body and they select, shape and emphasize certain features of movement, while gestures in mime are generally a presentation of ordinary actions with stresses on certain features, evoking the everyday world. For more details about these two gesture models, see [115].

2.4 Computational Gesture Models

Scientific research dependent on a qualitative model is a difficult and slow effort because investigators lack tools that could make measuring relevant phenomena inexpensive and

highly repeatable so that they can verify their theories and adjust their models easily and quickly. Advances in digital computing equipment and computational approaches to image processing, tracking and recognition, simulation and animation provide a possible answer. In fact, numerous computational models have been built on the topic of gesture study.

2.4.1 Gesture Models in Multimodal Interfaces and Computer Vision

New approaches to multimodal interfaces using hand/arm gestures, as well as voice/speech and facial expressions, have been proposed in recent years. This process has been remarkably expedited as virtual reality (VR) and distributed virtual environment (DIVE) becomes a part of our present space and time. Many researchers advocate that gestures are more natural to use in multimodal interfaces than conventional cumbersome human-computer interaction devices such as mice and keyboards.

2.4.1.1 First Steps

The ground-breaking work probably was done by Richard Bolt [22] [7] in the early 1980's. In his famous system "Put-That-There," Bolt used a combination of prototype Polhemus 6D pose tracking system and some simple voice recognition software. On the screen, the user saw objects. The user would then "point" at an object and say "Put that ..." move her finger to where she wanted the object to be, and say "there." Pure speech commands were also possible, "Put the red ball to the right of the yellow box."

The advantage of this technique is robustness and immediate visual feedback. The disadvantages are *inflexibility*, because the gesture recognition was hard-wired in mechanical devices and only those gestures it was designed for can be recognized, and *inconvenience*, in that the mechanical sensors had to be mounted on and calibrated to the user.

Since then, many technologies and approaches have been proposed and developed. These approaches can be roughly classified as glove-based or vision-based. Most of these approaches, however, focus on hand gestures only. The functions that arm gestures and body postures play in the human-computer interaction have been largely neglected.

[7] At the same time Myron Krueger also did some pioneer work in building Virtual Reality applications [74, 75].

2.4.1.2 Glove-Based Approaches

In a glove-based approach, some mechanical or optical sensors are usually attached to a glove, which transduces finger flexions and abductions into electrical signals in such a way that hand postures can be determined. The relative position of the hand is determined by some additional sensors (magnetic or acoustical) mounted to the glove. A detailed survey of glove-based input devices can be found in [122].

Baudel and Beaudouin-Lafon [14] develop a real-time glove-based system in which hand gestures are used to control browsing in a hypertext presentation. The system is called *Charade*. *Charade* can recognize sixteen hand gestural commands, each of which comprises three phases: start posture, dynamic phase, and end posture. The commands are distinguished based on the start posture as well as the dynamic phase.

Glove-Talk is a gesture-to-speech interface, developed by Fels and Hinton [45], using a VPL DataGlove^{TM} connected to a DECtalk speech synthesizer via five independent neural networks. They defined a 203 gesture-to-word vocabulary and used *Glove-Talk* to map complete gestures to complete words.

2.4.1.3 Vision-Based Approaches

A vision-based approach is more natural and convenient than a glove-based approach. Yet it is also more difficult, due to the limitations of today's computer vision in handling a highly non-convex and flexible volume like a human hand. Several different approaches have been proposed so far [110, 87, 47, 96, 80, 120, 59]. The most straightforward one is simply the use of a single video camera or a pair of cameras to acquire visual information about a person under a certain environment and try to extract the necessary gestures. Nonetheless, this approach faces several difficult problems: segmentation of the moving hand from a sometimes very complex environment, analysis of hand motion, tracking of hand position relative to the environment, recognition of hand postures, etc. To lower the burden, some systems use passive markers or marked gloves. The others use restrictive setups: uniform background, very limited gesture vocabulary, or just a simple static posture analysis.

Markers are usually placed on the fingertips. They are colored in such a manner

that they can be easily detected using image histogram analysis. Once the markers are detected and tracked, the gesture can be recognized using several classification schemes. Maggioni [87] describes a hand tracking system (called *Gesture-Computer*) that is based on the use of a specially marked glove. The glove has two slightly off-centered, differently colored circular regions. Using single camera images *Gesture-Computer* can compute several image geometry parameters based on the first and second moments and use them to estimate hand position and orientation.

Rehg and Kanade [110] propose a complete hand gestural interface, called *DigitEyes*, applicable in a restricted background. With *DigitEyes*, finger tip and link parameters can be extracted from either 2D or 3D video images using edge-based techniques. These parameters then can be applied to a 3D cylindrical kinematic model of the human hand with 27 degrees of freedom (DOF).

Some systems approach the issue through analyzing and extracting features that are associated with the images of hand/arm postures. The analyzed features range from basic geometric properties, such as image moments, to those that are the results of a more complex analysis (i.e., neural networks [45]). Hand/arm silhouettes are one of the simplest yet widely used features. Silhouettes can be easily extracted from local hand/arm images in restricted background setups. In case of complex backgrounds, techniques such as color histogram analysis can be employed. In his *VideoPlace*, *VideoDesk* and *VideoTouch*, Krueger [74, 75] uses silhouettes to analyze images and identify users' body parts. Segen and Kumar [120] use some edge-based techniques to extract from hand posture images, local features such as "peaks" and "valleys."[8] Gestures are then classified based on these local features. Experiments conducted on a 3D graphical editor, a virtual fly-through, and a video game find the parameter estimation is stable. The common characteristics shared by all the approaches is that they do not result in the estimation of the real hand parameters such as joint angles. The systems are applicable to both simple hand tracking and more complex gesture classification. Furthermore, some systems have taken a voice-vision combined approach [125, 86, 131, 132]. Such a multimodal approach is promising in offering a more natural human-computer interface. In Table 2.3 we summarize the

[8]Peaks are features whose curvatures are positive with a magnitude greater than a fixed threshold while valleys are features whose curvatures are negative with a magnitude less than the threshold.

Application and System	**Author**	**Input/Output Techniques Used**	**Gestural Capabilities Supported**
Put-That-There	Bolt [22]	6D pose tracking and simple speech recognition	Pointing and dragging gestures
Videoplace *Videodesk* *Videotouch*	Krueger [75]	Multiple video cameras, sensory floor; output sound graphical displays	Can identify users' head arms, legs, hands, fingers movements and response accordingly
Charade	Baudel & Beaudouin-Lafon [14]	DataGlove^{TM} and position sensing devices	Sixteen gestural commands
Gesture-Computer	Maggioni [87]	Marked glove, head tracking, mono camera	Six static hand gestures
Ymir (with animated characters: *Gandalf*, *Bilbo* and *Roland*)	Thórisson [125]	Cyberglove^{TM} and body tracking system, speech synthesizer	Understand limited utterance, intonation, body stance, hand gesture, eye gaze, head-face direction
GloveTalk	Fels & Hinton [45]	DataGlove^{TM}; speech synthesizer	203 gesture-to-word vocabulary, map complete gestures to complete words
ASL-recognizer	Vogler & Metaxas [131, 132]	Three cameras or a magnetic sensor system	Recognizes 53 ASL signs
GestureVR	Segen & Kumar [120]	Two cameras	Three hand gestures: point, reach, and click
Rea	Cassell *et al.* [28]	Two cameras	Turn-taking gestures
DigitEyes	Rehg & Kanade [110]	Mono camera or stereo camera; output 3D hands	Can track a fully articulated hand (27 DOF)
Hand-controlled TV	Freeman & Weissman [47]	A Flex-Cam^{TM} video camera; output graphical menu	Two gestural commands: open-hand and closed-hand
ALIVE (*Vitual Dogs*)	Maes, Blumberg, & Pentland [86]	Mono camera, use *Pfinder* as the hand/head/body tracking system; auditory output	Various gestures: pointing, hand-shaking, etc. Gestures can be interpreted depending on current states and past history

Table 2.3: A few HCI systems that employ gestures

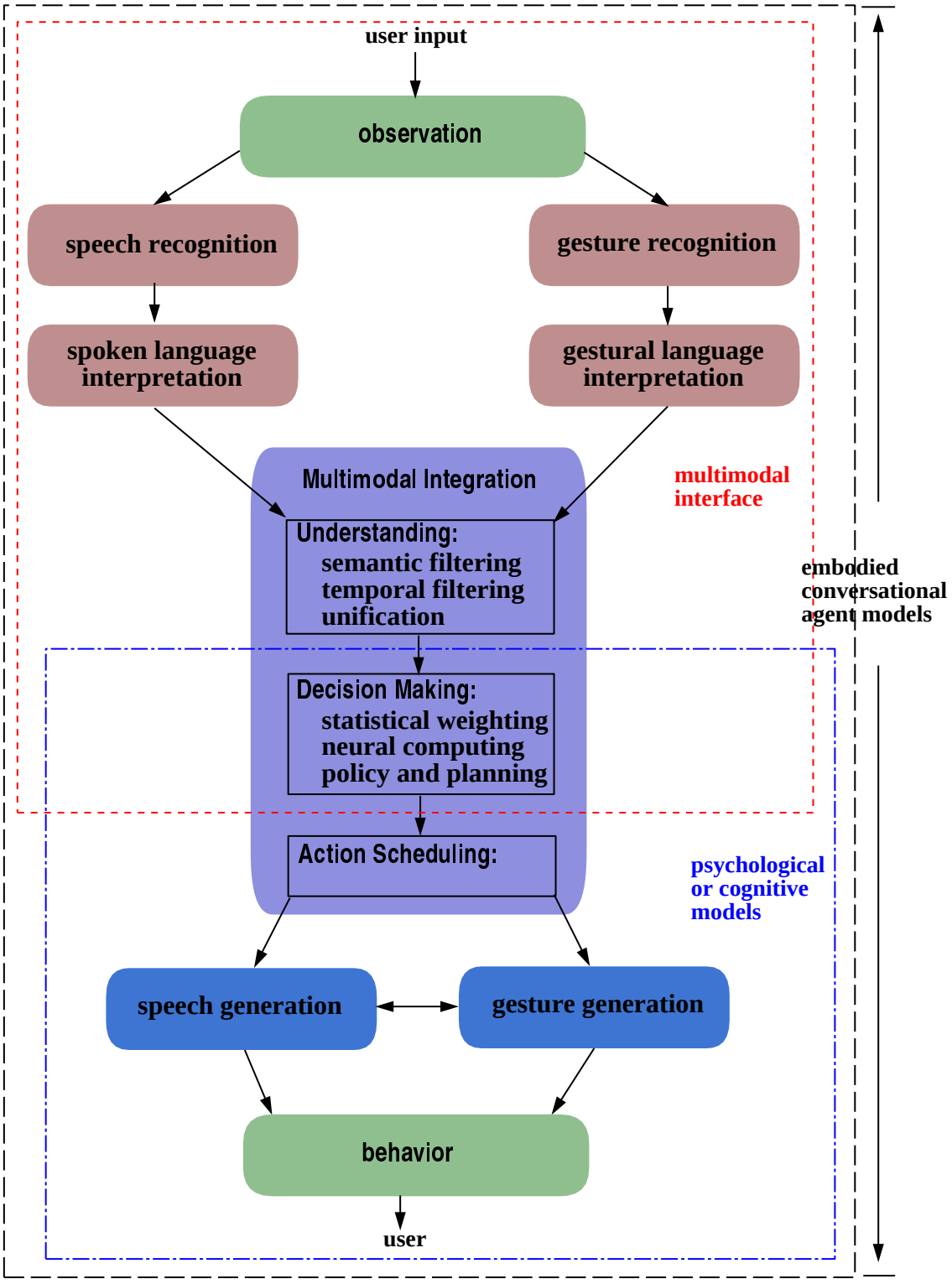

Figure 2.2: Information processing flow in psychological, multimodal interface, and conversational agent models

approaches that are related to our discussion.

2.4.1.4 Multimodal Interface

To allow our highly skilled communicative behavior to control interactions in a more natural, robust, reliable, and efficient way, multimodal systems usually combine different input modes. According to the studies conducted by Oviatt [102, 103], multimodal systems that incorporate multiple input modes can facilitate new uses of computing – some input modalities may be suited for some specific tasks and conditions but less ideal or inappropriate in others. With a multimodal architecture, however, adaptive weighting of the input modes can be performed to enhance and stabilize the system's overall performance. In addition, errors can be reduced or avoided if parallel or duplicate input modes are available, giving a more accurate and stable system.

It is interesting to compare the multimodal interface models with the psychological models we reviewed previously and the embodied conversational agent models that we will cover in the following. From an information processing flow standpoint, multimodal interface systems focus on the input side, which can be either language oriented (speech, gestures, and pen input) or more broadly defined (postures, gaze patterns, and lip movements), while psychological or cognitive models focus more on the output side where the speech and gestures are coordinated, complementing and/or competing with each other. Embodied conversational agent models cover both input and output. Thus, formally, the architecture of any approach to an embodied conversational agent model [30, 29, 10] consists of both a multimodal interface and a speech/gesture generator (see Fig. 2.2). As the human computer interactions shift toward natural multimodal behavior, the interface design may become more conversational or social in style, rather than limited to commands or mouse control.

2.4.2 Gesture Models in Computer Graphics

Studies on qualitative models and interface designs are so vast and extensive that comparatively the computer graphics literature is rather sparse, especially on the topic of creating natural gestures procedurally.

Cassell, Badler and their colleagues [30] describe a computational gesture-speech system which can automatically generate and animate conversations between two human-like agents with synchronized speech, gestures and facial expressions. In this work four communicative gesture types, *Iconics*, *Metaphorics*, *Deictics* and *Beat*, have been distinguished and studied on the cognitive basis of a gesture-speech relationship. The gesture and speech generations are managed by two cascaded planners. The first is the domain planner, which is a database of facts describing the way the world works, the goals of the agents, the beliefs of the agents about the world, and the communicative actions the agents will execute. The second is the discourse planner, which manages the communicative actions the agents must take in order to agree on a domain planner and in order to remain synchronized while executing a domain planner. The domain planner executes by decomposing an agent's current goals into a series of more specific goals according to the hierarchical relationship between actions specified in the agent's beliefs about the world. Once decomposition resolves a plan into a sequence of communicative actions to be performed, the discourse planner, in coordination with the domain planner, generates proper symbolic intonation and/or gesture specifications. The intonation specification which includes speech text, pitch accents, and phrasal melodies is converted automatically to a form suitable for input to the AT&T Bell Laboratories TTS synthesizer. Gesture generation is synchronized with speech output and carried out by a group of coordinated parallel transition networks (PaTNets) [7]: *parse-net* parses phoneme representations and is responsible for instantiating *gest-net* or *beat-net*; *gest-net* controls the generation of iconic, deictic, and metaphoric gestures, while *beat-net* controls the generation of beat gestures. The PaTNets system then issues gesture requests to the animation system, telling the human-like figure to rest, make a beat motion, or make a gesture involving the hand, wrist, and/or arm. Arm and wrist motions are specified by target positions and orientations while hand motions are specified in terms of a limited but expandable library of handshapes. Gestural movements are apparently predefined and can only be parameterized in terms of alteration of single gesture phrases, i.e., foreshortening the relaxation phase when the prerecorded "canonical" gesture time exceeds actual timing constraints [29], but there seems to be no means of coherently modifying the gestural movement while preserving natural movement features.

The impact of this work is two-fold. First, to psycholinguistic researchers, it provides *computational* simulations that gesture and speech can be generated and coordinated through a control scheme which originates from one single mental representation. As we mentioned earlier, many sources [64, 65, 67, 90] have already suggested that gesture and speech are physiologically, psychologically and cognitively linked, but most of these are *descriptive* and *distributional* and have been very difficult to evaluate and justify. This system offers a computerized testbed in which psycholinguistic researchers can explore the model by changing model parameters to simulate different varieties of, or breakdowns in, communication. Secondly, and more importantly, to computer graphics researchers, it introduces some psychological ground truth about what kind of gestures the computer graphics and animation community should consider with higher priority in order to build life-like communicative virtual humans.

Following Cassell's lead, new problems in gesture generation were exposed:

1. **Coarticulation**: Generating a smooth transition from one gesture to the next without returning to a specific rest pose.

2. **Spatialization**: Integrating a deictic gesture into the surrounding context.

3. **Selection**: Generating a metaphoric gesture that might be associated with an abstract concept.

4. **Expression**: Modifying the performance of a gesture to reflect the agent's manner or personality.

Problem 1, *coarticulation*, refers to changes in the articulation of a motion segment depending on preceding (backward coarticulation) and upcoming (forward coarticulation) segments. The problem has been addressed by a number of computer graphics researchers [106, 34, 52, 109]. Pelachaud *et al.* [106] use a coarticulation facial model to integrate actions of each muscle or group of muscles on the face as well as the propagation of their movements. Cohen and Massaro [34] present techniques to synchronize lip movement and voice output based on the articulatory gesture model of Lofqvist [85]. They use overlapping dominance functions to coproduce the speech segment and lip movement. Guenter *et al* [52] describe a scheme using space-time constraints and inverse kinematic

constraints to create transitions between motion segments of a human body model with 44 degrees of freedom. They use an interpreter of a motion language to allow the user to manipulate motion data, break it into segments, and reassemble the segments into new and more complex motions. In NYU's *Improv* project, Perlin and Goldberg [109] describe a technique called *motion blending* to automatically generate smooth transitions between isolated motions without jarring discontinuities or the need to return to a "neutral" pose. Some aspects of the issue such as preparatory and termination actions are addressed by Badler, Palmer and Bindiganavale and their colleagues [6, 19, 18]. Although there are other aspects of the issue that remain unresolved, the problem is relatively well explored.

Problem 2, *spatialization*, requires that the desired gesture is modified to point or align the gesturing body part with the spatial referent. This problem essentially is an inverse kinematics problem. An advantage of using analytical or hybrid analytical/numerical methods [126] is that they generally behave consistently and are not sensitive to minor perturbations of the starting state: when applied to the gesture spatialization problem, a satisfactory final posture can usually be achieved rapidly.

Problem 3, *selection*, entails determining gestures that people would likely interpret and accept as "representative" during a communicative act. But researchers disagree on whether gestures are products of communicative intentions or memory representations. The ramifications of this disagreement are two completely different approaches.

One approach accounts for what communicative intentions or conceptual information are to be conveyed in gesture and exactly at which time. Different types of information are expressed in different kinds of gestures, which can be predefined as a collection of abstract *gesture templates*, encoding the relevant information. The template is then passed down to a number of lower levels where the gestural movement is actually coordinated and carried out. A number of computer graphics researchers have been working along this line. In [97, 138], Noma, Zhao and Badler propose a representative mapping from concepts to gestures such that they are selected based on stylized rhetorical speaking. Olveres *et al.* [99] develop a system in which avatars can infer user emotions from text input in a fuzzy-logic fashion and, based on what emotions are conceptually derived, select appropriate facial expressions to display. The selection can be affected by some explicit cues such as keywords, modifiers like adverbs, and emoticons such as :-(and :-). In the

BODYCHAT project of Cassell and Vilhjalmsson [130], they develop a prototype system that allows a user to communicate via text while avatars *automatically* pick up appropriate gestures such as salutes and turn-takings, and simple body functions such as eye blinks. Again, these gestures are predefined and each is encoded with some specific conceptual or psychological meanings. When the linguistic context is matched, appropriate gestures are selected and invoked. In her recent work [27], Cassell rejects the idea of the use of a dictionary of gestures that speakers draw from to produce gestures and that listeners draw from for gesture interpretation due to the evidence about the absence of a one-to-one mapping between form and meaning in everyday gesture.

The other approach, in contrast, regards that gestures precede conceptualization, taking into consideration that gestures may convey information that is not explicitly intended [65, 67, 90]. In the gesture model introduced by Krauss and Hadar [72], they assume a separate module to be responsible for the selection of relevant and consistent spatial and dynamical features out of the activated representations in spatial or visual working memory. Referring to a Kendon's example [65] of the speaker saying ".. with a big cake on it ..." while making a series of circular motions of the forearm with index finger pointing downward, they argue that the iconic gesture accompanying the word "cake" is not part of the speaker's communicative intention to show the cake is large and round, but instead is reflected gesturally that the cake is represented in the speaker's memory as large and round. While they have not yet built a computational model to verify their assumptions, their approach provides an alternative way of selecting representative gestures under some linguistic circumstances. As things currently stand, there is so little experimental data to constrain theory on when and what gesture is selected that any processing model is considered to be tentative and speculative. Further investigations need to be done before one or another model is confirmed or disconfirmed.

Problem 4, *expression*, is concerned with how to add expressiveness to the performance of gestures so that an agent's manner, emotions and personality can be vividly depicted. The expression problem itself can be split into two subproblems: one is expressive movement generation and the other is coherent quality attachment. There have been an abundance of research results for generating expressive movements [136, 108, 129, 24, 2, 137, 33, 84, 52, 50, 111, 7, 128, 13, 21, 55, 17], while research in finding motion qualities

that are coherently attached to gestures is rather sparse.

2.4.2.1 Expressive Movement Generation

Techniques for the generation of expressive movement can be roughly divided into four categories: (1) adding expressiveness to neutral motions, or providing tools to modify motion expressions, (2) making the existing motions fit some constraints, (3) adding secondary movements, and (4) controlling behaviors. This classification is made for convenience of the presentation; in practice, these techniques are frequently combined to achieve the best animation results.

- **Adding expressiveness to neutral motions or providing tools to edit motion expressions**

  Several researchers have suggested methods of adding expressiveness to animated motions using such methods as stochastic noise functions [108], Fourier function models [129], signal processing [24], or emotional transforms [2].

  Perlin uses rhythmic and stochastic noise functions to define time varying parameters that drive animated puppets [108]. The user controls the puppet through a set of buttons, representing a set of primitive actions and discrete states of the puppet. The system can smoothly blend the selected primitive actions into a coherent animation if the relative contribution (weight) of each action is specified properly. The user can tune expressions by adding a pseudorandom noise function to joint motions, modifying joint angle frequency and amplitude, and controlling transition times for different actions. The noise functions give the effect of subtle restlessness and weight shifting, adding low frequency "texture" to the motion. The resulting animated puppet is thus in constant motion and appears to have a dynamic, life-like motion quality. However, it is hard to judge the range of expression possible with the system. It seems the scheme only works fine for rhythmic, repetitive actions, such as walking and dancing. Non-rhythmic motions are selected stochastically for variations. Also, varying expressions by modifying the puppet's scalar joint angles over time t via sine and cosine functions is non-intuitive and limits movement qualities. Setting transition times and action weight also requires a certain artistry and skill. If these

parameters are applied naively, the resulting animations can be disastrous.

Unuma *et al.* use Fourier analysis techniques to interpolate and extrapolate human locomotion data to capture a wide variety of expressions [129]. For instance, they can generate various degree of "tiredness" by interpolating between a normal "walk" and a "tired" walk. In addition, by quantifying the differences between the coefficients of a Fourier function model for a neutral locomotion and those for emotion-driven locomotion, they can generate different Fourier characteristic functions, which can then be, individually or in combination, applied to other neutral locomotion to produce different variations and expressivities. However, the process of generating Fourier functional models and characteristic functions could be very lengthy.

Bruderlin and Williams apply multiresolution filtering techniques from the image and signal processing domain to manipulate the neutral motions by treating motion parameters (such as joint angles and coordinates) as sampled signals [24]. When a motion parameter signal passes through a series of filters, an animator can add an emotional component, exaggerate the movement, or constrain joint ranges by adjusting the amplitudes of high, middle, or low frequency bands appropriately.

Witkin and Popovic describe a technique for editing of captured or keyframed motion by warping and blending motion parameter curves [137]. For each motion curve, the animator chooses a few keyframes and modifies their poses using a suitable timewarp function. The modified poses serve as constraints on a smooth deformation to be applied to the captured motion. The new motion curve satisfies the constraints while preserving the final details of the original curve. The animator warps each motion curve independently. The motion clips are concatenated using Perlin's blending techniques [108]. A wide range of new realistic motions can be created from a single prototype motion sequence. However, motion warping is a purely geometric technique, not based on any deep understanding of the motion's structure. Some warps may appear unnatural and distorted.

Amaya *et al.* present a method to derive emotional transforms by taking the differences between neutral and emotion-influenced actions [2]. They then apply the derived emotional transforms to neutral actions to generate a wide range of

movements with different types of expressivities. In order to express individuals' differences in gender, age, manner, culture, and personality, this approach may need to store and retrieve a large number of emotional transforms. As the same individual shows different emotions under different scenarios and internal states, this requires a clever indexing scheme if an emotional transform database is designed. The awkwardness in the manipulation may indicate that emotional transforms, along with the noise functions, Fourier functions and filtering functions, only capture the essence of the movement superficially.

- **Making the existing motions fit some constraints**

  Witkin and Kass present a spacetime constraint technique to produce the optimal motion which satisfies a set of user-specified constraints [136]. Cohen develops a spacetime control system which allows a user to interactively guide a numerical optimization process to find an acceptable solution in a feasible time [33]. Liu *et al.* use a hierarchical wavelet representation to automatically add motion details [84]. Guenter *et al.* adopt this approach to generate a smooth transition between motion clips efficiently [52]. Gleicher simplifies the spacetime problem by removing the physics-related aspects from the objective function and constraints to achieve an interactive performance [50].

- **Adding secondary movements**

  The use of secondary movements has been proposed as a way to enliven animated characters and/or scenes. Although the secondary movements are not the primary focus of the motions of an animated character, their absence can distract or disturb the viewer, making the character unbelievable and unnatural. One approach is to add secondary movements to the primary movements of walking characters based on user-specified personality and mood [94]. Another approach focuses on passive motions like the movement of clothing and hair, generated in response to environment forces or the movements of characters and other objects [98]. The secondary movements and the primary movements combined give a richer and more varied set of movements capable of responding to subtle changes in an animated character's personality, manner, and environment.

- **Controlling behaviors**

  Expressive movements are also investigated with an aim to build autonomous characters (or creatures) that are endowed with varying behaviors, personalities or goals. The prominent work in this area is that of Reynolds [111], followed by Badler [7], Bates [13], Tu and Terzopoulous [128], Hodgins [55], Thalmann and Thalmann [17], and Hayes-Roth [54]. Although self-animating characters or creatures have demonstrated more-or-less different high-level behaviors, their low-level movements are frequently stereotyped, or clumsy and unnatural. In addition, the expressions and their manifestations are usually hard-wired in the code and very inflexible to reconfigure and extend. Blumberg and Galyean [21] and Funge *et al.* [48] address these concerns by introducing mechanisms that give the animator greater control to direct autonomous characters to perform specific tasks, however, their work is at best partially successful, and the impression that one gets from watching even the most recent effort in making autonomous agents is that their basic movements are still fairly unexpressive, lacking the qualities that make them look "right."

In general, most of these techniques are valuable for generating expressive movement; however, either these methods require an off-line modeling process for each different type of expression, or the modification process involves nonintuitive low-level manipulations in such a way that some artistry or expertise is demanded in order to generate natural, expressive movements, or both. In addition, they may prove difficult or costly to use in generating the range of expressivity of human communicative gestures.

2.4.2.2 Coherent Quality Attachment

"Movement" and "gesture" are not synonymous. Some movements, such as involuntary or subconscious movements, are not gestures. Also, some movements are perceived as gestures in one culture but not in another. Gesture, as a special sort of movement, links closely to the individual's plans, emotions, imaginations, and desires, which are embodied in the whole body and manifested in the motion qualities during communicative acts. Gestures produce movements but movements do not necessarily produce gestures. Actually, gestures

of any type exist not just because they have underlying movements but also because they have some distinctiveness in their motion qualities. Different motion qualities distributed over the same underlying motion may produce dramatically different gestures. Suppose the underlying motions consists of arm movements portraying a single beat gesture that would accompany an accented speech utterance. By slowing down its time course and making it more indirect we may turn the beat gesture into an emblematic gesture (hand wave). Starting with a slow forward pointing motion, we can crank it up its Sudden and Direct qualities to focus and accent the movement into a deictic gesture ("yes, I mean YOU"). By making shoulders rise highly, making the muscles more tense, and adding more weight, we may turn it into a metaphoric gesture (i.e., threatening somebody). Thus, motion qualities associated with the underlying movement are *essential* components in a communicative gesture. However, the nature of these components have largely been ignored in most of the computational gesture models.

2.5 Acquisition of Communicative Gestures

We choose the word "acquisition" very deliberately here. Our work is not gesture recognition—*we are only concerned with acquiring the motion qualities associated with the underlying movement in communicative gestures, rather than determining the (linguistic or psychological) meaning of the gestural movement.*

As we mentioned previously, motion quality components play an indispensable role in the process, but recognizing motion qualities is closely related to gesture recognition. Thus, we shall briefly go through the approaches and techniques employed in gesture recognition.

Generally speaking, gesture recognition consists of two subproblems: feature representation and classification. Thus, formally, any complete gesture recognition framework consists of two subsystem: the *representer* and the *classifier*. The representer takes the raw data, captured through mechanical, optical, magnetical, or acoustic sensors, and outputs its internal representation. The internal representation, often a set of parameters and features extracted from the data, is in the most convenient form for the classifier, to take as input and hence output an appropriate classification, if one exists. Approaches to gesture recognition can be classified as template matching, statistical

methods, or neural networks, according to the classification scheme employed.

2.5.1 Template Matching

The simplest and most straightforward method of recognizing gestures is template matching. Within this method, essentially there is no representation stage. The raw sensor data is used as input to the classifier which typically uses an Euclidean closest-neighbor function to measure the similarity between the input and the templates of values. The input is either admitted as a member of the same class as the template to which it is most similar (or nearest), or rejected as belonging to none of the possible classes if the measurement is higher than the similarity threshold (too far from the nearest template).

Zimmerman and Lanier [141] use a template matching based method for recognizing postures. For each posture to be recognized each sensor has a range of values that are valid[9]. At each sample time, the sensor readings are compared with the values of the posture templates. The absolute value of the difference for each sensor is summed for each template. The gesture with the minimum sum, below a global threshold, is the one chosen. Lipscomb [83] uses a comparatively more complex multiresolution approach. During the recognition process, the templates are examined first at the lowest resolution and only if successful at the level would the template proceed to matching at a higher resolution level.

Template matching is easy to develop, computationally efficient, and practically very accurate. There are serious drawbacks with the use of templates, however. For example, how to make the templates adaptive? Adaptability plays a critical role in the system's performance, since most gestures will not be reproduced even by the same user with perfect accuracy, and when a range of users are allowed to use the system, the variation becomes even greater. Also, template matching does not have the formal and iterative approach to training that statistical classifiers and neural networks have.

2.5.2 Statistical Classification

Functionally, statistical classifiers operate in the same way as template matching – mapping from an m-feature vector to a point in n-space. The mapping function, however, uses

[9]They also designed a calibration scheme to allow the ranges to be altered to suit different users.

statistical techniques, such as Bayesian maximum likelihood theory [105, 62], to decide which class the input *most likely* belongs to.

One of the most important works on gesture recognition using statistical methods is that by Rubine [116]. In his work gesture comprises a 2D path of a single point over time[10]. The features chosen are geometrically based on the path and computed incrementally. A statistically based evaluation function, computed over the features, decides the classification.

Ball and Breese [10] use Bayesian networks to diagnose the emotions and personality of the user from speech and a variety of observable nonlinguistic behaviors such as size and speed of gestures. Causal links in the Bayesian networks capture the significant dependency from components of emotion and personality to these observable effects. A standard probabilistic inference algorithm based on [105] is used to update the estimates of emotional state and personality given the observations.

2.5.3 Neural Networks

Neural networks have received much attention for their successes in pattern recognition [117, 135, 92, 119]. Gesture recognition is no exception to this and several systems have been reported in the literature [95, 15, 23].

Murakami and Taguchi [95] use a set of recurrent neural networks to recognize 42 finger-alphabet gestures taken from Japanese Sign Language (JSL) with an accuracy of up to 92.9%. But the system works poorly when applied to JSL word gestures which involve free hand movements. The neural nets can distinguish any two JSL word gestures but are not very reliable in identifying an arbitrary gesture from a learned set. Beale and Edwards [15] employ a multilayer perceptron model [117, 92] to classify input into five postures, taken from American Sign Language (ASL). The structure of the net includes ten input units each associated with a sensor of a DataGlove^{TM}, five output units (one for each of a, i, e, o, and u), and a single hidden layer which consists of three hidden units. They reported a high recognition accuracy and found both the learning rate and network momentum [117, 92] had a negligible effect on the final effectiveness. This may indicate that their data set is very simple and the learning task is very straightforward.

[10]The gesture is also called 2D mouse-based gesture.

Both systems only apply to discrete gestures instead of continuous gestures, so this clearly affects the naturalness of the gestures their systems are eligible to recognize [134].

Brooks [23] reports use of a neural net to control a mobile robot by interpreting DataGlove motion. In order to incorporate dynamic gestures into the system's vocabulary, Kohonen nets[11][70] are employed to recognize paths traced by degrees of freedom in n-dimensional space. Each Kohonen net, typically as small as 20 units, is trained to recognize a single gesture. However, the system is only an early prototype since the experiments conducted are very simple such as opening the thumb and index finger simultaneously and moving from a neutral posture to a grasping posture.

2.6 Summary and Our Approach

In terms of techniques, template matching, statistical classification, and neural network matching can be combined or mixed, depending on specific systems and applications. What differentiates these approaches from one to another is the feature extraction: almost every approach we investigated chooses a special feature extraction method, either for practical usefulness or for empirical purposes. What is missing is that the whole body of current approaches does not clearly point to a set of relevant features which are consistent, and less susceptible to noise and other external, environmental factors.

Our approach is *unique* in that we are working towards such a set of relevant features: Effort and Shape qualities. As we mentioned before, Effort and Shape qualities are a set of high-level parameters that describe qualitative aspects of human movement that relate to individual predisposition and characteristics. Furthermore, Effort and Shape qualities or their combinations, when they are involved in a communicative gesture, are observable.

If we look at the problem from an even broader context, it is clear that gesture recognition (or acquisition) is closely related to handwriting and speech recognition. Indeed, they can be viewed from a signal processing point of view as a time-variance analysis. In handwriting recognition research, it has long been known that the most important theoretical problem is to find a set of extractable features which are hardly

[11] Kohonen nets are formally defined in linear algebra, thus, strict linear algebraic relationship between gestural patterns can be learned, however, much analysis needs to be done to ensure the gestural patterns are algebraically suitable for training.

affected by handwriting distortions [71]. In the recognition of speech, the same thing holds true, that is, to find features of the speech waveform that are at higher levels where they are more insensitive to noise without losing details such as stress and intonation [82]. We believe this applies to gesture recognition (or acquisition), too.

Badler originally proposed the use of Effort as a higher level of control for human figure animation [8]. Bishko suggested analogies between the "Twelve Principles of Animation" [124] and Laban Movement Analysis. She showed that there is an abstract relationship between LMA and traditional animation techniques [20], but did not provide any computational means to exploit the relationship. Others [123] have done work with computerizing Labanotation but primarily focused on automation of the dance recoding rather than qualitative aspects of movement. Chi [31] created and implemented a kinematic analog to the Effort component. We, including Monica Costa[12], have extended her system to include the Shape qualities, the torso, and the legs for the gesture synthesis. We further use Effort qualities and their combinations as a set of higher level features to be extracted for gesture acquisition.

[12]During her sabbatical at University of Pennsylvania on a fellowship from National Scientific and Technological Development Council (CNPq) of Brazil.

Chapter 3

Laban Movement Analysis

Laban Movement Analysis (LMA) is a method for observing, describing, notating, and interpreting human movement for the purpose of improving awareness, efficiency, and ease of movement and to enhance communication and expression in everyday and professional life[1]. Originated in Germany at the beginning of the 20th century by Rudolf Laban (1879-1958), pioneer of European modern dance and proponent and theorist of movement education, LMA today is a creative synthesis that has been considerably expanded and enriched by concepts developed by Laban's colleagues and later students of human movement working within the Laban tradition. This method of movement study focuses on the interdependence of thinking, feeling, and action by developing awareness and activating the relationship between personal intention, attention, and action in all that we do and say. In the perspective of what we have investigated about the relationships between gesture and speech, gesture and thought, and gesture and emotional state and personality, we find the principles of this study perfectly mesh to our needs in synthesizing communicative gestures and acquiring motion qualities of communicative gestures.

A wide variety of researchers have applied the LMA theories in many movement-related fields such as acting, drama, choreography, psychology, ergonomics, anthropology, clinical and physical therapy, verbal and nonverbal communication and presentational skills, and management behavior [35, 11, 36, 79].

[1]LMA is not the same as Labanotation [60]. The former focuses on the movement qualities while the latter focuses on the structural aspects of movement and provides a means to record movement directions, places, positions, and involved body parts by means of symbols.

3.1 General Principles of the LMA

Moore and Yamamoto list five general principles that underlie Laban's conception of human movement [93]:

1. *Movement is a process of change.* Commonly, movement is defined as a change in place or position. That is, an action begins in one place and ends in another and, through the perception of this change, we know that movement has occurred. But while the difference between the beginning and ending locations of an action may be indicative of motion, movement itself is not a fixed position or even a change of positions. Rather, movement is the *process* of the changing. Furthermore, human movement involves not merely a change in position, but also a change in the activation and involvement of the body and in the quantity and quality of energy necessary to affect the motion. In other words, human movement is a fluid, dynamic transiency of simultaneous changes in spatial positioning, body activation, and energy usage.

2. *The change is patterned and orderly.* At first glance the spatial pathways traced by a body in motion may appear random and disorderly. But closer study reveals that a series of natural sequences of movement exists. Laws of sequencing, the alternating rhythms of stability and mobility and exertion and recuperation — all these provide a governing pattern and order that prevents movement from being chaotic.

3. *Human movement is intentional.* The human being moves to satisfy a need. Actions are guided and purposeful, and the intentions are made clear by the way in which the person moves. Moreover, the manner that the person moves allows an observer to penetrate the "inner world in which impulses continually surge and seek an outlet in doing ..." ([76], pp. 17). While individuals do show habitual predilections for certain effort configurations, human beings also possess the capacity to comprehend the nature of effort qualities and their patterning in dynamic sequences.

4. *The basic elements of human movement may be articulated and studied.* Through his scrutiny of human movement in a variety of contexts, Laban discovered basic elements of physical action that are common to all human motion (see Section 3.2).

5. *Movement must be approached at multiple levels if it is to be properly understood.* As noted above, movement is a dynamic, fluid process involving simultaneous changes in spatial positioning, body activation, and energy usage; movement study, as Laban envisioned it, should incorporate multiple levels of analysis. The analysis should consider not only *what the movement is made of* (the basic elements that comprise the action), but also *how it is put together* (the laws of sequencing and rhythmic patterns).

3.2 Basic Components of LMA

LMA is composed of five major components: Body, Space, Effort, Shape, and Relationship[2]. Together these components constitute a textual and symbolic language for describing movement. Body deals with which body parts move, where the movement initiates, and how the movement spreads through the body. Space describes how large the mover's kinesphere, and what crystalline form is being revealed by the spatial pathways of the movement. Shape describes the changing forms that the body makes in space, while Effort involves the "dynamic" qualities of the movement and the inner attitude towards using energy. Relationship describes modes of interaction with oneself, others, and the environment. Each individual has his/her own unique repertoire of and preferences for combinations of these basic elements, which can be sequenced, phrased, patterned, and orderly organized together in a particular personal, artistic, or cultural way. Our work focuses on the Effort and Shape components of LMA, because these two are the major direct specifications or indications of expressive human movements.

3.3 Effort and Shape

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. In LMA these extreme Effort Elements are seen as basic, "irreducible" qualities, meaning that they are the smallest units needed in describing

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

an observed movement. These eight Effort Elements are: Indirect/Direct, Light/Strong, Sustained/Sudden, and Free/Bound. The eight elements can be combined and sequenced for many variations of phrasings and expressions. Table 3.1 illustrates the motion factors, listing their opposing Effort Elements with textual descriptions and examples.

Space — attention to the surroundings	
Indirect	spiraling, deviating, flexible, wandering, multiple focus *examples*: waving away bugs, surveying a crowd of people, scanning a room for misplaced keys
Direct	straight, undeviating, channeled, single focus *examples*: threading a needle, pointing to a particular spot, describing the exact outline of an object
Weight — attitude to the movement impact	
Light	buoyant, weightless, 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, forceful, vigorous, 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	leisurely, lingering, indulging in time *examples*: stretching to yawn, stroking a pet
Sudden	hurried, urgent, quick, fleeting *examples*: swatting a fly, lunging to catch a ball, grabbing a child from the path of danger, making a snap move
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, rigid *examples*: moving in slow motion, tai chi, fighting back tears, carrying a cup of hot tea

Table 3.1: Motion Factors and Effort Elements ([31, 32])

The Shape component involves three distinct qualities of change in the form of movement: Shape Flow, Directional Movement, and Shaping. A Shape Flow attitude primarily reflects the mover's concern with the changing relationship among body parts.

These changes can be sensed as the increasing or decreasing volume of the body's form or a moving toward or away from the body center. Shape Flow can be seen from these two different perspectives: the first one emphasizes the torso, which can be said to Grow or Shrink. A continuous breathing pattern reveals changes in Shape Flow as seen from the torso perspective. The other perspective stresses the limbs, which are said to be Opening or Closing with respect to the horizontal axis. Shrinking from the cold or stretching to wake up would be characterized as having a Shape Flow quality.

While Shape Flow is mainly concerned with sensing the body's shape changes within itself, Directional Movement describes the mover's intent to bridge the action to a point in the environment. These movements can be simple spoke-like or arc-like actions to reach a direction or object, such as a reach to shake a hand or to touch an object or to move to a specific location.

Shaping Movement depicts the changes in movement form that demonstrate a carving or molding attitude as the body interacts with the environment. This form can be dictated by objects in space or simply created by the mover. An active adapting of the body shape in order to move through a crowd, or a gesture describing an elaborately carved sculpture might illustrate a Shaping mode.

Shape changes in movement can be described in terms of three dimensions: Horizontal, Vertical, and Sagittal. Each one of these dimensions is in fact associated with one of the three main dimensions (Length, Width, and Depth) as well as one of three main planes (Horizontal, Vertical, and Sagittal) related to the human body. Changes in Shape in the Horizontal dimension occur mainly in the side-open and side-across directions; as the movement becomes planar there would be more of a forward-backward component added to the primary side component. Changes in the Vertical dimension are manifested primarily in the upward-downward directions; the plane would add more sideward component to the up-down. Finally, changes in the Sagittal dimension are more evident in the body's depth or the forward-backward direction; planar movement would add an upward-downward component.

We note that while there is distinct vocabulary for each quality – Shape Flow, Directional Movement, and Shaping – in the various dimensions, we have merged these three concepts (using them interchangeably) and chosen to use the Shaping terminology.

The terms we are using to describe the opposing changes in these dimensions are Spreading and Enclosing, Rising and Sinking, Advancing and Retreating. It is important to point out that limbs and torso movements are not required to involve the same Shape qualities at a given time. In this way, Shape Flow functions as a breathing baseline to support Directional and Shaping movement of the limbs. In another example, a traffic officer might hold up one arm with a Directional reach, while the other arm gestures in a circular Shaping mode, and the head does small tilting Shape Flow actions to accompany the Shaping arm.

Another LMA concept is Reach Space in the Kinesphere (near, middle, and far). Our current approach regards Reach Space only from the perspective of the limbs in relation to the distance from the body center. Though this is a simplified view, it adds an important feature to the limb range of movement.

Shape changes can occur in affinity with corresponding Effort Elements [11, 79]. Table 3.2 shows the opposing attitudes towards Shape, some examples, and their affinities with Effort Elements.

Horizontal	
Spreading	affinity with Indirect (i.e., deviating, circling) *examples*: opening arms to embrace, sprawling in a chair, smoothing the wrinkles of a table cloth, a fisherman throwing out a net
Enclosing	affinity with Direct (i.e., undeviating, pointing) *examples*: clasping someone in a hug, crossing one's arms as when feeling cold
Vertical	
Rising	affinity with Light (decreasing pressure) *examples*: reaching for something in a high shelf, showing off with a pompous bearing, looking over the shoulder
Sinking	affinity with Strong (increasing pressure) *examples*: stamping the floor with indignation, pulling down a shade, a boxer ducking to avoid a punch
Sagittal	
Advancing	affinity with Sustained (i.e., decelerating) *examples*: reaching forward to shake hands, reaching forward to listen more carefully
Retreating	affinity with Sudden (i.e., accelerating) *examples*: darting back, avoiding a punch, pulling one's hand back from a hot stove, shocked by a sad or surprising news

Table 3.2: Shaping Dimensions and Affinities

Chapter 4

Gesture Synthesis

Generating communicative gestures may involve synthesizing information from multiple channels such as facial expression, eye gaze, intonation, and muscle tension. Our current research focuses on the gesture synthesis that chiefly involves limb and torso movements. We use EMOTE, a 3D animation control module for expressive limb and torso movements, to create communicative gestures that convey naturalness and expressiveness. EMOTE starts with basic movements specified through key time and pose information. We could also start with motion defined by some other methods, for example, keyframe data, a procedurally generated motion, motion capture data, or a gesture in a motion library, and then extract the necessary information. More importantly, EMOTE provides a flexible and powerful tool that allows the user to specify motion qualities in an intuitive way. Gestures exist not just because they have underlying movements but also because they have some distinctiveness in their motion qualities. Different motion qualities distributed over the same underlying motion may convey different meaning and therefore produce different gestures.

4.1 Expressive Limbs

EMOTE uses a limb model with a 1 degree-of-freedom (DOF) elbow/knee joint and spherical (3 DOF) shoulder/pelvis and wrist/ankle joints, as shown in Figure 4.1 [1]. The

[1] The human model is fully articulated, and commercially available through Unigraphics Solutions Inc. [44]. For more information about the model, check the web site http://www.eai.com/products/jack/

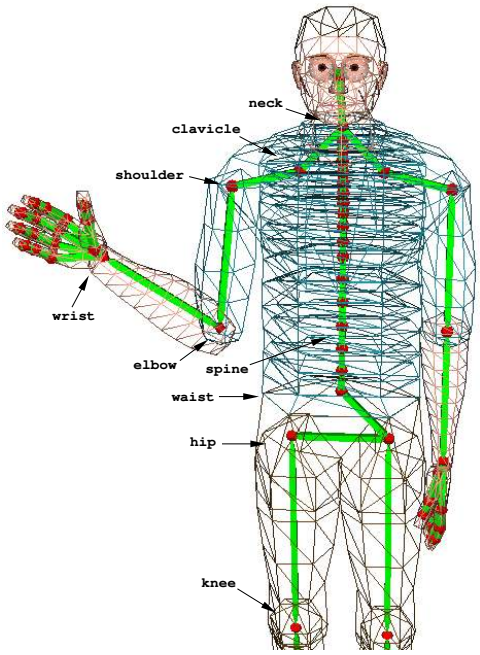

Figure 4.1: Human model

underlying key poses are defined as end-effector positions of the bases of the wrist/ankle (we call an end-effector key pose a *keypoint*). Keypoints can be defined as being *global* or *local*. Specifically for the arms, *local* keypoints are defined relative to the human's shoulders. *Global* keypoints, on the other hand, establish a constraint relative to the global environment. Keypoints can also be classified into *Goal* and *Via* points. *Goal* points define a general movement path; the hand follows this path, stopping at each *Goal* point. *Via* points direct the motion between keyframes without pausing. For instance, a *Via* point might be used to generate a semi-circular path between two *Goal* points.

The determination of arm/leg posture given a 3D keypoint is under-specified, however. A simple physical interpretation is based on the observation that if the hand is held fixed, the elbow is still free to swivel about a circular arc whose normal vector is parallel to the shoulder-to-wrist axis. Tolani [126] uses the *swivel angle* to solve this problem. Figure 4.2 shows the basic idea about how to use swivel angle to constrain the arm posture [2]. In the figure, **S**, **E**, and **W** define the positions of the shoulder, elbow, and the goal location of

[2]The rationale applies to the ankle similarly.

the wrist, respectively. **S** is chosen as the origin of the coordinate system. The normal vector $\hat{\mathbf{n}}$ of plane **P** that contains the circular arc can be computed as:

$$\hat{\mathbf{n}} = \frac{\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|} \tag{4.1}$$

The two unit vectors $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ that form a local coordinate system for plane **P** are given by

$$\begin{cases} \hat{\mathbf{u}} &= \frac{\hat{\mathbf{a}}-\hat{\mathbf{a}}\cdot\hat{\mathbf{n}}}{\|\hat{\mathbf{a}}-\hat{\mathbf{a}}\cdot\hat{\mathbf{n}}\|} \\ \hat{\mathbf{v}} &= \hat{\mathbf{n}} \times \hat{\mathbf{u}} \end{cases} \tag{4.2}$$

where $\hat{\mathbf{a}}$ is an arbitrary axis selected by the user [3]. The center of the circle $\hat{\mathbf{c}}$ and its radius **R** can be computed from simple trigonometry

$$\begin{cases} \hat{\mathbf{c}} &= \cos(\alpha)\|\hat{\mathbf{e}}\|\hat{\mathbf{n}} \\ R &= \sin(\alpha)\|\hat{\mathbf{e}}\| \\ \cos(\alpha) &= \frac{\hat{\mathbf{w}}\cdot\hat{\mathbf{w}}+\hat{\mathbf{e}}\cdot\hat{\mathbf{e}}-\hat{\mathbf{e}}_w\cdot\hat{\mathbf{e}}_w}{2\|\hat{\mathbf{w}}\|\|\hat{\mathbf{e}}\|} \\ \sin(\alpha) &= \sqrt{1-\cos(\alpha)^2} \end{cases} \tag{4.3}$$

Finally the elbow position and therefore the arm posture can be uniquely specified by

$$\mathbf{e}(\theta) = \hat{\mathbf{c}} + R\left(\cos\left(\theta\right)\hat{\mathbf{u}} + \sin\left(\theta\right)\hat{\mathbf{v}}\right)$$

Given a goal specified by three-dimensional position coordinates and an elbow/knee swivel angle, an analytical inverse kinematics algorithm (IKAN) [126, 127] computes the shoulder/pelvis and elbow/knee rotations. Wrist/ankle rotations are determined according to Effort settings [31]. Reflecting Effort and Shape definitions provided by the LMA system, Shape parameters are used to modify the keypoints that specify limb movements, while Effort parameters affect the execution of those movements resulting from the modified keypoints.

4.1.1 Applying Shape to Limb Movements

As described in Chapter 3, Shape comprises four parameters: Horizontal, Vertical, Sagittal and Flow (or Reach Space). A Shape Flow primarily reflects the mover's concern with the changing relationship among body parts. These changes can be sensed as the increasing or

[3] In EMOTE system we chose $\hat{\mathbf{a}}$ such that it is lying in the plane that contains the circular arc, and pointing downward.

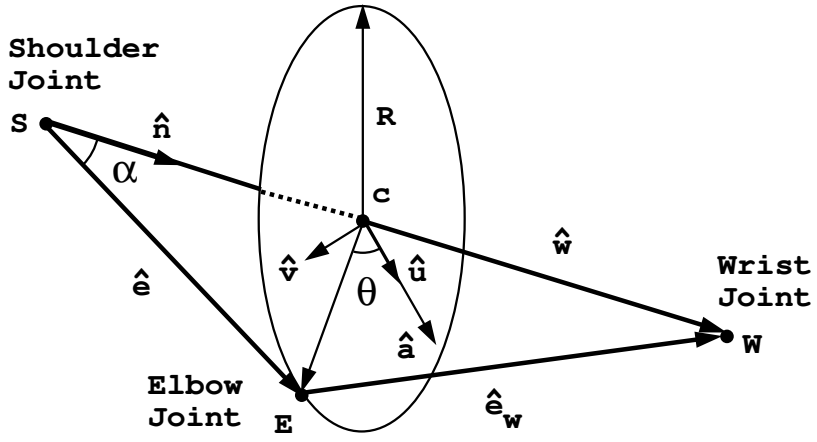

Figure 4.2: The arm posture constrained by the swivel angle (After Tolani [126])

decreasing volume of the body's form or a moving toward or away from the body center. Shape changes in movement can be additionally described in terms of three dimensions: Horizontal, Vertical, and Sagittal. Each one of these dimensions is in fact associated with one of the three main dimensions (Length, Width, and Depth) as well as one of three main planes (Horizontal, Vertical, and Sagittal) related to the human body. We describe mathematically how they work when applied to the underlying keypoints in the following.

4.1.1.1 Keypoints Modified by Horizontal, Vertical and Sagittal Parameters

In order to simulate volume-like changes in the movement, we associate Shape changes more with planar action than with strictly dimensional movement. Presently we expand or contract key points along ellipses oriented according to the Shape parameter values.

For a particular keypoint, let the variables *ver*, *hor*, and *sag* in the interval $[-1, +1]$ represent the parameters corresponding to the Horizontal, Vertical, and Sagittal dimensions, respectively. We define two constants *abratio* (always > 1) and $maxd\theta$ [4]. For each one of the above dimensions, we find an ellipse containing the keypoint and lying in a plane parallel to the plane associated with the dimension. The center of the ellipse is the projection of the shoulder/pelvis joint position on that plane (see the top figure in

[4]These constants can be changed by the user through a provided Graphical User Interface (GUI). The default values are 2.5 for *abratio* and $\frac{\pi}{6}$ for $maxd\theta$.

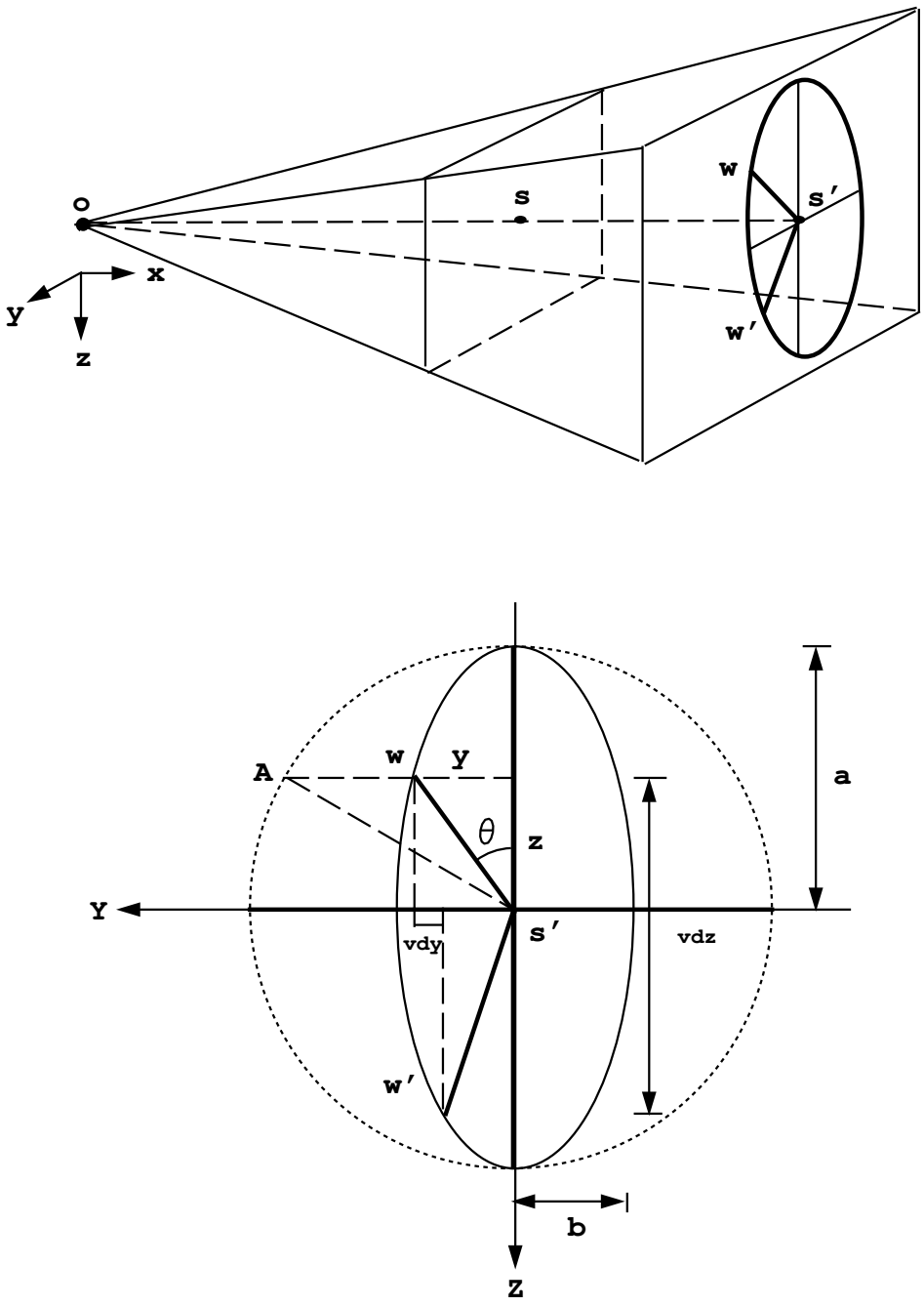

Figure 4.3: Using Vertical parameter to modify keypoints (Top: the shoulder projection to the parallel Y-Z plane, Bottom: the ellipse lying on the Y-Z plane.)

Figure 4.3). The major axis of the ellipse is parallel to the direction mostly affected by changes in that dimension and its minor axis is parallel to the other direction affected by such changes. The quotient between its major radius **a** and its minor radius **b** is *abratio*. We find the contributions of that dimension to the modified keypoint by rotating the keypoint by $\mathbf{d}\theta$, a fraction of $maxd\theta$ determined by the numeric parameter associated with the dimension in consideration. Figure 4.3 and 4.4 illustrate how we calculate **vdy** and **vdz**, the contributions of the Vertical parameter **ver** to a particular keypoint.

Let $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ be the original coordinate of the keypoint $\mathbf{w}$. We find θ such that

$$\theta = atan(\frac{y}{-z} * abratio) \tag{4.4}$$

The calculated θ is in the range $(-\frac{\pi}{2}, \frac{\pi}{2})$. We do the simple transformation

$$\begin{cases} \theta = \theta + \pi & if \quad -z < 0 \\ \theta = \theta + 2\pi & if \quad \theta < 0 \end{cases} \tag{4.5}$$

to make it lie in the range $[0, 2\pi)$.

The major axis **a** of the ellipse is calculated by the following equation:

$$a = \frac{-z}{sin(\theta)} \tag{4.6}$$

The angle φ formed by the rotated keypoint and the major axis of the ellipse is given by (see Figure 4.4):

$$\varphi = \begin{cases} 0 & ver = 0 \\ min(\theta - ver * maxd\theta, \pi) & ver < 0, 0 < \theta \leq \pi \\ max(\theta + ver * maxd\theta, \pi) & ver < 0, \pi < \theta \leq 2\pi \\ max(\theta - ver * maxd\theta, 0) & ver > 0, 0 < \theta \leq \pi \\ min(\theta + ver * maxd\theta, 2\pi) & ver > 0, \pi < \theta \leq 2\pi \end{cases} \tag{4.7}$$

Finally, the contributions **vdy** and **vdz** are calculated as follows:

$$\begin{cases} vdz &= -(a * cos(\varphi)) - z \\ vdy &= (a * \frac{1}{abratio} * sin(\varphi)) - y \end{cases} \tag{4.8}$$

Similarly we find the Horizontal contribution $(\mathbf{hdy}, \mathbf{hdx})$ and the Sagittal contribution $(\mathbf{sdx}, \mathbf{sdz})$. We compute the new keypoint $\mathbf{w}'$ (whose coordinate is $(\mathbf{x}', \mathbf{y}', \mathbf{z}')$) by

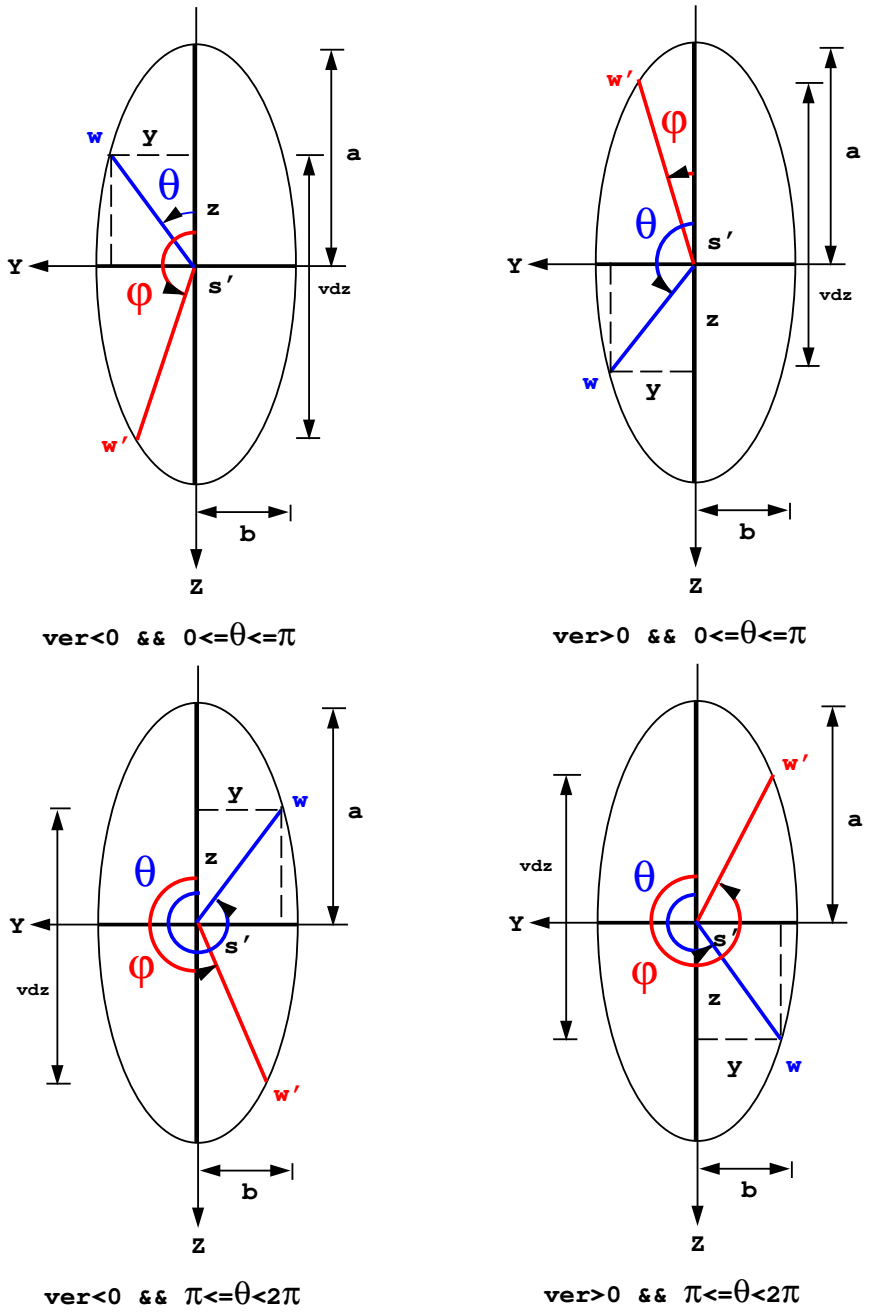

Figure 4.4: Rotation angles affected by the Vertical parameter

superimposing these contributions on the original keypoint:

$$\begin{cases} x' &= x + hdx + sdx \\ y' &= y + vdy + hdy \\ z' &= z + vdz + sdz \end{cases} \tag{4.9}$$

4.1.1.2 Keypoints Modified by a Kinespheric Reach Space Parameter

Let us now consider how the Kinespheric Reach Space parameter (also called Flow parameter) affects a particular keypoint. As stated before, when considered from the perspective of the limbs, Reach Space design describes the limb relationship with the body as it moves toward or away from the body center. Therefore, our Shape model modifies a particular keypoint by moving it along the direction that passes through the keypoint and the center of mass (henceforth, COM) of the human figure. As shown in Figure 4.5, vectors $\hat{\mathbf{w}}$ and $\hat{\mathbf{c}}$ represent the positions of the wrist and the COM in the global coordinate system originated at $\mathbf{o}$, while vector $\hat{\mathbf{t}}$ represents the position of the wrist in the local coordinate system originated at the shoulder $\mathbf{s}$. Suppose the matrix of the shoulder is $\mathbf{M}_s^g$ in the global coordinate system, the matrix of the wrist is $\mathbf{M}_w^l$ in the local coordinate system and $\mathbf{M}_w^g$ in global coordinate system. We can compute the vector $\hat{\mathbf{w}}$ in the following way:

$$\begin{cases} \mathbf{M}_w^g &= \mathbf{M}_w^l * \mathbf{M}_s^g \\ \hat{\mathbf{w}} &= \begin{bmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} & \mathbf{1} \end{bmatrix} \mathbf{M}_w^g \end{cases} \tag{4.10}$$

where $*$ represents the operation of homogeneous transformation multiplication and $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$ are the unit vectors of the global transformation system.

We use Reach Space parameter **flo** to calculate the amount by which the keypoint is moved toward or away from the center of mass. In Figure 4.5, $\mathbf{w}$ is the original position of the wrist. It is first moved by the Horizontal, Vertical, and Sagittal parameter to $\mathbf{w}'$, and then modified further by the Reach Space parameter to $\mathbf{w}''$. Let $\hat{\mathbf{v}}$ denote the vector $\widehat{\mathbf{W}'\mathbf{C}}$, and let vectors $\hat{\mathbf{u}}$ and $\hat{\mathbf{s}}$ denote the new position in the local transformation system with respect to the shoulder $\mathbf{s}$ and to the wrist $\mathbf{w}'$, respectively. The two vectors are given by

$$\begin{cases} \hat{\mathbf{u}} &= \frac{\hat{\mathbf{v}}}{\|\hat{\mathbf{v}}\|}(-\mathbf{f}) \\ \hat{\mathbf{s}} &= \hat{\mathbf{t}} + \hat{\mathbf{u}} \end{cases} \tag{4.11}$$

where

$$\begin{cases} \hat{\mathbf{v}} &= \hat{\mathbf{c}} - \hat{\mathbf{w}} \\ \mathbf{f} &= \mathbf{flo} \cdot \mathbf{maxds} \end{cases} \tag{4.12}$$

The parameter **maxds** is a constant value which specifies the maximum incremental distance for keypoints towards the COM that can be affected by the Reach Space parameter. Note that the Reach Space modifier is considered after the keypoint has been modified according to its associated Horizontal, Vertical, and Sagittal parameters.

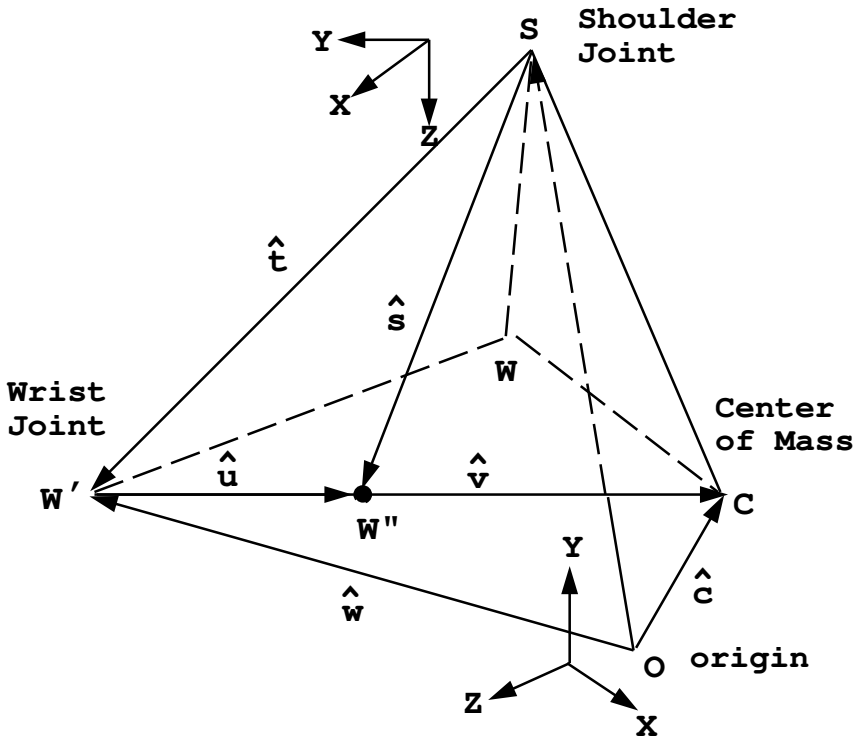

Figure 4.5: Using Flow parameter to modify keypoints

Additionally, when the achievement of the modified keypoint requires any of the angles beyond the human body limits, stored joint limits avoid unattainable configurations of the body. Furthermore, *global* keypoints are not affected by the Shape parameters as they establish a constraint relative to the environment instead of local joints.

4.1.2 Applying Effort to Limb Movements

Using high level qualitative Effort parameters for human animation control was first proposed by Badler [8] and later implemented by Chi [31]. Our EMOTE system is based on Chi's Effort module. For the sake of consistency and continuity, we briefly describe her methods that are relevant to our discussion. Refer to her thesis for more technical materials.

The key component of the Effort module is to translate the qualitative Effort parameters into a set of low-level quantitative parameters that are directly related to the control of the characteristics of the movement. There are three types of low-level movement parameters: (1) parameters that affect the limb trajectory; (2) parameters that affect timing, and (3) flourishes that add to the expressiveness of the movement.

The trajectory parameters include *path curvature*, which determines the straightness or roundness of the path segments between keypoints, and *interpolation space*, which defines the space in which the interpolation is performed. The path curvature is controlled through the tension parameter introduced by Kochanek and Bartels for interpolating splines [69]. There are three different kinds of interpolation space: end-effector position, joint angle, and elbow/knee position. Which interpolation space to use is determined by Effort settings. The default interpolation space is end-effector position. Free movements use angular interpolation to achieve a less path-driven and less controlled movement. Indirect movements tend to be driven by the elbow/knee, and thus are interpolated in elbow/knee position space.

Parameterized timing control is achieved by using a variation of the double interpolant method introduced by Steketee and Badler [121]. The interpolating splines that define the trajectory compute values between keypoints using an interpolation parameter that varies from 0 to 1 over the interval from keypoint i to keypoint $i + 1$ [69]. A frame number-to-time function is defined and can be parameterized by a set of low-level variables, such as number of frames between keypoints, inflection time, time exponent, start velocity and end velocity, to achieve various timing effects. Flourishes are miscellaneous parameters, such as squash and stretch, wrist bend, arm twist, that add to the expressiveness of the movements.

4.2 Expressive Torso

The underlying key poses of the torso involve, in fact, the neck joint, the spine, the pelvis, and the two clavicle joints. The neck has 3 DOF, the spine has 17 joints with 3 DOF each, the pelvis has 3 DOF and each clavicle has 2 DOF. A key pose consists of neck, pelvis, and clavicle angles, and spine configuration [44]. When, for a particular keyframe, no pose information is provided, the system assumes a neutral posture, where all the angles are 0.

Figure 4.6: Expressive torso examples (left: Advancing and Rising, right: Enclosing and Retreating)

4.2.1 Applying Shape to Torso Movements

The association of Shape and body parts is based on the suitability of each body part in producing changes in the form of the body in given directions (upward or downward, sideways-open or sideways-across, and forward or backward). Thus, the upward/downward direction is associated with the neck and the spine; the sideways direction is associated with the clavicles, and the forward/backward direction is associated with the pelvis and the hips. Therefore, changes in Horizontal dimension, which occur mainly in the sideways

direction but also have a forward/backward component as the movement becomes planar, affect mostly the angles of the clavicles but also slightly alter the pelvis rotations. Changes in Vertical dimension, which occur mainly in the upward/downward direction but also have a sideways component in planar movement, affect mostly the angles of the neck and the spine but also change clavicle angles. Finally, changes in Sagittal dimension, which are more evident in the forward/backward direction but also involve an upward/downward component in planar movement, mainly affect the pelvis and hip rotations but also change the angles of the neck and spine.

The Shape model was designed considering the available control of the articulated figure model [44]. The present approach adjusts spine, pelvis, hip, and clavicle angles to approximate Shape volume changes. Fig. 4.6 illustrates two examples of the expressive torso model. Fig. 4.7 shows a sample keypoint file defining Effort and Shape parameters.

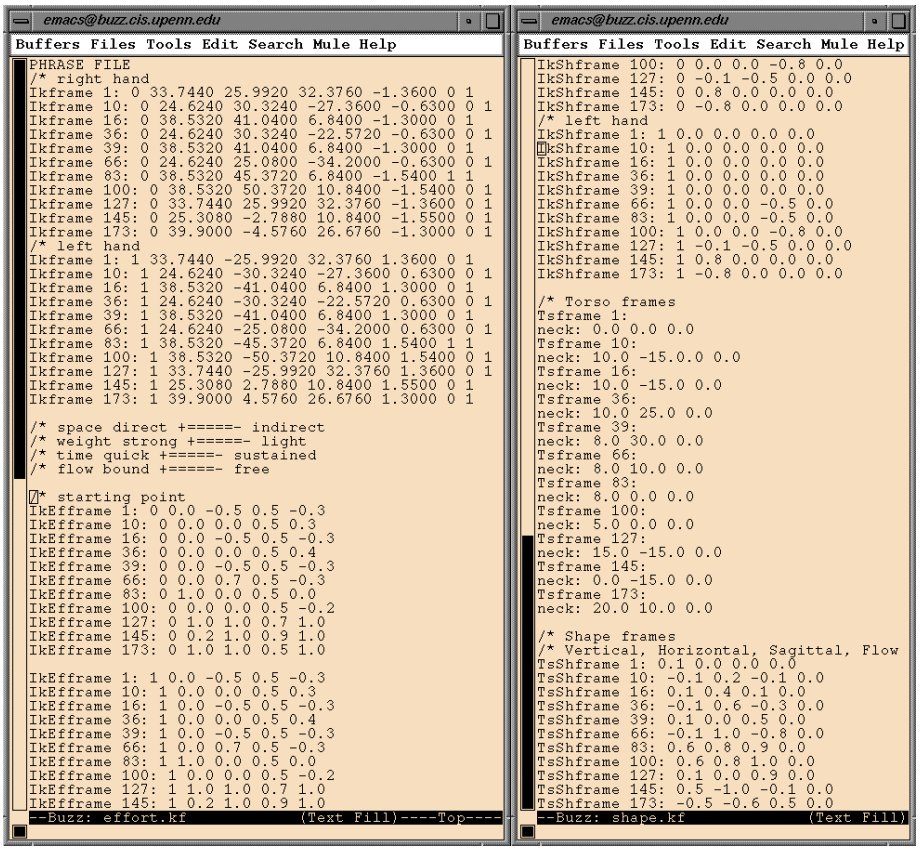

Figure 4.7: A sample keypoint file defining Effort and Shape parameters

4.3 Animation Examples

Using the EMOTE system, we have done some experiments and created (1) a virtual actor, (2) a virtual ASL signer, (3) a virtual salesperson, and (4) a virtual tour guide. In this section, we use the virtual actor animation as an example to demonstrate the power and flexibility of the EMOTE system in synthesizing gestures, in particular, we focus on how to interactively add or adjust Effort and Shape parameters to accomplish the improvement and variations in motion qualities.

In the example, the virtual actor performs a line from Shakespeare [5]. The original performance includes three opening gestures. The first two are Bound and Sudden, while the third is Free, Sudden, and Strong. This is followed by a Free and Sudden lifting of the arms, ending in a Strong, Sudden and Direct emphatic end. To demonstrate the usefulness of motion quality, we first define the basic upper body movement as a sequence of keypoints and a simple linear interpolation is employed to generate a neutral animation without Effort or Shape settings involved. Then we apply appropriate Effort and Shape parameters to mimic the original performance. Applying Effort and Shape parameters can be done easily and interactively, taking advantage of the graphical user interfaces provided by the EMOTE system. For example, instead of using Sudden, Strong and Bound qualities, users can simply move the sliding bars towards the reverse extremes to make the movements considerably more Sustained, Light and Free. In such a case, a dramatically different performance from the original one is produced. If an Enclosing Shape parameter is applied, a more confined gesture will be generated. Similarly, if a Rising and a Spreading Shape parameter are used instead, a bigger and more opening gesture will be produced. Finally, the experiments also demonstrate that the torso plays an important role in life-like animated movements. If we keep the original Effort and Shape settings for the limbs, but remove all Shape specifications for the torso, the animations lose conviction and naturalness. Figures 4.8–4.13 demonstrate the sample performance and its variations. The animations are recorded in AVI files, which can be found in the CD-ROM attached to this document (also available at http://www.cis.upenn.edu/~lwzhao/thesis).

[5] The line is "Love me, why? It must be requited" from the play *Much Ado About Nothing.*

Figure 4.8. An actual actor

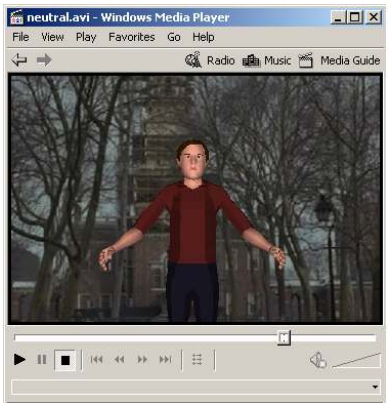

Figure 4.9. Neutral settings

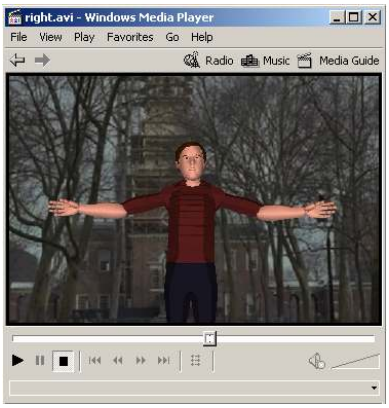

Figure 4.10. Right settings

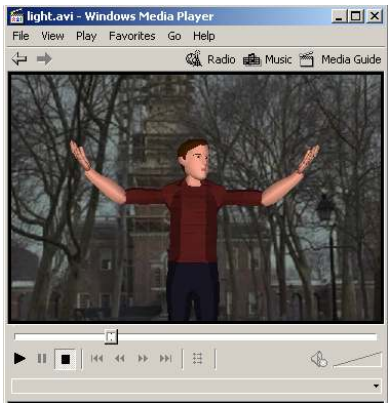

Figure 4.11. Light variation

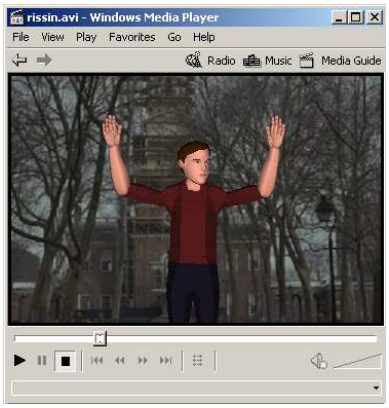

Figure 4.12. Rising variation

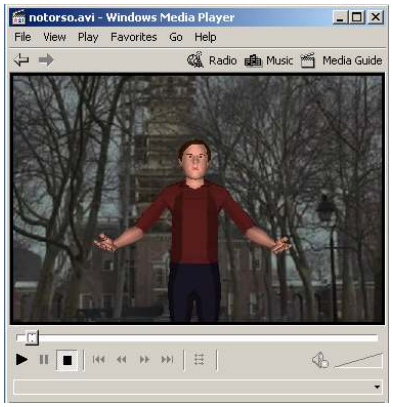

Figure 4.13. Without torso

4.4 Agent Model and Communicative Gesture Performance

We commonly use actions and non-verbal communicative movements to try to infer affective states, attitudes, and ultimately intents in the person we are observing. Our EMOTE model has a demonstrated capability of producing a wide range of expressive movements on a fully articulated human body. We believe that connecting EMOTE parameters to the agent model may therefore link personality and affective state with appropriate and communicative gesture performance.

Based on a Parameterized Action Representation (PAR), our agent model includes explicit slots for manner, role, culture, and capabilities as well as rules and standing orders linking PAR execution to specific conditions in the world [6, 4, 19]. Detailed description of the PAR components is too lengthy for this document, instead we focus on one aspect of the agent model that is well activated by the EMOTE model — the motion *manner*.

Motion manner describes the way an agent carries out an action. Although how the action is carried out also depends on the agent's skills and personality, motion manner stresses the specifications of the characteristics being used in carrying out a *specific* action. For examples, "open the door quickly," "move the vase carefully," and "walk along the shore leisurely." These manner terms can be transformed into Effort and Shape parameters that affect low-level motion generation [31, 139]. To see the importance of the motion manner component consider the differences between actions with essentially the same participants and path: ease, slide, push, tap, shove, wedge, force and slam. All vary in when and how much force is applied. The motion of the object involved is clearly affected differently, but so is the agent's movement. The general form of the action is stored as key poses, constraints, or even captured motion in a PAR, but the actual performance is mediated through the chosen EMOTE settings.

Modifying motion manners by changing Effort and Shape parameters is demonstrated in the following MPEG movies (also available at http://www.cis.upenn.edu/~lwzhao/thesis). The agent system has an incorporated natural language interface where the user can dynamically direct and refine the agent's behavior by issuing directions and instructions. Connecting the EMOTE parameters with the agent model enables us to instruct the agent to generate movements with appropriate manners from linguistic adverb constructs.

Figure 4.14. Touch

Figure 4.15. Delicate touch

Figure 4.16. Force touch

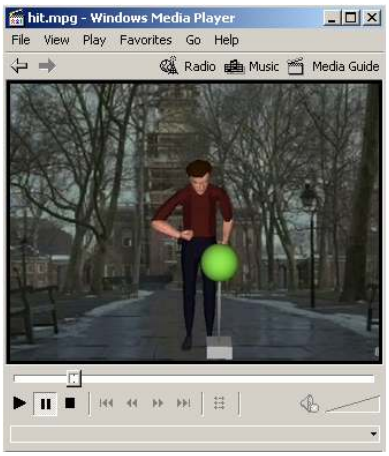

Figure 4.17. Hit

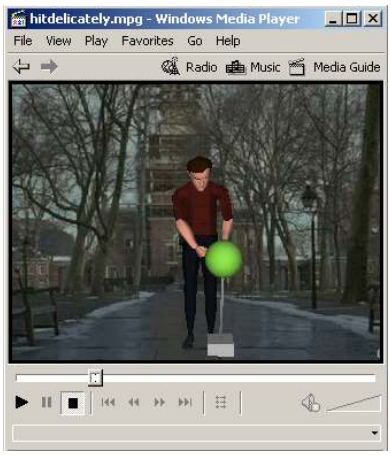

Figure 4.18. Delicate hit

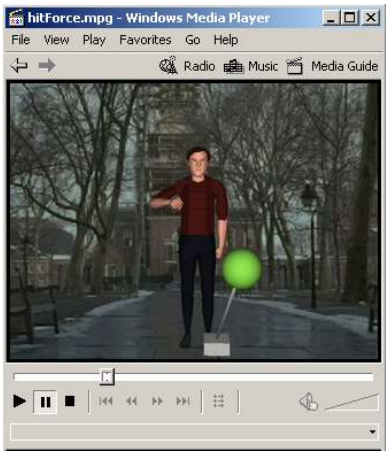

Figure 4.19. Force hit

4.5 Applying EMOTE Parameters to Motion Capture

Although the EMOTE system is capable of procedurally generating a great variety of expressive and natural movements, it suffers from three drawbacks: (1) it requires the manual specification of key points for constructing the underlying movements; (2) the current human model does not allow us to deform the body properly; and (3) an external LMA notator is needed to provide appropriate motion qualities in terms of Effort and Shape parameters. In this section, we attack the first problem by connecting a motion capture system with the EMOTE system and calculate the key points as well as other key parameters required by the IK module automatically. In the next section, we implement an EMOTE plug-in in Alias|Wavefront's Maya 3.0 to partially solve the second problem. The automatic acquisition of EMOTE Effort parameters is addressed in the next two chapters, one using motion capture data and the other using video capture data, respectively.

We use the methods described in [18] to solve the problems of motion calibration and motion retargeting. The important parameter that is still missing in the motion data is the swivel angle. The angle is required in order to uniquely define a limb posture. What we have in the data is only the 3D positions and orientations of the elbow/ankle. Given the coordinates of the end-effector, we can in fact compute the corresponding swivel angle that gives us the elbow/ankle position which is closest to the coordinates. The algorithm is shown below.

Suppose the origin of the coordinate system is located at $\mathbf{c}$, and the actual position of the elbow given by motion capture is $\mathbf{P}_e$. The elbow position on the circular arc (swept out by swiveling around the shoulder-to-wrist axis) is represented as $\mathbf{P}_\theta$. We chose subscript θ to show that the position is subject to change according to the value of the swivel angle θ, see Figure 4.20. To make the arm movement fit the motion capture data as much as possible, we need to find the shortest distance from $\mathbf{P}_\theta$ to $\mathbf{P}_e$, That means we need to find a swivel angle θ that minimizes $\|\mathbf{P}_e - \mathbf{P}_\theta\|$.

Let us define two vectors $\hat{\mathbf{P}}$ and $\tilde{\mathbf{P}}$ such that

$$\begin{cases} \hat{\mathbf{P}} = \mathbf{P}_e - \mathbf{c} \\ \tilde{\mathbf{P}} = \hat{\mathbf{P}} - (\hat{\mathbf{P}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} \end{cases} \tag{4.13}$$

The angle that minimizes $\|\mathbf{P}_e - \mathbf{P}_\theta\|$ is the angle between the unit vector $\hat{\mathbf{u}}$ and vector $\tilde{\mathbf{P}}$,

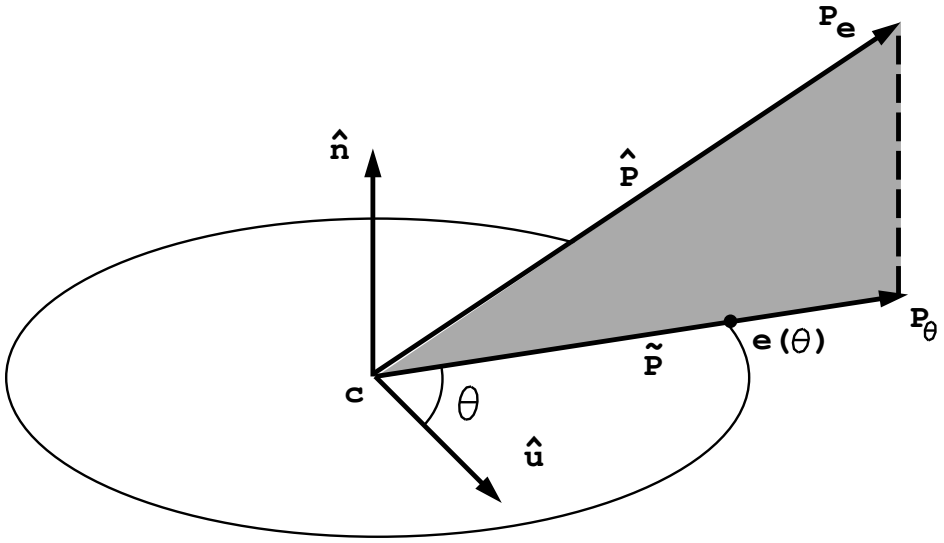

Figure 4.20: Find the closest possible elbow position to a motion captured position

and it satisfies

$$\begin{cases} \sin(\theta) = \frac{\left\|\tilde{\mathbf{P}} \times \hat{\mathbf{u}}\right\|}{\left\|\tilde{\mathbf{P}}\right\|} \\ \cos(\theta) = \frac{\tilde{\mathbf{P}} \cdot \hat{\mathbf{u}}}{\left\|\tilde{\mathbf{P}}\right\|} \end{cases} \tag{4.14}$$

Thus, we can calculate the angle as follows:

$$\theta = \frac{\sin(\theta)}{\cos(\theta)} = atan2(\left\|\tilde{\mathbf{P}} \times \hat{\mathbf{u}}\right\|, \tilde{\mathbf{P}} \cdot \hat{\mathbf{u}})$$

By extracting keypoints from motion capture data and then varying Effort and Shape parameters, we have achieved interesting variations in movements. For example, we collected 3D upper-body motion capture data of a person throwing a ball, then we extracted every ten frames and used them as direct input to the EMOTE system. Using the same keypoint input, we generated three dramatically different motions by merely picking different Effort and Shape parameter settings.

4.6 Applying EMOTE to Deformable Human Models

We chose a commercially available package, Maya 3.0 from Alias|Wavefront^{TM}, as the visualization environment. Maya is used for modeling, animation, rendering and special effects applications and has been very popular in broadcasting, film making, multimedia and game development. It features a full range of deformation functions such as bend, squash, twist and warp nonlinear deformers [1]. In our experiments we find that these deformers are very powerful and flexible to apply to human characters. Moreover, these deformers can leverage more subtle changes without destroying the motion qualities specified by EMOTE parameters. For example, by passing warping parameters generated by EMOTE parameters to the Trax Editor, we can explore more variations.

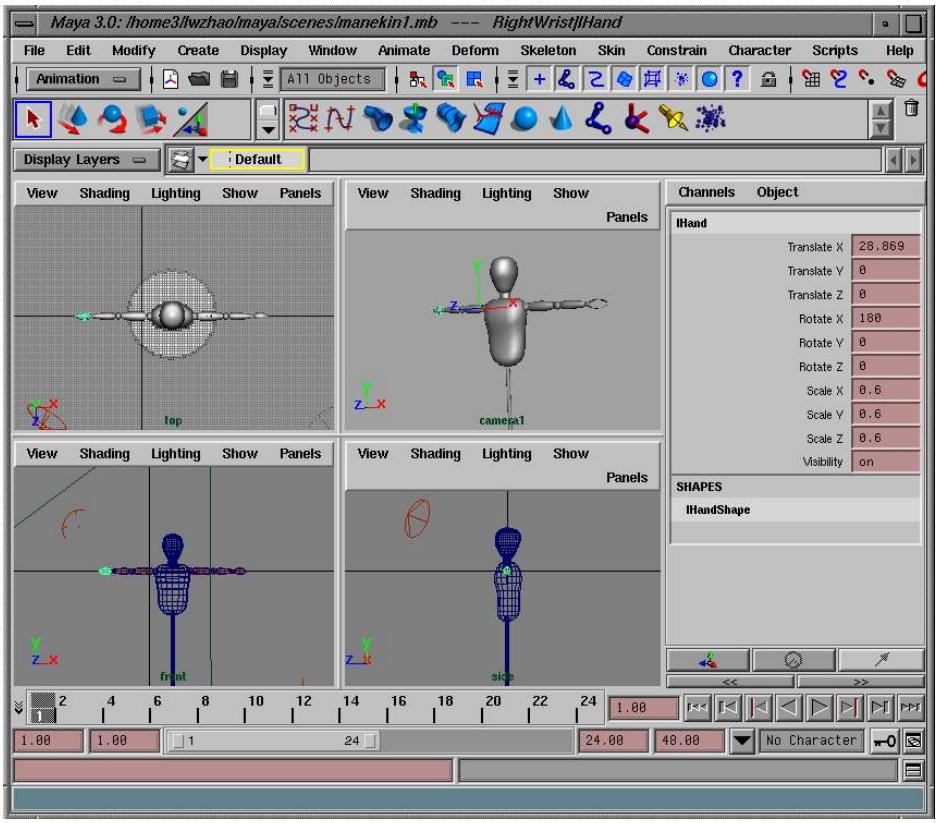

Figure 4.21: The Maya environment and the deformable human model

Furthermore, Maya has an open architecture and is customizable through two ways: the Maya Embedded Language (MEL) and the Maya Application Programmer Interface

(API). MEL is a powerful high-level command and scripting language that offers direct control over Maya's features, processes and workflows. It allows easy creation of custom graphical user interfaces and procedures to carry out modeling, animation, dynamics, and rendering tasks. The Maya API allows low-level direct access to internal data structures and therefore offers significantly fast execution of the tasks.

In the current implementation of the EMOTE plug-in [6], a simplified upper-body model was created based on a wooden mannequin (see Figure 4.21). A shoulder-elbow-wrist bone structure was inserted into each arm to help move the arm kinematically. The biceps contraction is simulated by inserting an ellipse sculpt deformer into the upper arm that moves arm vertices in their vicinities regularly. The limb volume is affected by a YZ-direction scaling factor, which is linearly correlated with factors such as the elbow angles and the Effort Weight parameters: the stronger a movement and the further bend the elbow, the larger the resulting scaling deformations. Figures 4.22, 4.23, and 4.25 display the limb deformation affected by no deformer, YZ-direction scaling deformer, and ellipsoid sculpting deformer, respectively. Note that these deformations are purely artistic and do not represent anatomical shapes, nor are they meant to be accurate. The deformation of the torso can be simulated using a similar but more complex deformer and it requires the deformation be affected by not only the Effort factors but also the Shape factors and the breath pattern as well.

At the heart of the EMOTE plug-in is a DG (Dependency Graph) node called *emoteNode*. It is written in C++ and Maya API and is responsible for the actual interpolation of animation parameters. Its accompanying MEL script plays two roles: one is an administrative role for loading, executing, and unloading the EMOTE plug-in module; the other role is communicative for supplying the motion data to the DG node, and manipulating the shape nodes that store the model's geometry. Maya's built-in inverse kinematic (IK) solver, rather than an independent IK solver, is used as it is faster and more consistent. The swivel angles output from EMOTE are connected to attributes that help define the twist of the Maya IK chain. Experiments carried out based on the simplified human model successfully generated the anticipated expressive arm gestures with deformation (see Figures 4.22–4.27).

[6] Bjoern Hartmann made a considerable contribution to the design and implementation of the plug-in.

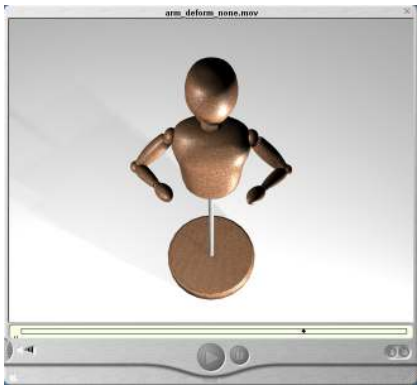

Figure 4.22. Limb deformation: none

Figure 4.23. Limb deformation: YZ-scaling

Figure 4.24. The virtual salesman in Maya

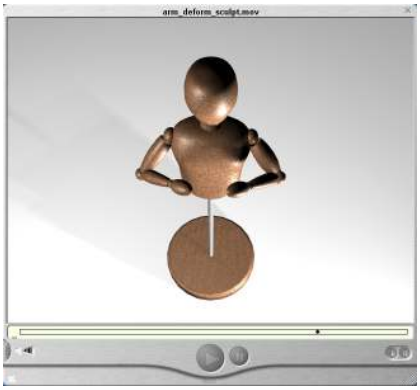

Figure 4.25. Limb deformation: Sculpting

Figure 4.26. A sample in EMOTE plug-in

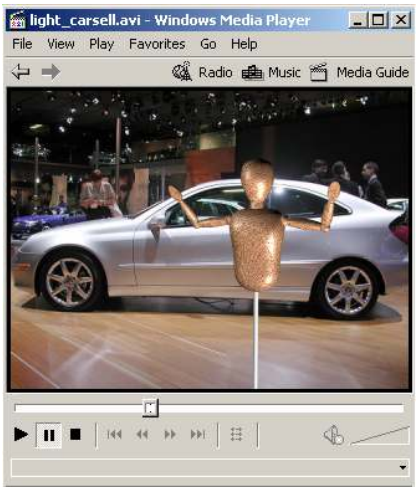

Figure 4.27. Light variation

4.7 Summary

In summary, EMOTE has four features which we believe are essential for creating communicative gestures that convey naturalness and expressiveness.

1. A given movement may have Effort and Shape parameters applied to it independent of its geometrical definition.

2. A movement's Effort and Shape parameters may be varied along distinct numerical scales.

3. Different Effort and Shape parameters may be specified for different parts of the body involved in the same movement. [7]

4. The Effort and Shape parameters may be phrased (coordinated) across a set of movements.

The underlying movements of a gesture are specified through key time and pose information defined for the torso and all the limbs. With the key pose information, the EMOTE parameters can then be applied to vary the original performance (property 1). Effort and Shape qualities are expressed using numeric parameters that can vary along distinct scales (property 2). Each Effort and Shape factor is associated with a scale ranging from -1 to $+1$. The extreme values in these scales correspond to the extreme attitudes of the corresponding factors. For example, a $+1$ value in Effort's Weight factor corresponds to a very Strong movement; a -1 value in Shape's Vertical dimension corresponds to a Sinking movement. Effort parameters are translated into low-level movement parameters, while Shape parameters are used to modify key pose information. By using combinations of one or many of the Effort and Shape parameters, we can search for the desired quality of a particular movement. EMOTE parameters create kinematic changes in the underlying movements. During gesture synthesis, EMOTE parameters can be applied directly based on parameter values dependent on a character's particular utterance, reactions, personality, or emotions.

[7] Some movements (for example, those in the virtual actor examples) are symmetric in both arms, however, the hit/touch and ball-throwing motions are not symmetric due to the different specifications of Effort and Shape parameters as well as the different key poses.

	Factors/ Dimensions	Right Arm	Left Arm	Right Leg	Left Leg	Torso
Effort	Space	yes	yes	yes	yes	no
	Weight	yes	yes	yes	yes	no
	Time	yes	yes	yes	yes	no
	Flow	yes	yes	yes	yes	no
Shape	Horizontal	yes	yes	yes	yes	yes
	Vertical	yes	yes	yes	yes	yes
	Sagittal	yes	yes	yes	yes	yes
	Reach Space	yes	yes	yes	yes	no

Table 4.1: Body parts and Effort and Shape Dimensions

EMOTE permits independent specification of Effort and Shape parameters for each part of the body (property 3). In its current implementation however, Effort parameters do not apply to torso movements. Although Shape parameters are effective in the specification of expressive torso movements, further investigation should be carried out to identify how Effort qualities are manifested in the torso. Table 4.1 summarizes which dimensions of Effort and Shape can be used to modify the movements of the different parts of the human body. Furthermore, our approach allows the specification of different sets of values for the Effort and Shape parameters across any series of keys that define the underlying motion (property 4). By property 3, this can be done separately for each part of the body.

Finally, we have developed the EMOTE system in several new ways:

- Connect EMOTE with an agent model so that agent affect and communicative needs can set appropriate EMOTE parameters for gesture performance. Currently the setting is achieved through a manually defined mapping table. Further investigation need to be carried out to build a more coherent and automatic mapping, particularly when a natural language interface bridging the natural language instructions and agent affect states is to be experimented based on the extended EMOTE system [140].

- The manual key specification is averted by connecting a motion capture system with EMOTE and automatically extracting the key point definitions.

- Experiment porting EMOTE to a commercially available visualization package (Alias|Wavefront Maya 3.0) where deformable human models are supported.

In the following two chapters, we shall investigate motion analysis techniques for extracting EMOTE Effort parameters from live input, both in 3D motion capture and 2D video data. Extracting EMOTE Shape parameters is beyond the scope of this work.

Chapter 5

Gesture Acquisition from Motion Capture

So far we have examined EMOTE as a motion quality generation system. Now we consider the inverse problem: deriving the Effort qualities from a live performance *automatically* via motion capture or video projections. The inverse problem is much harder to solve:

- Humans can synthesize multiple factors (such as speech intonation, muscle volume and tension, and facial expressions) from multiple channels simultaneously to analyze an action, however, a computerized system often limits available data to one or two channels.

- Mathematically, the inverse problem is often harder. Formulating a set of mathematical formulas and tweaking their parameters to generate the visual "impression" of some particular motion patterns is relatively easy, however, recovering a formula and its parameters that are functioning behind the patterns is more complicated, ambiguous, and sometimes even intractable.

The problem is challenging but not infeasible. Our approach is to build a computational model to simulate the LMA recognition and classification process. We first derive a set of relevant motion features based on the motion capture data or video projections and then use a three-layered feedforward neural network with a stochastic gradient descent backpropagation to estimate the relationships between the motion features and the motion

qualities in terms of Effort factors. Training and validation data sets created for this specific study with the assistance of professional LMA notators, are used as the ground truth for training, validating, and testing the neural networks.

In this chapter, we focus on gesture acquisition from motion capture data. The next chapter addresses the issues of learning motion qualities from video projections. The remainder of this chapter is organized as follows. Section 5.1 describes the motion capture system we use for acquiring the position and orientation data. Section 5.2 discusses a choreography plan we have designed to ensure the diversity of the baselines. Section 5.3 illustrates the smoothing methods we use to suppress noise in the captured data. Section 5.4 gives a high-level description about the relationships between the two major components of the system: feature extraction and quality recognition, which are covered in Section 5.5 and Section 5.7, respectively. Section 5.6 describes a simple but reliable segmentation method. Feature extraction and quality recognition are both segment-based.

5.1 Motion Capture System

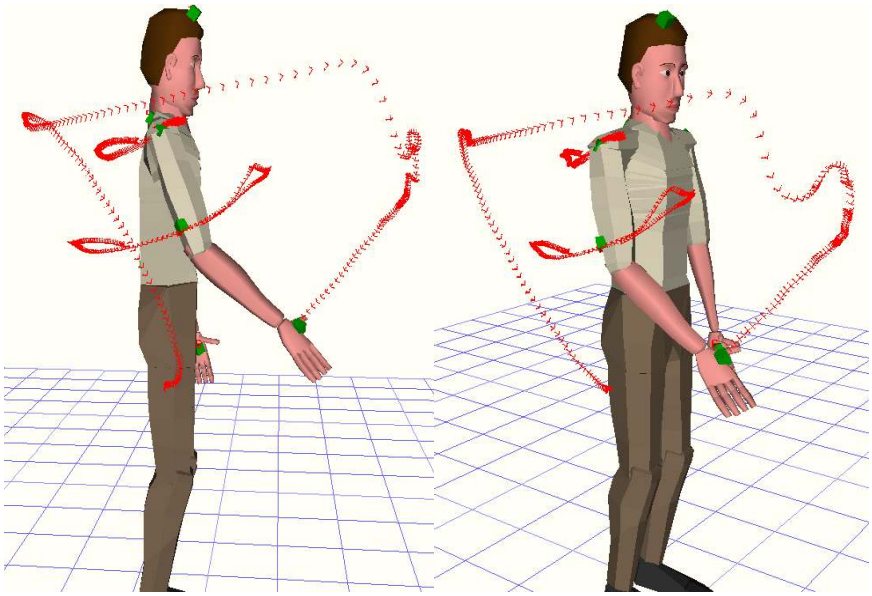

Figure 5.1: Trajectories of sensors (including the shoulder, the elbow, and the hand)

Our approach to estimating the Effort qualities from motion capture starts by using

the MotionStar system from Ascension Technology, which consists of one Extended Range Controller (ERC), one Extended Range Transmitter, and 12 Bird units, each controlling a single receiver (referred to as a sensor throughout the paper). Position and orientation sensors collect 6D motion trajectory data for the head, neck, sternum, back, and the right shoulder, right elbow, and right hand. The sampling frequency is 103.3 Hz. While the system is cost-effective and efficient in capturing data, it has a major drawback—it requires fastening the electro-magnetic sensors to the body. Also, preprocessing is necessary to filter noise (see Section 5.3). After the preprocessing, motion calibration and retargeting [18] are used to map all of the sensors to the human models in the *Jack* environment. Figure 5.1 shows the trajectories of sensors attached to the right arm in an action of throwing a ball.

#	direction	space	form	handshape
1	forward	mid-reach	spoke-like	point
2	downward	mid-reach	spoke-like	closed
3	upward	mid-reach	spoke-like	neutral
4	downward	near-reach	spoke-like	fist
5	horizontal	mid-left	arc-like	claw
6	horizontal	mid-right	circular	fist
7	diagonal	mid-left	arc-like	neutral
8	diagonal	mid-right	arc-like	open
9	sagittal	mid-reach	arc-like	claw
10	sagittal	mid-reach	spoke-like	neutral
11	backward	far-reach	circular	neutral
12	"glide"	far-reach	transverse	open

Table 5.1: Twelve simple and short movements

5.2 Choreography Plan

Because the whole inference system is trained and validated on the baselines of LMA notators, it is crucial to make the baseline motions as diversified as possible to cover different spatial directions/planes/dimensions and have different forms. With the help of two professional LMA notators, we carefully designed a "choreograph plan." Table 5.1 shows the twelve actions chosen and performed by our professional LMA notators. (Figure 5.2 illustrates these actions.) Each of the two LMA notators performs the twelve

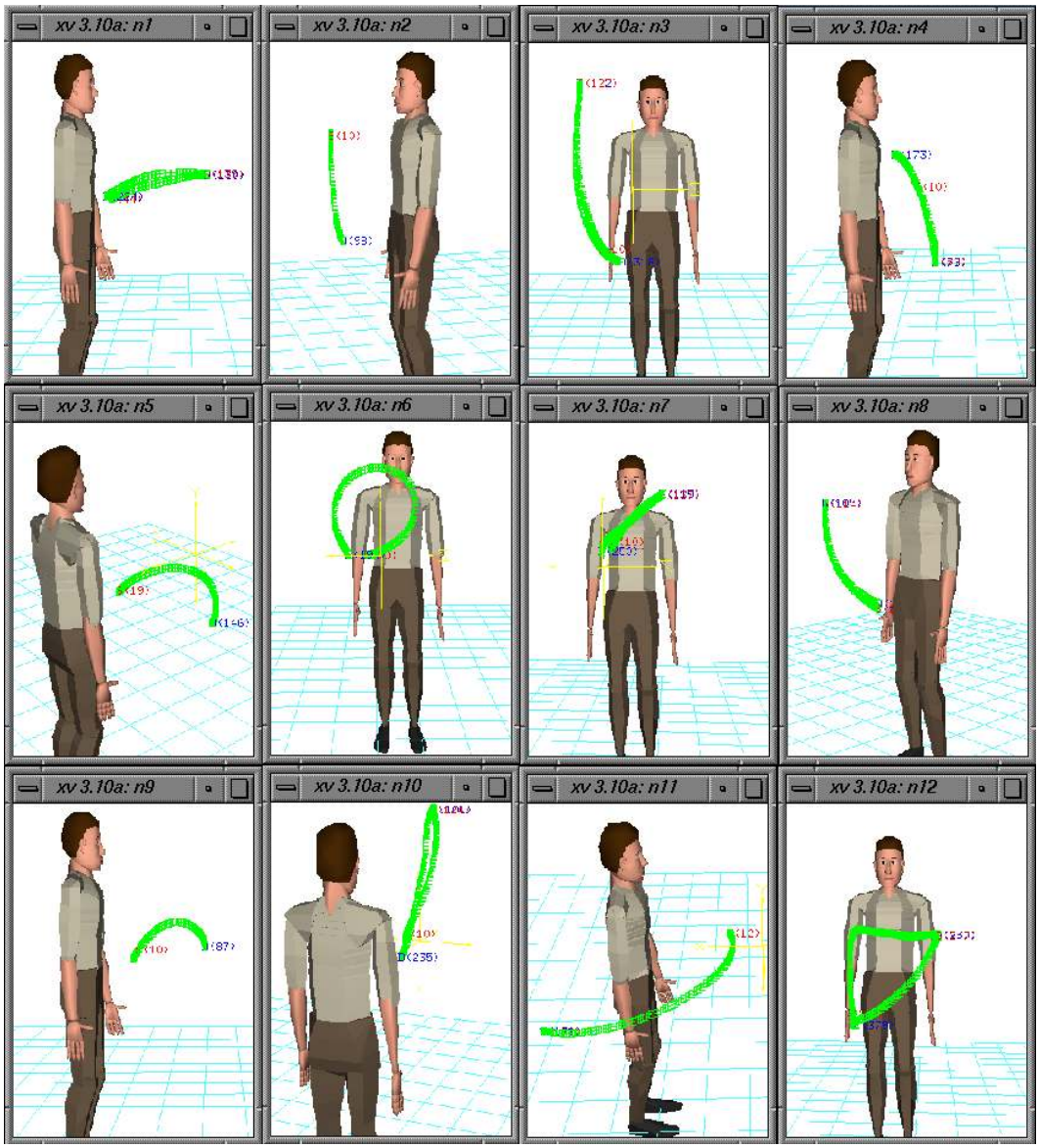

Figure 5.2: Motion plan

motions with one of the twelve Effort factors. In all we captured 288 samples with basic Effort elements. In addition, we captured 64 samples with Effort combinations. We focused on *simple* and *short* actions, usually consisting of one or two motion segments. Although we focused on the right arm only, one of our professional LMA notators is left-handed.

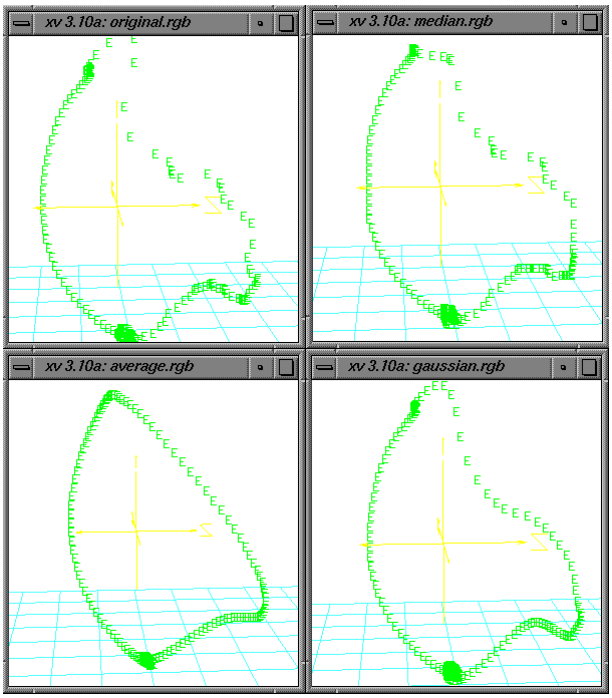

Figure 5.3: The comparison of smoothing algorithms (top-left: original trajectory, top-right: median smoothing, bottom-left: average smoothing, bottom-right: Gaussian smoothing)

5.3 Noise Filtering

As the motion capture system we use is an electro-magnetic tracker system, it is susceptible to interference from neighboring external sources such as metals and the power supply. A noise filtering process is employed to suppress irrelevant details in the captured data. We use a popular zero-phase digital filtering method [100] by processing the captured data in both the forward and reverse directions. Before filtering in the forward direction, it first generates a small extrapolation array at each end of the sequence. After filtering in

the forward direction, it reverses the filtered sequence and runs it back through the filter. Finally, the extrapolation arrays are discarded and the resulting sequence has precisely zero phase distortion. At the kernel of the filtering process is a neighborhood smoothing algorithm. We conducted experiments using Median, Average, and Gaussian smoothing. Figure 5.3 shows the filtering results for a sample action with these smoothing algorithms, respectively. Our experiments show that the Gaussian smoothes out the trajectories while maintaining the original profile. Therefore Gaussian is used as the smoothing filter in our system.

To assure that it is noise not motion quality features that are removed by the smoothing filters, we did experiments by comparing the original motions with the motions played along the smoothed trajectories. Our empirical study showed that a smoothing window size of 10 frames consistently gave us good results.

5.4 Hierarchical Abstraction

Hierarchical abstraction enables one to construct a model layer by layer in a constrained context and by a set of constrained elements and relations. Such an approach effectively reduces the search space of the interpretation using heuristics while still maintaining the essential relational structures. In the following, we give a mathematical description about how to abstract motion qualities from observation data hierarchically.

L : Total number of abstraction layers

N_l : Number of motion features at abstraction layer l $(1 \leq l \leq L)$

$f_{i,j}$: Motion feature i at abstraction layer j $(1 \leq i \leq N_j,\ 1 \leq j \leq L)$

S_j : A set of motion features at layer j or below

$F_{k,j}$: Abstraction function at layer j for computing feature k $(1 \leq k \leq N_j)$

where

$$\begin{cases} S_j & = \bigcup_{\substack{1 \leq k \leq j \\ 1 \leq i \leq N_k}} f_{i,k} \\ f_{k,j+1} & = F_{k,j+1}(S_j) \end{cases} \tag{5.1}$$

The abstraction function $F_{k,j}$ can be linear or nonlinear (i.e., polynomial, exponential, or logarithmic). It can be derived from some statistical, neural network models, or simply an empirical function based on experiments.

Eq. 5.1 specifies that the abstracting relations only take the lower level features as arguments, which means that the higher level motion factors are abstracted from the lower level motion factors. Specifically in our abstraction architecture, the lowest layer contains motion capture data while the highest layer contains motion factors specified by Effort qualities (Space, Weight, Time, and Flow). What are covered in between are two additional abstraction layers: one is the motion feature extraction layer (MFEL) and the other is the neural network abstraction layer (NNAL). The output from the MFEL are the direct input to the NNAL. As we go up in the architecture, low-level data are filtered away and high-level analytic data are filled in. Moreover, the complex motion factors encoded in the observation data, which are hard for the neural networks to discern directly, are computed mathematically. On the other hand, the coherent relationships among the extracted motion factors and Effort motion qualities, which cannot be directly computed due to the unknown mathematical equations, can be estimated by the neural networks. In the following, we describe MFEL first, then discuss the NNAL.

5.5 Motion Feature Extraction

The decision of which motion features to compute is mostly an art, since there are an unlimited number of possibilities and much more research is needed to determine which features are best for motion quality recognition. A variety of motion features had been employed [22, 116, 110, 120]. Features used by Rubine [116] to recognize simple pen gestures are mainly geometrically based. Segen and Kumar [120] use some local features such as "peaks" and "valleys" on the contour of the hand shape to help classify gestures. In our experiments, we have employed five categories of motion features that we believe are helpful in the acquisition process. Features are chosen according to the following criteria:

- **Efficiently computable**: each feature should be geometrically, algebraically, or incrementally computable, using only data available from the motion capture process.
- **Meaningful**: features should be correlated to the motion qualities.
- **Minimum coverage**: there should be sufficient features to capture and differentiate the motion qualities, but the feature set should not be redundant.

All the features are extracted within a motion segment. Section 5.6 discusses how to break a motion trajectory into motion segments.

5.5.1 Basic Motion Features

From the motion capture data, we know the displacement $\hat{d}_i$ and the timestamp t_i at each frame. For a given segment, we can easily compute the total time:

$$\mathbf{T} = t_n - t_1 \tag{5.2}$$

and the total displacement:

$$\mathbf{D} = \sum_{i=1}^{n} \left\| \hat{d}_{i+1} - \hat{d}_i \right\| \tag{5.3}$$

Also, we can compute the velocity and acceleration at each frame. The average velocity over the time interval Δt is defined as the quotient of the displacement Δd and the time interval Δt. When the time interval Δt is very small, we can assume that the *instantaneous velocity* at the frame is the average velocity over Δt.

$$\begin{cases} \hat{\mathbf{v}}_\mathbf{1} & = \frac{\hat{\mathbf{d}}_\mathbf{2} - \hat{\mathbf{d}}_\mathbf{1}}{t_2 - t_1} \\ \hat{\mathbf{v}}_\mathbf{n} & = \frac{\hat{\mathbf{d}}_\mathbf{n} - \hat{\mathbf{d}}_\mathbf{n-1}}{t_n - t_{n-1}} \\ \hat{\mathbf{v}}_\mathbf{i} & = \frac{\hat{\mathbf{d}}_\mathbf{i+1} - \hat{\mathbf{d}}_\mathbf{i-1}}{t_{i+1} - t_{i-1}} \end{cases} \tag{5.4}$$

Similarly, the instantaneous acceleration can be approximated by the average acceleration over a small time interval Δt.

$$\begin{cases} \hat{\mathbf{a}}_\mathbf{1} = \frac{\hat{\mathbf{v}}_\mathbf{2} - \hat{\mathbf{v}}_\mathbf{1}}{t_2 - t_1} \\ \hat{\mathbf{a}}_\mathbf{n} = \frac{\hat{\mathbf{v}}_\mathbf{n} - \hat{\mathbf{v}}_\mathbf{n-1}}{t_n - t_{n-1}} \\ \hat{\mathbf{a}}_\mathbf{i} = \frac{\hat{\mathbf{v}}_\mathbf{i+1} - \hat{\mathbf{v}}_\mathbf{i-1}}{t_{i+1} - t_{i-1}} \end{cases} \tag{5.5}$$

The average velocity and acceleration of the segment can be computed as:

$$\begin{cases} \bar{\mathbf{v}} = \frac{\sum_{i=1}^{n} \|\hat{\mathbf{v}}_\mathbf{i}\|}{n} \\ \bar{\mathbf{a}} = \frac{\sum_{i=1}^{n} \|\hat{\mathbf{a}}_\mathbf{i}\|}{n} \end{cases} \tag{5.6}$$

All these features are very basic but important, and our acquisition process is based on these features. For example, our experimental study shows that there is a strong correlation between Free Flow and "spontaneity," which is manifested in the abundance of accelerations and decelerations in a motion. Bound Flow shows few such fluctuations (see Figure 5.4). We therefore defined a feature called *PAD*, which is the percentage of accelerations and decelerations arising in a specific motion segment.

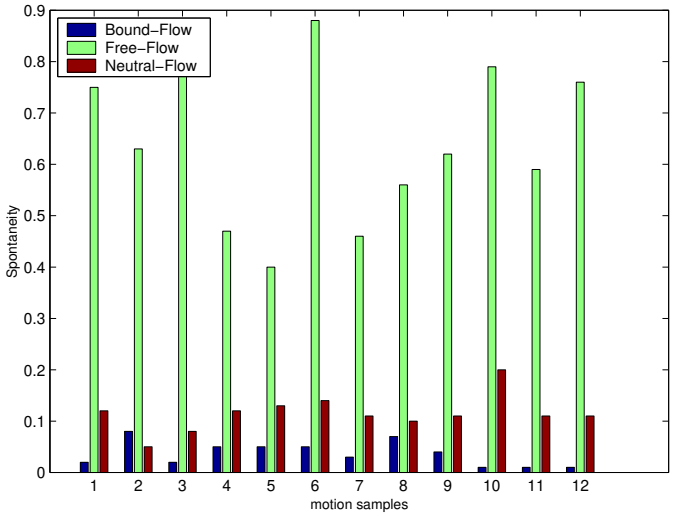

Figure 5.4: PAD in Bound, Neutral, and Free Flow

5.5.2 Curvature and Torsion

Curvature and torsion are two important geometric properties of the motion trajectory. Curvature (κ) is a measurement of the rate at which the tangent vector $\hat{\mathbf{T}}$ turns as the trajectory bends, while torsion (τ) is a measurement of how much the trajectory rotates or twists as it moves along.

Curvature can be computed as the cross product of vectors $\mathbf{v}$ and $\mathbf{a}$

$$\kappa = \hat{\mathbf{v}} \times \hat{\mathbf{a}} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \dot{x} & \dot{y} & \dot{z} \\ \ddot{x} & \ddot{y} & \ddot{z} \end{vmatrix}$$

and torsion

$$\tau = \pm \frac{\begin{vmatrix} \mathbf{v}(x) & \mathbf{v}(y) & \mathbf{v}(z) \\ \mathbf{a}(x) & \mathbf{a}(y) & \mathbf{a}(z) \\ \dot{\mathbf{a}}(x) & \dot{\mathbf{a}}(y) & \dot{\mathbf{a}}(z) \end{vmatrix}}{\|\hat{\mathbf{v}} \times \hat{\mathbf{a}}\|^2} \tag{5.7}$$

where $\mathbf{v}(x)$, $\mathbf{v}(y)$ and $\mathbf{v}(z)$ are components of velocity vector $\hat{\mathbf{v}}$ on the x, y and z dimensions respectively. The sign ($\pm$) is chosen to make τ always ≥ 0.

The remarkable property that both curvature and torsion have is no matter how variable

the motion may be, the curvature and torsion seem to be independent of the way the trajectory is traversed.

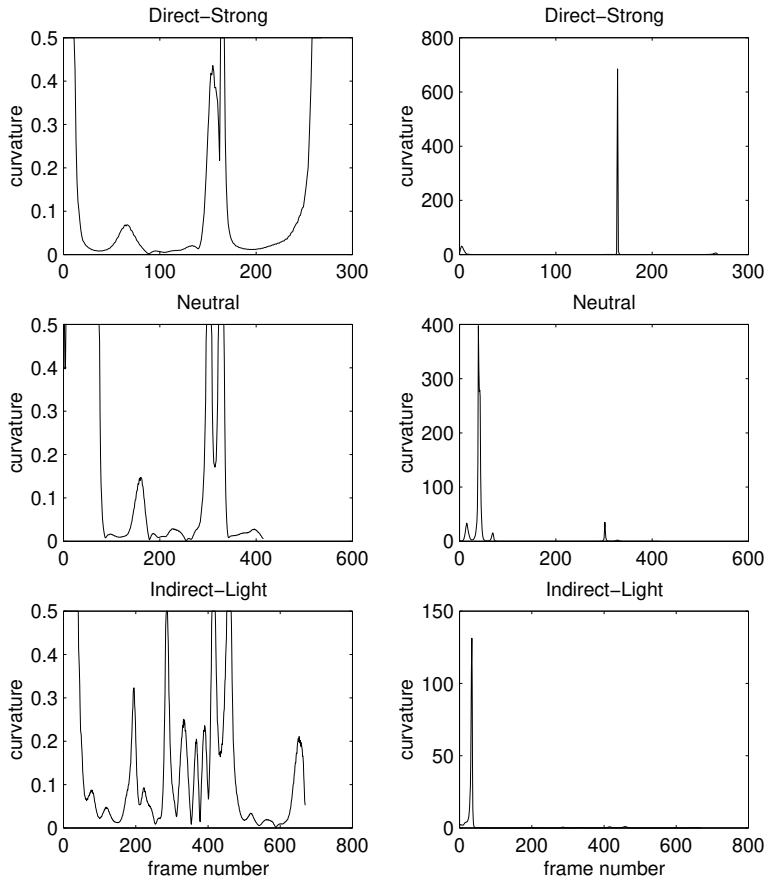

Figure 5.5: Path curvatures (left) and corner curvatures (right)

Our investigation over 288 motion samples finds that, in general, the curvature is prominently high when the motion starts from rest, comes to a stop, or changes its direction (see the right column in Figure 5.5). To analyze this feature more carefully, we break the curvature into two categories: one is the curvatures at the direction-changing proximities, the other is the curvatures that are not in the proximities of the direction-changing locations. We call them *corner curvature* and *path curvature*, respectively.

As shown in Figure 5.5, *corner curvatures* of Direct and Strong motions are eminently higher than those of Indirect and Light motions. However, *path curvatures* of Indirect and Light motions are noticeably higher than those of Direct and Strong motions.

5.5.3 Swivel Angles

Empirical studies [32] show that Indirect and Free movements tend to be driven by the elbow, which implies that there may be some significant swivel changes during Indirect and Free movements. Our approach is to first estimate the swivel angles (see Figure 5.6) given the known positions of the shoulder, elbow, and wrist, and then compute a group of parameters associated with the swivel angles, which are in turn used to help discern the motion qualities.

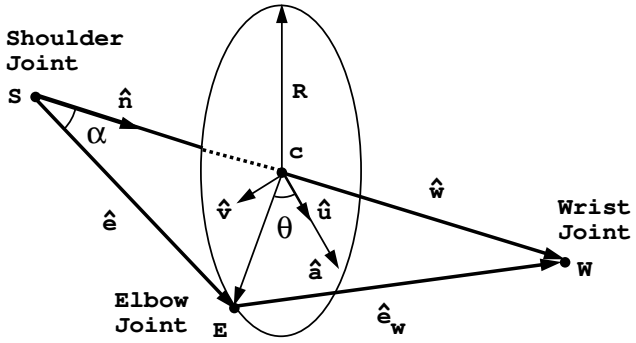

Figure 5.6: Swivel angle in an arm posture

Let $\hat{\mathbf{b}}$ be the vector from the center of the swivel plane $\mathbf{c}$ to the elbow position $\mathbf{E}$. Then the swivel angle is the angle formed between $\hat{\mathbf{b}}$ and $\hat{\mathbf{a}}$, which is chosen to be lying in the swivel plane and pointing downward.

$$\hat{\mathbf{b}} \;=\; \hat{\mathbf{e}} - \hat{\mathbf{c}} \;=\; \hat{\mathbf{e}} - \cos(\alpha)\,\|\hat{\mathbf{e}}\|\,\hat{\mathbf{n}}$$

where $\hat{\mathbf{n}}$ is the normal vector of the swivel plane, and α can be decided from simple trigonometry

$$\cos(\alpha) = \frac{\hat{\mathbf{w}}^T\hat{\mathbf{w}} + \hat{\mathbf{e}}^T\hat{\mathbf{e}} - \hat{\mathbf{e}}_w^T\hat{\mathbf{e}}_w}{2\|\hat{\mathbf{w}}\|\|\hat{\mathbf{e}}\|}$$

According to the definitions of cross and dot products,

$$\begin{cases} \sin(\theta) = \frac{\|\hat{\mathbf{a}}\times\hat{\mathbf{b}}\|}{\|\hat{\mathbf{a}}\|\|\hat{\mathbf{b}}\|} \\ \cos(\theta) = \frac{\hat{\mathbf{a}}\cdot\hat{\mathbf{b}}}{\|\hat{\mathbf{a}}\|\|\hat{\mathbf{b}}\|} \end{cases}$$

It immediately follows

$$\theta = atan(\frac{\sin(\theta)}{\cos(\theta)}) = atan2(\left\| \hat{\mathbf{a}} \times \hat{\mathbf{b}} \right\|, \hat{\mathbf{a}} \cdot \hat{\mathbf{b}}) \tag{5.8}$$

Careful examination of 288 motion samples proves the empirical studies [32] are statistically correct. Figure 5.7 shows some of the comparisons among Indirect, Direct, and Neutral movements in the Space dimension.

We observed that Indirect movements tend to have larger changes in swivel angles than Direct and Neutral movements and such changes often occur in the high frequency spectrum. We also observed that Direct and Neutral movements may have large absolute swivel angles depending on particular movements. To make our analysis independent of the peculiarities of the movements and focus on the regions in which differences appear more often and larger, we use a discrete Fast Fourier Transformation (FFT) to filter out the low frequencies and only compare the differences in the high frequencies. Figure 5.8 shows the transformations on an Indirect and a Neutral movement (The filtering threshold value is set to 5).

We have identified five parameters to quantify the spatial and temporal differences among these movements: (1) the average swivel angle changing rate (or velocity), (2) the total summation of the swivel angle velocities (3) the number of zero-crossings of the second derivative, (4) the total pendulum distances (swivel angle changes between all the neighboring zero-crossings), and (5) the difference between the maximum and minimum swivel angles.

5.5.4 Wrist Angles

Wrist angle is another important index for showing motion qualities. Careful human movement observation reveals that Indirect and Free movements tend to have more frequent wrist angle changes than Direct, Bound, and Sudden movements. Wrist angle is easily computable from the 6D (position and orientation) motion capture data.

Suppose $\hat{\mathbf{e}}$ is the vector from the elbow to the wrist, and $\hat{\mathbf{n}}$ is the normal vector of the palm, which can be captured by the sensor attached to the hand. Then, the wrist angle φ

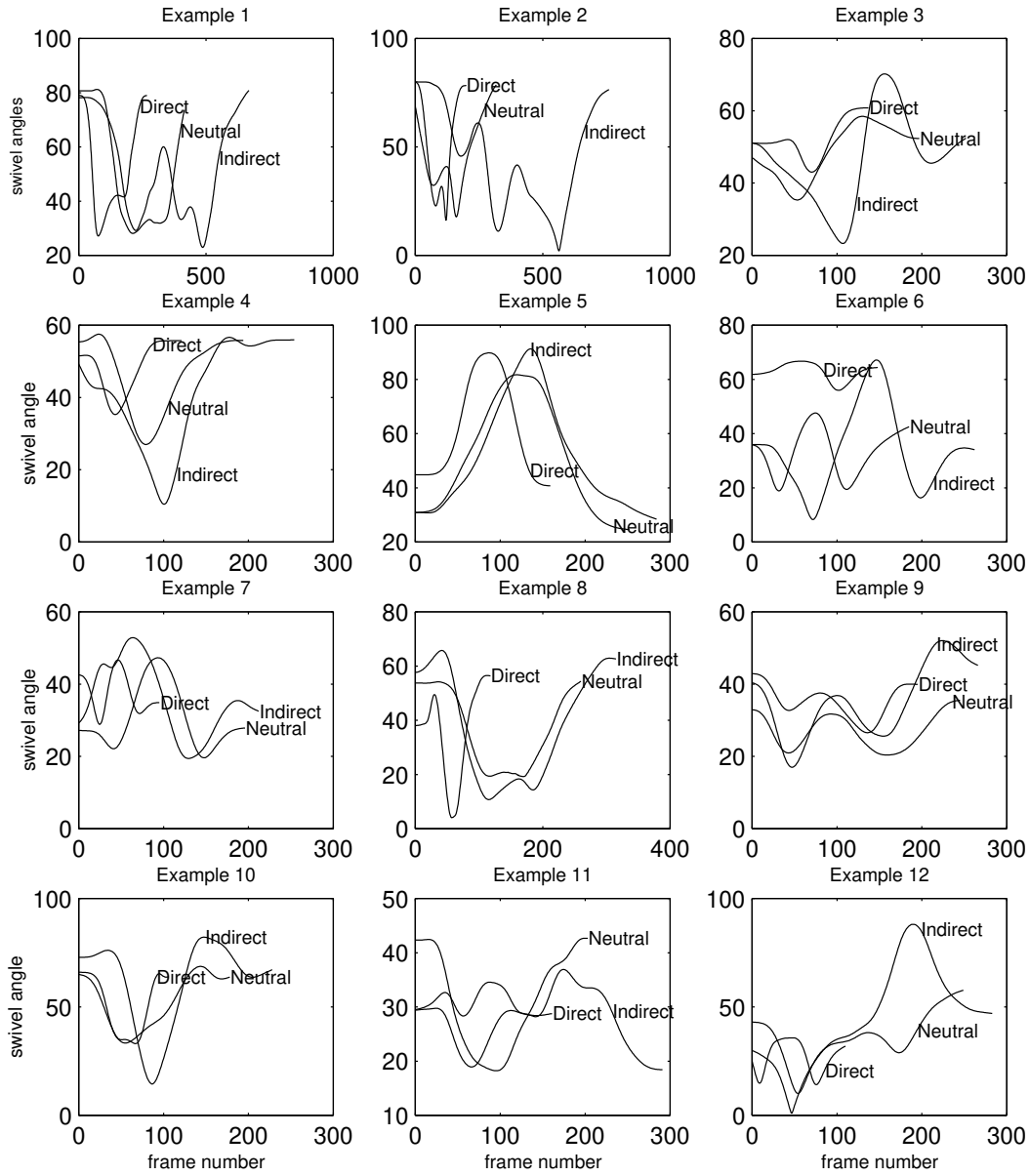

Figure 5.7: Swivel angle examples

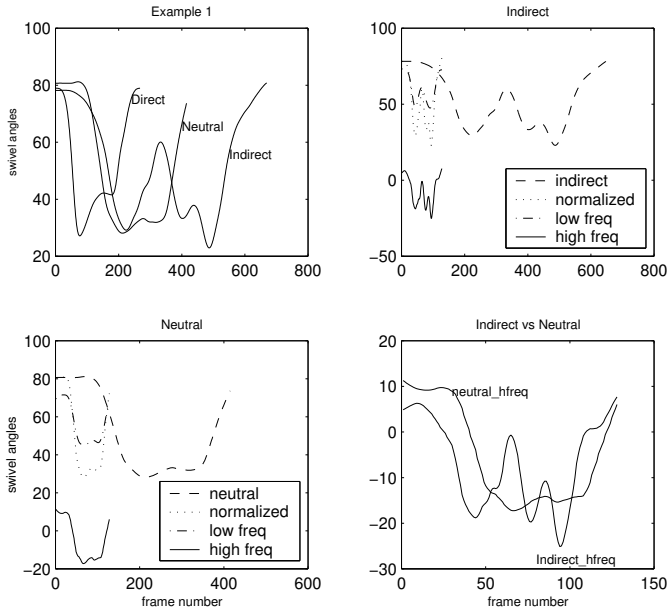

Figure 5.8: Use FFT to extract high spectrum

can be estimated by

$$\varphi = \begin{cases} \beta - 90 & if \quad \beta > 90 \\ 90 - \beta & if \quad \beta \leq 90 \end{cases} \tag{5.9}$$

where

$$\beta \quad = \quad acos(\frac{\hat{\mathbf{n}} \cdot \hat{\mathbf{e}}}{\|\hat{\mathbf{n}}\| \, \|\hat{\mathbf{e}}\|})$$

Figure 5.9 shows the changes of the wrist angle during the same corresponding motions shown in Fig 5.7. Again we can clearly tell the differences among the Indirect, Direct and Neutral movements.

We have chosen five variables to quantitatively measure these differences: (1) the total summation of the wrist angles, (2) the maximum wrist angle, (3) the total summation of the wrist angle changing rate (or velocity), (4) the number of zero-crossings of the second derivatives, and (5) the total pendulum distances (wrist angle changes between zero-crossings).

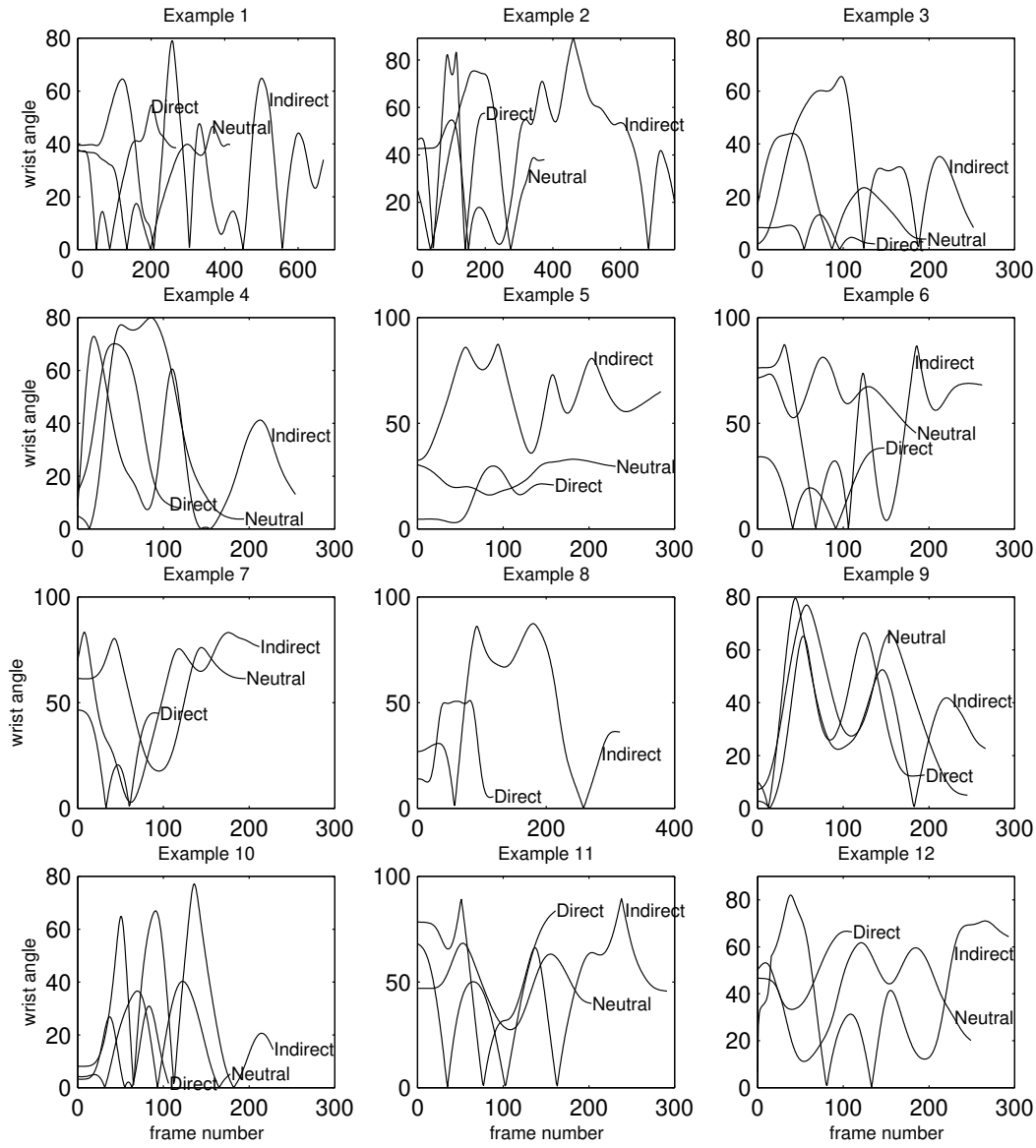

Figure 5.9: Wrist angle examples

5.5.5 Sternum Height

Sternum height is one of the motion features that we use to estimate Weight qualities and to help discern between different Effort qualities.

In a Weight motion, "the prevailing effort is of muscular tension."([93], p. 199) However, the motion capture system cannot directly measure muscular tension. We use the sternum height as an indirect indicator of muscular tension. According to the Effort-Shape affinities [12], a Light Weight motion generally corresponds to a Rising Shape while a Strong Weight motion usually appears in tandem with a Sinking Shape. A sensor, attached to the body near the sternum, is used to track the ups and downs of the body. The sternum height is measured as the distance between the lowest and the highest point in a movement.

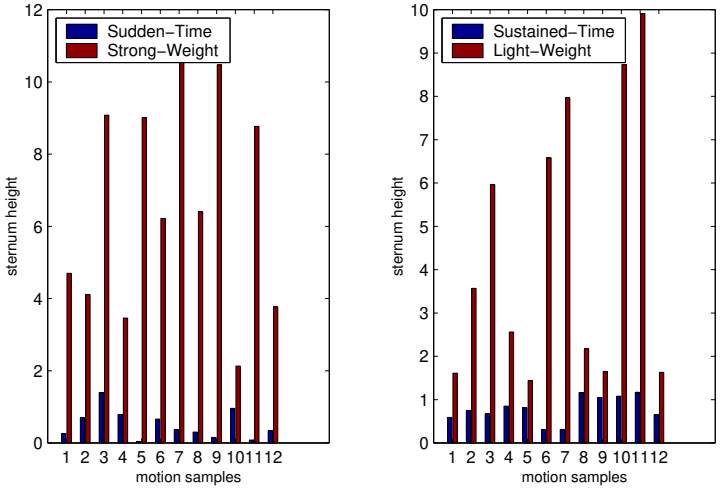

Figure 5.10: Sternum height differences between Strong Weight and Sudden Time (left), and between Light Weight and Sustained Time motions (right)

Sometimes notators fail to recognize the actual presence of the Weight element when Sudden Time and Strong Weight are simultaneously active in the same movement, or mistake Sudden Time as Strong Weight and vice visa when only one of them is actually present in a movement. The addition of sternum height to form a combination of motion factors can help to discriminate between such cases. Although the correlation between muscular tension and the sternum height does not always hold up, particularly in some

subtle gestures, our experimental data shows that the sternum heights are prominently higher in Strong Weight and Light Weight motions than those in motions with other Effort qualities (see Figure 5.10 for a comparison between Weight and Time dimensions).

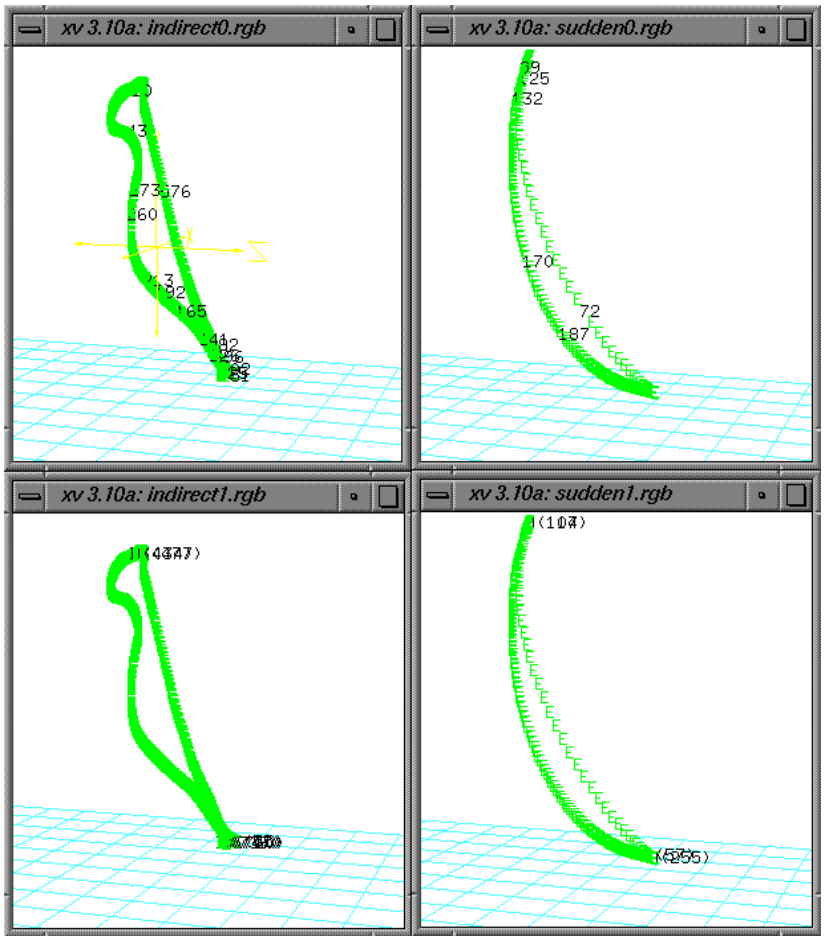

Figure 5.11: Zero-crossing and curvature

5.6 Segmentation

Bindiganavale [18] uses the zero-crossings of the second derivative of the motion data to detect the descriptive changes in the motion. However, we find the approach does not work well in our case because it is hard to find a consistent threshold value that can reliably detect all the significant changes over a variety of motions—if the threshold is set low there

are many zero-crossing points detected in some Indirect, Sudden, and/or Strong motions; if the threshold is set high there are few, and sometimes no zero-crossing points detected in some Indirect, Sustained, and/or Free motions.

We use a combined zero-crossing and curvature method. Digital examination of 288 motion samples shows that the motion curvature is prominently high when the motion starts from rest, comes to a stop, or changes its direction. We skip the noisy periods that are shortly after the start and shortly before the end, and focus on the turning points: ones where significant motion quality changes frequently arise [12, 9]. The method gives us consistently good results over the 288 samples. The top row in Figure 5.11 shows the breakpoints produced by the zero-crossing method with threshold set to 1.0; the points shown in the bottom row are detected by our method with the zero-crossing threshold set to 1.0 and the curvature threshold set to 0.5. Motions on the left column use Indirect and Light while those on the right use Sudden and Direct.

5.7 Backpropagation Networks

We use a one-hidden-layered feedforward neural network with error backpropagation to estimate the relationships among the motion features and the Effort qualities that are associated with the movements. Figure 5.12 shows the architecture of a backpropagation neural network with one hidden layer.

The input to the jth hidden neuron is a linear projection of the input vector $\vec{x}$,

$$u_j = \sum_{i=0}^{I} x_i w_{ij}$$

where x_0 is the bias (equal to 1), and w_{ij} is the weight connecting input neuron i and hidden neuron j. The output of the hidden neuron is

$$h_j = \sigma(u_j) = \sigma(\sum_{i=0}^{I} x_i w_{ij})$$

where $\sigma(\cdot)$ is a nonlinear activation function. The most commonly used activation function is the sigmoid function

$$\sigma(y) = \frac{1}{1+e^{-y}}$$

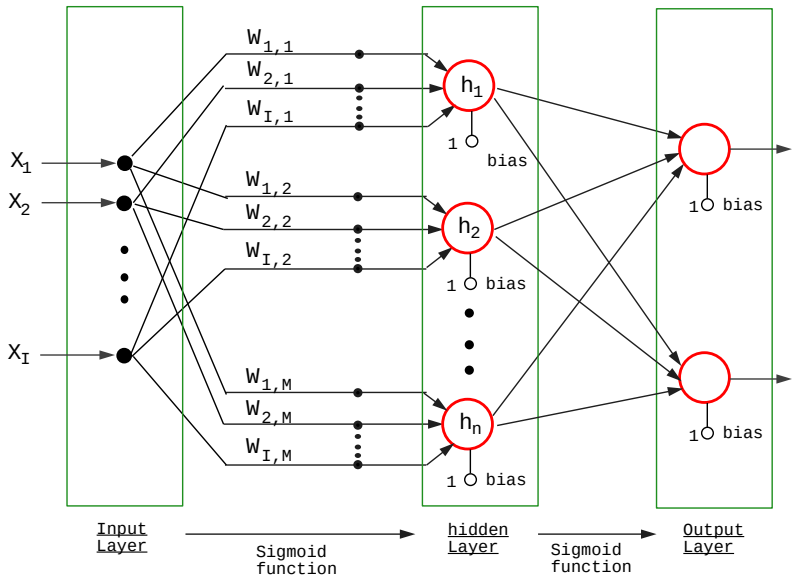

Figure 5.12: Backpropagation neural network with one hidden layer

The input to output neuron is

$$v_k = \sum_{j=0}^{H} x_i w_{jk}$$

and the output is

$$o_k = \sigma(v_k) = \sigma(\sum_{k=0}^{H} h_j w_{jk})$$

where w_{jk} is the weight connecting hidden neuron j and output neuron k.

The sum-of-squares error function has been frequently used as a measurement of training errors. Backpropagation employs gradient descent to attempt to minimize this error term:

$$E = \frac{1}{2} \sum_{k \in D} (t_k - o_k)^2$$

where D is the set of training samples, t_k is the target output and o_k is the network output for training sample k. (more details are presented in Section 5.7.3.)

Instead of using one network for all Effort dimensions (Space, Weight, Time, and Flow), we use one network for each dimension. This provides more degrees of freedom to the

networks for learning the hidden classification functions. It also provides more flexibility in choosing a variable number of motion features for different dimensions.

5.7.1 Input Encoding

As explained in Section 5.5, motion capture data is preprocessed and motion features are extracted. These features serve as different dimensions in the sample space where a nonlinear decision surface will be learned by the neural network. Multiple input features used for the Space network are shown and compared in Figure 5.13. They are extracted from 38 sample motions performed by our professional LMA notators.

Before the input features are fed into the network, they must be first scaled and normalized because they are in different measurement units. We use a simple linear mapping of the motion features' practical extremes to the acceptable neural network extremes:

$$X = s(F - F_{min}) + X_{min} \tag{5.10}$$

where

$$s = \frac{X_{max} - X_{min}}{F_{max} - F_{min}}$$

F_{max} and F_{min} are maximum and minimum limits of the motion feature, respectively, depending on the whole data set. X_{max} and X_{min} are the scaled maximum and minimum limits, which are assigned to 1.0 and 0.0, respectively.

5.7.2 Output Encoding

We could output the three-way classification using a single output neuron, assigning outputs of, say 0.1, 0.5, and 0.9, to encode the three possible values. Instead we use three distinct output neurons, each representing one of the three possible qualities. Also, rather than using 0 and 1 values, we use values of 0.1 and 0.9. The reason for avoiding the use of 0 and 1 is that the sigmoid function cannot produce them given any finite weights. The gradient descent will force the weights to grow without bound but the target can never be reached.

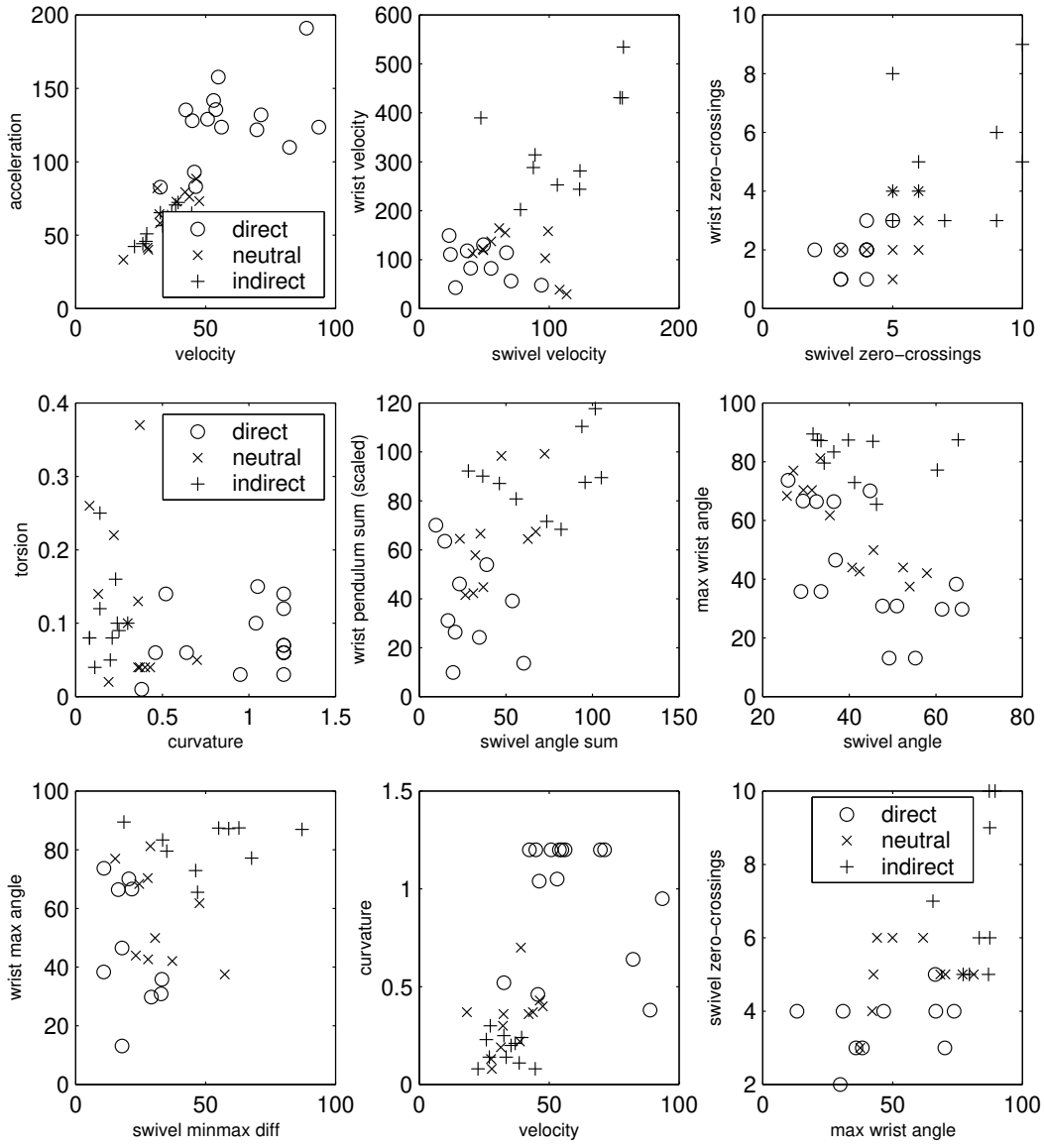

Figure 5.13: Some input features to the Space Network

5.7.3 Training Algorithm

The training vectors consist of pairs of the form $<\vec{x},\vec{o}>$, where $\vec{x}$ is the vector of network input values including the derived motion features such as velocity and acceleration that are computed from motion segments, and $\vec{o}$ is the vector of target network output values, which are numerical settings for the Effort Elements provided by an LMA notator. The training algorithm is as follows:

- Construct a three-layered feedforward network with N_i inputs, N_h hidden neurons, and N_o output neurons.
- Initialize all network weights to small random numbers. (see Section 5.7.5 for an explanation and justification.)
- Until the termination condition is satisfied, Do
  - For each $<\vec{x}, \vec{o}>$ in the training vectors, Do
    1. *Propagate the input forward through the network*
       (a) Feed the input vector $\vec{x}$ to the network and compute the hidden output h_j (for hidden neuron j) and target output o_k (for output neuron k).
    2. *Propagate the errors backward through the network*
       (a) For each network output neuron k, calculate its error item δ_k:

$$\delta_k \leftarrow (t_k - o_k)o_k(1 - o_k)$$

δ_k is the $(t_k - o_k)$ from the delta rule [1], multiplied by the factor $o_k(1-o_k)$, which is the derivative of the sigmoid function.

       (b) For each hidden neuron h, calculate its error term δ_h:

$$\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{hk}\delta_k$$

The error term for hidden neuron h is calculated by summing the error terms δ_k for each output neuron influenced by h, weighting each of the δ_k's by w_{hk}, which characterizes the degree to which hidden neuron h is *responsible for* the error in output neuron k.

[1] The delta rule is a method that uses *gradient descent* to search for possible weight factors to find the weights that best fit the training vectors.

(c) Update each network weight w_{ij}:

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$$

where

$$\Delta w_{ij} = \eta \delta_j x_{ij} + \alpha \Delta w_{ij}$$

where η is the learning rate, which is the factor that determines the size of the steps that the network takes in navigating the weight space in order to minimize the magnitude of the training error; and α is the momentum that increases the size of a step when the direction in the weight space is the same as the direction of the previous step, and decreases the size of a step when the directions of the current and previous step are not the same.

After the network has been trained, it can be used to predict motion qualities of new motions.

5.7.4 Network Structure Determination

We use a one-hidden-layered network structure. Although the two-hidden-layered network is more flexible in describing a complicated relationship, it has a drawback—a significant increase in processing time. Hornik [58] proves that the one-hidden-layered network with a sufficiently large number of hidden neurons can represent *any* functional relationship between the input variables and the output variables. Therefore, we concentrate on the second issue: how many hidden neurons are preferred to achieve the optimal classification in a one-hidden-layered structure. We do not address the problem of whether or not a network of more than one hidden layer may have a smaller total number of neurons in the hidden layer, however.

The method we use is based on [104]: check whether there is any redundant information on the outputs of the hidden neurons and, if any exists, the redundancy is eliminated using principal component analysis (PCA).

5.7.4.1 Principal Component Analysis

The goal of using PCA is to identify as few as possible hidden neurons that can explain all (or nearly all) of the total variance. Since each hidden neuron has as input the linear sum of the input variables and produces as output the sigmoidal transformation of the input. If there is any redundant information (for example, a hidden neuron can be represented by another hidden neuron or a set of hidden neurons), the "rank" of the correlated coefficient matrix of outputs of the hidden neurons will be less than the number of hidden neurons [2]. The steps of the PCA method that we use to narrow down the hidden neurons are as follows:

1. Initially train the network with an arbitrarily large number of hidden neurons. [3]
2. Obtain the correlated coefficient matrix ($p \times p$) of outputs of the hidden neurons, where p is the number of hidden neurons previously chosen.
3. Compute the eigenvalues of the matrix.
4. Count the number (p^*) of eigenvalues whose value is greater than one.
5. (a) If p^* is less than p, choose p^* as the number of hidden neurons. The process may repeat on the new structure to ensure that the optimal number of hidden neurons is actually obtained.

   (b) Otherwise, no redundant information is confirmed; however, it is not guaranteed that the current network has the optimal number of hidden neurons. The network may need more hidden neurons to improve its performance. This should not happen as long as the initial network has a sufficiently large number of hidden neurons.

After determining the optimal number of hidden neurons, the network is retrained with the new structure.

[2] Because the real-world data has a component of random variations, mathematically, the rank of the matrix will never be less than the number of hidden neurons. Here we donot count the eigenvalues that are very close to zero.

[3] In our experiments we arbitrarily picked a number between 8 and 16 as the initial number of hidden neurons. Starting with different numbers does not have a significant effect on the computation results of the optimal number of hidden neurons.

5.7.4.2 Space Network

Fourteen motion features are chosen to feed into the input layer. These features are the average velocity and acceleration, corner curvature and torsion, and a set of swivel angle and wrist angle parameters (see Sections 5.5.3 and 5.5.4). As to which features or combination of features are the best to use, we base our decision on observation, logical design, and the PCA method. A feature that can be represented by another feature or a set of other features is excluded.

We start to train the Space Network by initially setting the number of hidden neurons to eight. The eigenvalues of the correlated coefficient matrix of outputs of the hidden neurons are: <6.4164, 1.1015, 0.2899, 0.0858, 0.0523, 0.0266, 0.0172, 0.0105>. The two principal components [4] can explain up to 93.97% of the total variation. Therefore, two hidden neurons are used as an optimal size for the hidden layer and the network is retrained. For comparison the same data set is trained by four other different network models as well. All networks are trained with a momentum factor of 0.3 and a learning rate of 0.3. The experiment is repeated twenty times, each time with different initial weights. Table 5.2 shows that the network with a structure of $14 \times 2 \times 3$ has the smallest validation error (all the errors are computed based on the scaled input).

Network Structure	Training Mean Square Error (TMSE)	Training Mean Absolute Error (TMAE)	Validation Mean Square Error (VMSE)	Validation Mean Absolute Error (VMAE)
$14 \times 14 \times 3$	0.0018	0.0458	0.1124	0.3518
$14 \times 8 \times 3$	0.0020	0.0551	0.0960	0.3105
$14 \times 3 \times 3$	0.0057	0.0796	0.0801	0.2580
$14 \times 2 \times 3$	0.0183	0.0881	*0.0550*	*0.2044*
$14 \times 1 \times 3$	0.1390	0.4124	0.2921	0.6204

Table 5.2: Training and validating results from different structures of the Space network

Note that networks with more than two hidden units have a lower MSE and MAE over the training samples, but a higher MSE and MAE over the validation samples. It shows that, with increasing hidden neurons, the weights are being tuned to fit idiosyncrasies

[4] 6.4164 and 1.1015, which are greater than one.

(or noise) of the training examples that are not representative of the general distribution of examples. (This leads to the so-called *overfitting* or *overtraining* problem, which is discussed in Section 5.7.6.) The network structure with two hidden neurons gives the best generalization performance for the Space network.

5.7.4.3 Time Network

Four motion features—total time (T), total distance (D), average velocity ($\bar{v}$), and average acceleration ($\bar{a}$)—are used for the Time network.

We start with twelve hidden neurons. The eigenvalues of the correlated coefficient matrix of outputs of the hidden neurons are: <6.9384, 2.7981, 1.8804, 0.2028, 0.1067, 0.0331, 0.0267, 0.0069, 0.0027, 0.0022, 0.0019, 0.0001>. The three principal components can explain up to 96.81% of the total variation. Therefore, a structure of $4\times3\times3$ is used for the Time Network. The training and validating results from different network structures are shown in Table 5.3. Although the performance of networks with more hidden neurons is not degraded quickly, the training time is increasingly longer.

Network Structure	Training Mean Square Error (TMSE)	Training Mean Absolute Error (TMAE)	Validation Mean Square Error (VMSE)	Validation Mean Absolute Error (VMAE)
$4 \times 8 \times 3$	0.0014	0.0459	0.0169	0.1168
$4 \times 4 \times 3$	0.0020	0.0435	0.0142	0.1126
$4 \times 3 \times 3$	0.0025	0.0476	*0.0133*	*0.1075*
$4 \times 2 \times 3$	0.0032	0.0521	0.0179	0.1217
$4 \times 1 \times 3$	0.1298	0.3961	0.1311	0.4183

Table 5.3: Training and validating results from different structures of the Time network

5.7.4.4 Weight Network

In the Weight network, we use as input six motion features: total time, total distance, average velocity, average acceleration, corner curvature, and sternum height.

Initially the network is trained with sixteen hidden neurons. The eigenvalues of the correlated coefficient matrix of outputs of the hidden neurons are: <6.2793, 4.5634, 2.8555, 0.7646, 0.5021, 0.3923, 0.2480, 0.1758, 0.0965, 0.0472, 0.0360, 0.0199, 0.0123, 0.0041,

0.0018, 0.0011>. We choose to use $6 \times 3 \times 3$ as the network structure. The validating results from different network structures (shown in Table 5.4) prove that the structure gives the optimal performance on previously unseen samples, and therefore the best generalization accuracy.

Network Structure	Training Mean Absolute Error (TMSE)	Training Mean Absolute Error (TMAE)	Validation Mean Square Error (VMSE)	Validation Mean Absolute Error (VMAE)
$6 \times 8 \times 3$	0.0372	0.1992	0.1760	0.4260
$6 \times 6 \times 3$	0.0234	0.1780	0.1672	0.4179
$6 \times 4 \times 3$	0.0280	0.2053	0.1636	0.4311
$6 \times 3 \times 3$	0.0593	0.2774	*0.1358*	*0.3984*
$6 \times 2 \times 3$	0.0891	0.2778	0.1941	0.4142
$6 \times 1 \times 3$	0.2524	0.6438	0.4872	0.8896

Table 5.4: Training and validating results from different structures of the Weight network

5.7.4.5 Flow Network

Seven motion features are selected as the input to the Flow network. They are the total time, total distance, average velocity, average acceleration, corner curvature, number of wrist angle zero-crossings and the PAD (the percentage of accelerations and decelerations in a motion as discussed in Section 5.5.1).

We arbitrarily set the initial number of hidden neurons to twelve. The eigenvalues of the correlated coefficient matrix of outputs of the hidden neurons are: <8.5278, 2.3050, 0.5340, 0.3535, 0.1571, 0.0501, 0.0386, 0.0234, 0.0035, 0.0033, 0.0024, 0.0014>. Since the two principal components can explain up to 90.27% of the total variations, we choose $7 \times 2 \times 3$ as the flow network structure. In the experiments where a momentum factor of 0.3 and a learning rate of 0.3 are used, this structure has the best performance over the validation data set.

Examining the lists of eigenvalues (including the ones computed in previous sections) reveals that, while the size of the eigenvalues decreases steadily, it almost never drops to zero. This makes sense, because the real-world data has a component of random variation in the data that never can be linearly represented by one another.

Network Structure	Training Mean Square Error (TMSE)	Training Mean Absolute Error (TMAE)	Validation Mean Square Error (VMSE)	Validation Mean Absolute Error (VMAE)
$7 \times 14 \times 3$	0.0092	0.1178	0.2405	0.4321
$7 \times 7 \times 3$	0.0095	0.1200	0.2574	0.4409
$7 \times 3 \times 3$	0.0130	0.1324	0.2499	0.4525
$7 \times 2 \times 3$	0.0189	0.1296	*0.2232*	*0.4143*
$7 \times 1 \times 3$	0.1434	0.4454	0.3034	0.5445

Table 5.5: Training and validating results from different structures of the Flow network

5.7.5 Convergence and Local Minima

Theoretically, the backpropagation classifier is an optimization of a criterion function with respect to a set of parameters (weights), and the gradient descent is a local optimization technique, therefore, only local minima can be converged upon. In practical applications, local minima have not been found to be as severe as one might fear [92]. If carefully designed, the classifier can be a highly effective function approximation method, despite the lack of assured convergence to a global minimum. We use several heuristics to alleviate the problem of local minima:

- We try to use as many features as possible that can be reliably derived from the motion capture data, without incurring a significant increase in the processing time of the network. For example, we use 14 motion features for the Space network. This generates many input-to-hidden connections and therefore many weights in the network. Since the gradient descent process traverses a weight-error surface in a high dimensional space (one dimension per weight), the more weights in the network, the more dimensions that might provide "escape routes" for the gradient descent to fall away from a local minimum. When the gradient descent falls into a local minimum with respect to one weight, it is not necessarily in a local minimum with respect to all other weights.

- A momentum factor is used in the weight-update rule. The momentum factor can sometimes carry the gradient descent through narrow local minima, although it can also carry it through narrow global minima into other local minima. Local minima in

the region very close to a global minimum are generally considered to be acceptable.

- A method introduced in [133] of initializing the connection weights to small random values is used to avoid false local minima. The sigmoid function is approximately linear when the weights are close to zero. Only after the weights have had time to grow will they make the weight-error surfaces highly nonlinear and generate more local minima. By then we can expect weights have already moved to a region very close to a global minimum.

- A stochastic gradient descent rather than a true gradient descent is used during the learning process [5]. Since each training sample usually has different local minima, it is less likely for a stochastic gradient descent to get stuck in any of them.

- Multiple networks are trained using the same data but with different random starting weights. Since the different training efforts lead to different local minima, the network with the best performance over a separate validation data set may converge to the local minima closest to the global minimum. Iyer and Rhinehart [61] present a method to determine how many networks need to be trained to ensure that the best of those is within a desirable performance within a certain level of confidence. According to the method, twenty networks need to be trained with different initial weights in order to be 99% confident that the best performance of them will result in one of the best 20% values for the sum-of-squared errors over the validation set.

- Alternatively, all the networks can form a voting committee and the final decision is based on the majority of the voting and their past voting credibilities (see Fig. 5.14).

All the heuristics described above are employed in our neural networks.

5.7.6 Generalization and Cross-validation

As mentioned in previous sections, when there are too many weight-tuning iterations during the training process, backpropagation tends to create overly complex decision surfaces that

[5] The difference between the two lies in when to update the weights. A stochastic gradient descent updates the weights after seeing *each* training sample, while a true gradient descent alters the weights after seeing *all* the training samples.

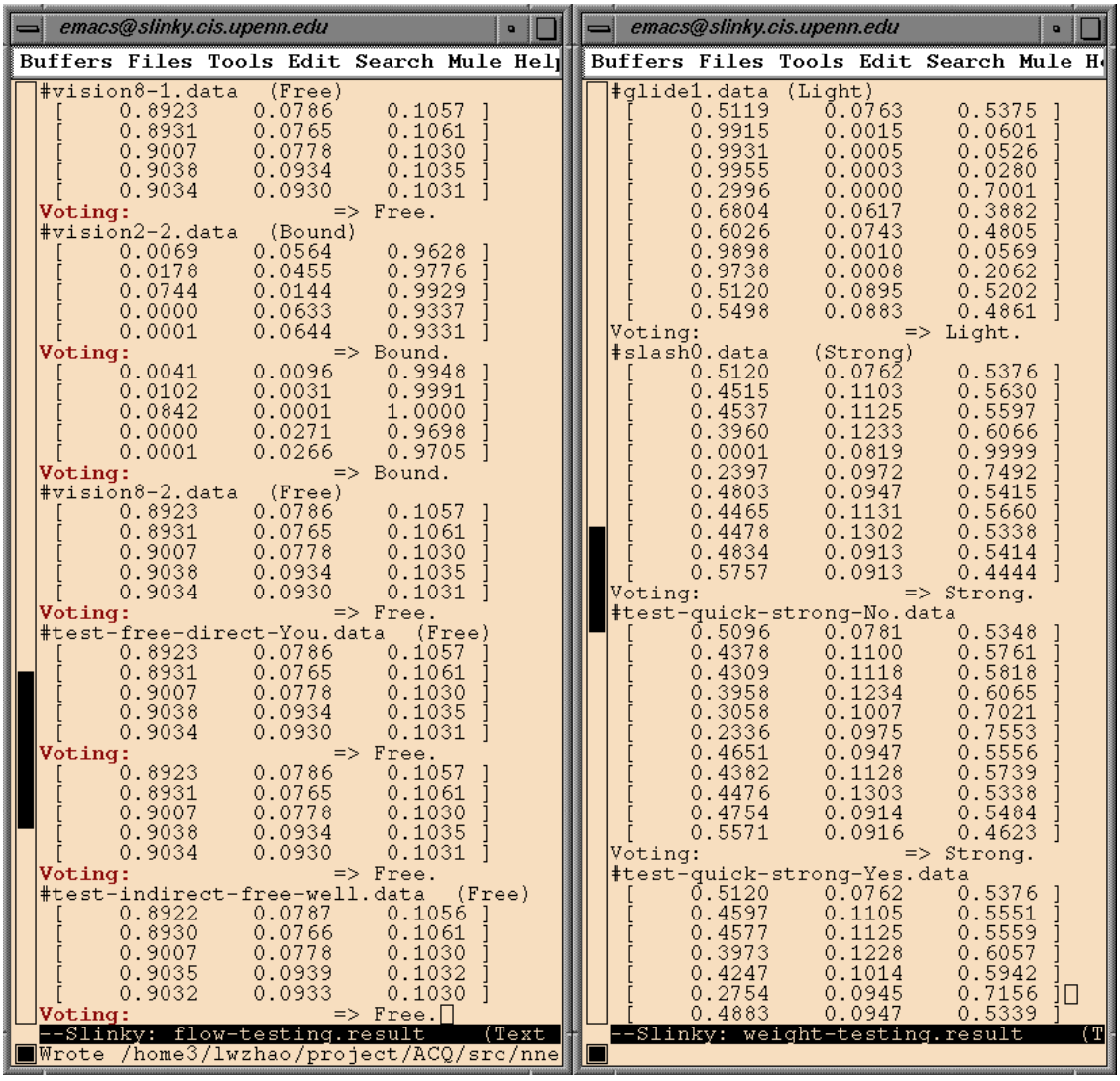

Figure 5.14: Network voting results

fit noise or unrepresentative characteristics of the particular training samples. This is the overtraining problem, which affects negatively the generalization accuracy of the network.

To avoid the overfitting problem, we use a set of validation data, independent of the training data set, to measure the generalization accuracy. The network monitors the error with respect to the validation set while using the training set to drive the gradient descent search. Once the trained weights reach a significantly higher error over the validation set, implying that the networks starting to learn the unimportant details of the training set, the training is terminated. Figure 5.15 shows the cross-validation process in the Space network.

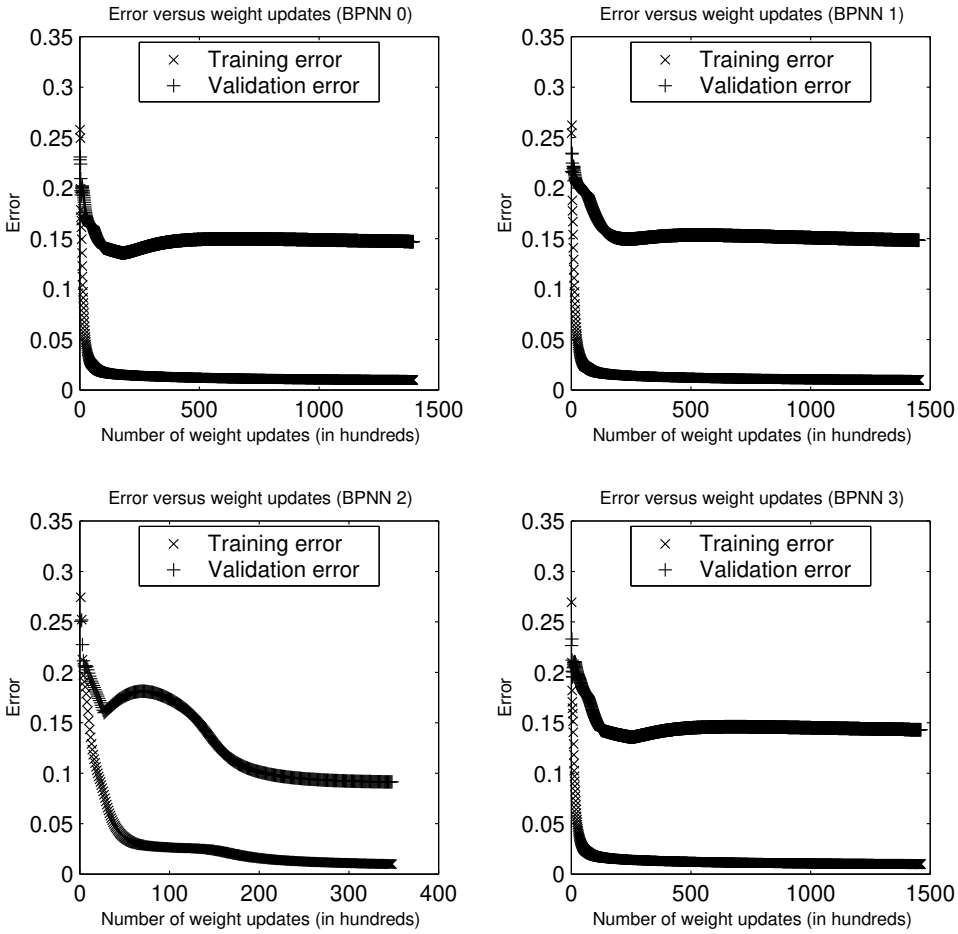

Figure 5.15: Overcoming the overfitting with cross-validation

5.8 Experimental Results

For each Effort dimension, we have constructed a dedicated neural network. Each network is trained, validated, and tested over a number of motion samples.

Every motion sample is created through trial and error. We have two professional LMA notators (a female and a male) working with us. They serve as a subject and an observer in one session and in another session they switch their roles. This makes the observation independent of performance and gender bias. Moreover, the subject and the observer are in the loop. The subject can exploit the immediate visual feedback from the observer, as well as the feedback that is felt kinesthetically by herself during the performance, to change the motion, if necessary, to ensure that the desired quality is reflected both kinesthetically and visually. The process may be repeated multiple times until a consensus between the LMA notators is reached. Furthermore, each of the captured motions is replayed in the Jack environment and an off-line labeling process is undertaken with the help of LMA notators to prune the samples that donot accurately reflect their desired motion qualities.

The entire set of motion samples can be divided into three groups. One group consists of "pure" Effort motions. The motion quality in this group is "pure" in the sense that one particular Effort element is prominently evident while other elements are not readily apparent. In general, pure, isolated Weight, Time, and Space Effort elements rarely, if ever, appear spontaneously; motions with one single isolated Effort element are not only very difficult to perform, but are also very unnatural. In our experiments, instead of trying to capture purely isolated Effort elements, our LMA notators try to demonstrate one Effort dimension as kinesthetically and visually as possible while making other dimensions as neutral as possible.

The second group is composed of motions in which Effort elements are in "mixed" form. We use the principal combinations that have been long identified and well studied in the LMA theory. These combinations are *Action Drives*, *Passion Drives*, *Vision Drives*, and *Spell Drives* (see [12], pp. 57-68). Action Drives are the combinations of Effort Space, Weight, and Time. They include the basic Effort actions: Punch, Float and their modifications: Glide, Slash, Dab, Wring, Flick and Press (see Table 5.6). Combinations of three Effort elements in which Flow is active at the expense of either Space, Weight or

Time are identified as Passion Drives (Spaceless), Vision Drives (Weightless), and Spell Drives (Timeless). These combinations are shown in Table 5.7, 5.8, and 5.9, respectively.

The third group includes some simple gestures that people consciously do in everyday life, such as waving, touching or hitting a balloon.

Each of the motion samples is then visualized in our motion capture system and examined against the video sequences of the same motion that are captured during the live performance. Motions that do not have a visually distinguishable Effort element are removed from the data set. Motions that remain are labeled and used as training, validation, and/or testing samples with known Effort qualities. Table 5.10 shows the partitions of the motion samples for each of the networks [6].

Action	Space		Weight		Time		Flow	
Drive	Indirect	Direct	Light	Strong	Sustained	Sudden	Free	Bound
Punch		X		X		X		
Float	X		X		X			
Glide		X	X		X			
Slash	X			X		X		
Wring	X			X	X			
Dab		X	X			X		
Flick	X		X			X		
Press		X		X	X			

Table 5.6: Effort combinations in the Action Drive

Our testing strategy was chosen considering the following criteria:

- Each network is only responsible for recognizing the Effort elements in its dimension. During the labeling process we only mark down the prominent Effort elements, but the motion may have more-or-less other Effort elements in it. Thus, if a motion being labeled as a Strong Weight, for instance, is fed into the Time network, no matter what the Time network concludes, we do not count it as a failure, nor as a success, for the Time network. This implies that we do not test using Neutral samples.

[6] A k-fold cross-validation method as described in [92] is used to determine how many gradient descent iterations should be performed before the training is forced to terminate. After the optimal number of iterations has been found, a final run of backpropagation is performed training on all the training *and* validation samples—we do not lose any training samples when validating the network.

Passion Drive	Space		Weight		Time		Flow	
	Indirect	Direct	Light	Strong	Sustained	Sudden	Free	Bound
Passion 1				X		X	X	
Passion 2				X		X		X
Passion 3			X			X	X	
Passion 4			X			X		X
Passion 5				X	X		X	
Passion 6				X	X			X
Passion 7			X		X		X	
Passion 8			X		X			X

Table 5.7: Effort combinations in the Passion Drive

Vision Drive	Space—		Weight		Time		Flow	
	Indirect	Direct	Light	Strong	Sustained	Sudden	Free	Bound
Vision 1		X			X			X
Vision 2	X				X			X
Vision 3	X				X		X	
Vision 4		X			X		X	
Vision 5		X				X		X
Vision 6	X					X		X
Vision 7	X					X	X	
Vision 8		X				X	X	

Table 5.8: Effort combinations in the Vision Drive

Spell Drive	Space		Weight		Time		Flow	
	Indirect	Direct	Light	Strong	Sustained	Sudden	Free	Bound
Spell 1		X	X					X
Spell 2	X		X					X
Spell 3		X	X				X	
Spell 4	X		X				X	
Spell 5		X		X				X
Spell 6	X			X				X
Spell 7		X		X			X	
Spell 8	X			X			X	

Table 5.9: Effort combinations in the Spell Drive

Network Name	Network Structure	Total Samples	# Training Samples	# Validation Samples	k-fold Validation	# Testing Samples
Space	$14 \times 2 \times 3$	117	91	9	10-fold	26
Weight	$6 \times 3 \times 3$	114	80	5	16-fold	34
Time	$4 \times 3 \times 3$	199	101	7	14-fold	98
Flow	$7 \times 2 \times 3$	96	47	5	9-fold	44

Table 5.10: Partitions of the available motion samples

- However, if the Strong Weight motion in the previous example is fed into the Weight network but the Weight network predicts Light, it counts as a mistake for the Weight network; if a conclusion of Strong is reached, a success is counted.

- Motions with Effort factors in combination are fed to corresponding networks and tested separately. For example, a Strong and Quick motion is fed into the Weight and Time networks. If the Weight network returns a Strong, a success is counted for the Weight network, otherwise a failure is counted. Similar testing is done for the Time network as well.

The testing data set for the Time network contains 98 motion samples, each of which encodes an Effort Time factor, either as an isolated or as a combined component. The confusion matrix (in Table 1.7) shows that the trained Time network only mistakes 4 Sustained samples as Neutral and predicts perfectly on Sudden samples. Further investigation finds that each of the four Sustained samples occurs at the finishing segment of a motion. We suspect that the Sustained factor perhaps has not been well manifested by our LMA notators in some of the original motions. In the experiments the LMA notators always return to a resting pose and maintain a neutral readiness after finishing each motion; they may transit to the neutral readiness a little too early in some of the four cases.

The experimental results reveal that the Weight network generally predicts motion qualities successfully over a number of motion samples but may get confused when a Strong motion is performed very slowly, or when a Light motion is done very rapidly. As we mentioned before, the network does not have information such as muscle tension and volume changes, instead it makes its decision primarily on the geometric information computed from the motion trajectories. The feature of sternum height can help in some

cases to distinguish between a Strong and a Light motion, but the feature is not always able to draw the line. In some cases, the Weight network may mistakenly interpret a Strong motion that is performed very slowly as a Light motion, and a Light motion that is performed very quickly as a Strong motion. This is certainly a bad interpretation. However, humans cannot do better in such cases, given the data currently available in the system. It would be unrealistic to expect the network to accurately and precisely recognize motion qualities 100 percent of all the time. If additional information such as muscle tension and volume changes can be somehow acquired, this kind of confusion can very probably be avoided.

Time Network		Actual S	N	Q
Predicted	S	44	0	0
	N	4	0	0
	Q	0	0	50

Table 5.11. Confusion matrix

Weight Network		Actual L	N	S
Predicted	L	12	0	2
	N	0	0	0
	S	2	0	18

Table 5.13. Confusion matrix

Flow Network		Actual F	N	B
Predicted	F	27	0	2
	N	1	0	0
	B	0	0	14

Table 5.12. Confusion matrix

Space Network		Actual I	N	D
Predicted	I	13	0	0
	N	1	0	1
	D	0	0	11

Table 5.14. Confusion matrix

Among the 44 testing samples which have either Bound or Free Flow, 41 are recognized correctly by the Flow network. However, the network twice mistakes the Bound component in a Quick and Bound motion as a Free Flow, and misinterprets the Free component in a Free and Sustained motion as a Neutral Flow. The overall recognition rate is 93.18%. There are several possible reasons for the misinterpretations: (1) A Quick and Bound motion has sort of "contradicting" qualities, comprising a quality of impact with an increasing speed and a quality of holding back. This may result in spontaneous changes of the velocity,

misleading the Flow networks to predict Free. In addition, the Flow networks may have relatively larger weights on the input feature of velocity, outweighing the quality of holding back in this case. (2) Additional features, in particular those that can capture the subtleties as the motion ends, may be needed to further separate the dimensions. (3) The networks might have not seen sufficiently enough training samples, particularly those that encode some "contradicting" qualities. Therefore the boundaries formed in the weight space are not fine enough to discriminate the samples that are very close to the boundaries.

The Space network functions very well, as it uses fourteen motion features as input and these features can be very distinctive between Direct and Indirect motions, as shown in Figure 5.13. However, the network does not recognize the Effort quality 100 percent of the time either, particularly when the Indirect component is encoded in an Indirect-Strong-Sudden motion, or when the Direct is mixed with Light and Sustained components, the Space network might be confused.

In summary, all the trained networks have a demonstrated accuracy of about 90% in recognizing Effort motion qualities for a group of people who deliberately made these expressions. The recognition accuracy is equal to or slightly better than an LMA notator, and significantly higher than a naive observer. According to our experiments, the naive observer frequently miss one or two Effort factors in motions that involve a combination of Effort factors. In addition, in order to recognize or realize the subtleties of a particular movement pattern, careful and repetitive observations are often required. Our neural network based systems do much better in such cases. The performance of the acquisition systems could be further improved if more diversified training samples are available, however, whenever computers are asked to make decisions related to problems that cannot be solved with rigorous rules, pure logic, or exhaustive search of a space of all possibilities, they are always subject to errors of judgment. The training based recognizer described in this chapter is no exception.

Chapter 6

Gesture Acquisition from Video

Motion capture provides good accuracy and quick measurements, but attaching electromagnetic or optical sensors and devices limits applications and is too cumbersome and restricting for natural gestures. Markerless video processing and analysis can in fact be used to recover motion structure and styles *directly from 2D images*. The recovered image positions of a specific location (i.e, the right hand) can be transformed into 3D trajectories via triangulation of the measurements from multiple cameras and a parameterized representation of the actor's movements can be calculated [57].

Our major goal in this chapter is to extract the four Effort parameters from 2D image projections. Our vision-based motion estimation algorithms will provide the low-level motion parameters such as 2D (image) position, velocity, and acceleration data and our 3D analysis will provide correlated 3D motion factors. The neural network model trained to recognize EMOTE qualities in the previous chapter can clearly be applied to the reconstructed 3D factors. Like the motion capture model developed in the previous chapter, the video model is essentially a low-to-medium level transformation which involves capturing spatiotemporal patterns and signals of both local and global changes in a movement, and relating these patterns to a category of motion quality, namely the Effort quality.

Figure 6.1 shows the architecture of the whole system, incorporating both the acquisition and the synthesis process. The synthesis part is chiefly used to re-animate the data in graphical output for the purpose of visual evaluation. The acquisition part, which

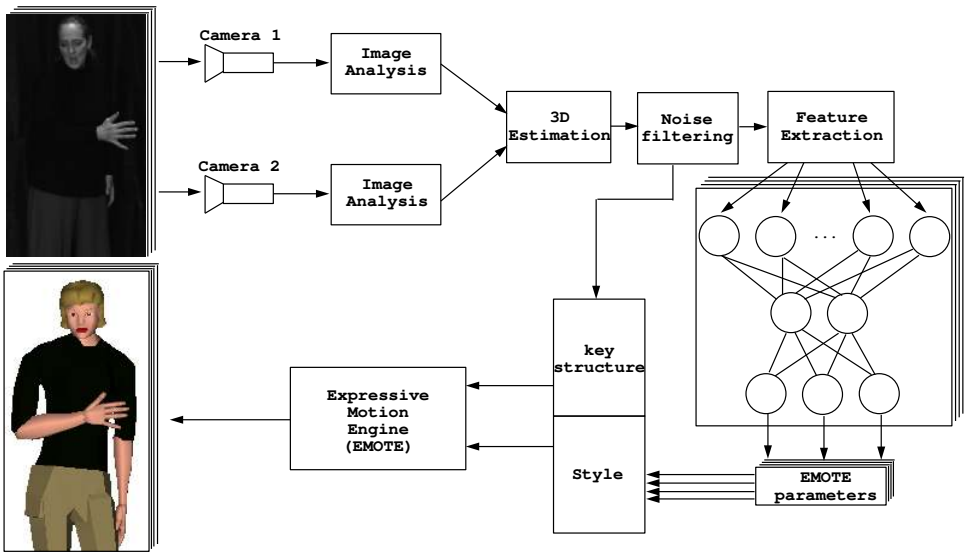

Figure 6.1: The system architecture

we concentrate on in this chapter, comprises three major components: image analysis, feature extraction, and the neural network. The remainder of the chapter is organized as follows. Section 6.1 describes the video system we used to capture motion images. A variety of computer vision techniques are employed to extract the image features in Section 6.2. Section 6.3 briefly describes the algorithm used to estimate the 3D position of the hand and the head. Finally, Section 6.4 presents the sample space used in the experiments, the motion features extracted from the reconstructed motion trajectories, and the experimental results based on the trained neural networks.

6.1 Video Capture System

The video capture system we used is from $Vision\ 1^{TM}$ with two Kodak ES310 cameras. The cameras run in a continuous mode and collect images at frame rate of 43 fps. The cameras and the capture devices (installed on two PCs) are synchronized with an external pulse signal.

To make the acquisition process fast and reliable, we impose some requirements that can be easily satisfied in practice. First, we require the background be of uniformly low

intensity, and the performing person wear dark-color clothes. This enables us in efficient and reliable extraction of the moving hand from the background. Second, the two cameras are pre-calibrated and remain stationary. A planar checkerboard with 13×15 square pattern (each 30mm×30mm) placed at twelve different locations and orientations is used for the calibration. Figure 6.2 shows two different locations and orientations out of the twelve.

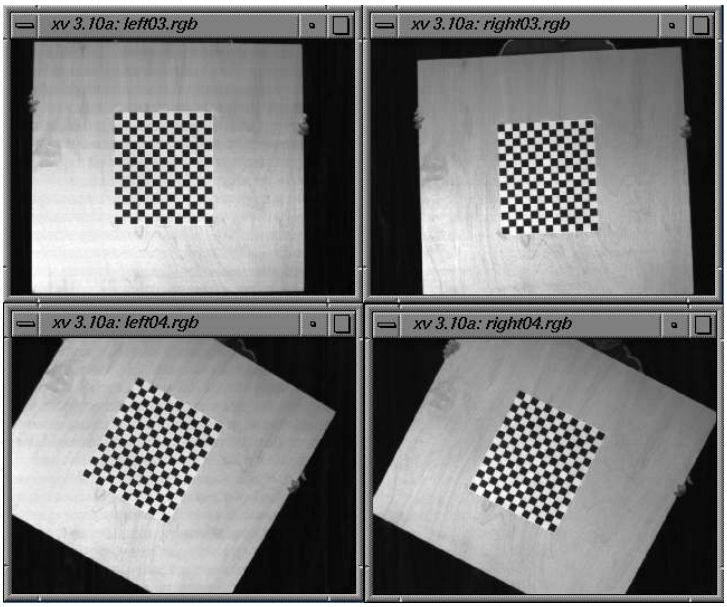

Figure 6.2: The checkboard used for the camera calibration (left column: images captured by the left camera; right column: images captured by the right camera)

6.2 Image Analysis

Upon receipt of each image, the system first processes the image by applying a threshold function

$$B(i,j) = \begin{cases} 1 & I(i,j) > threshold \\ 0 & otherwise \end{cases}$$

where $I(i,j)$ represents the intensity of pixel at (i,j). The resulting image can be regarded as a black and white binary image. The two white regions in the image are the head and

the hand. We use a sequential connected component algorithm [118] to extract and label the two regions. Then, we use a simple heuristic to distinguish the two:

$$\mathbf{A}_h \quad < \quad \mathbf{A}_H$$

where $\mathbf{A}_H$ and $\mathbf{A}_h$ represent the area covered by the head and the hand, respectively. The area of a region is measured by the total number of pixels it covers in the image.

$$\mathbf{A} = \sum_i \sum_j B(i,j)$$

To make the detection algorithm simpler and faster, we assume the hand always starts from a resting position before any motion occurs. A Cartesian coordinate system is defined with its origin fixed at the center of the body and the x-axis going horizontally from the right to the left and the y-axis going upwards. The initial position of the hand in such a coordinate system is supposed to be either in the third or fourth quadrant.

- **Centroid of the hand**

  The coordinates of the centroid are determined by simply averaging the coordinates of each pixel in the hand's area.

  $$\begin{aligned} \bar{x} &= \frac{M_{10}}{M_{00}} \\ \bar{y} &= \frac{M_{01}}{M_{00}} \end{aligned}$$

  where M_{00}, M_{10} and M_{01} are the image moments.

  $$\begin{aligned} M_{00} &= \textstyle\sum_i \sum_j I(i,j) \\ M_{01} &= \textstyle\sum_i \sum_j yI(i,j) \\ M_{10} &= \textstyle\sum_i \sum_j xI(i,j) \end{aligned}$$

  The harder problem is how to locate the centroid on the next image (or the following images)—the well-known inter-frame point/feature correspondence problem. In our approach, a simple heuristic is used to determine which point represents the centroid of the hand. The first image is used as the referencing template to locate the hand and compute its centroid. A 3×3 grid around the centroid is initially calculated:

  $$\begin{bmatrix} I_{0,0} & I_{0,1} & I_{0,2} \\ I_{0,3} & I_{0,4} & I_{0,5} \\ I_{0,6} & I_{0,7} & I_{0,8} \end{bmatrix}$$

For the next image, we use the 3×3 grid as a convolution mask to check each pixel i in the most promising area and determine if it is the centroid according to the least squares matching criterion:

$$\sum_{j=0}^{8}(I_{i,j} - I_{0,j})^2 \quad = \quad min$$

where $I_{i,j}$ is the intensity of pixel j, a neighbor of pixel i in the 3×3 grid.

- **Centroid of the head**

  The centroid of the head can be similarly estimated as well. The locations of the hand and the head are shown in a bounding box, respectively (see Figure 6.4). In our experiments, the head rarely makes any big movement, therefore the 3D position of the head can be well approximated most of the time. The hand moves throughout the near-body and mid-body space, however, and its estimated position can be temporarily deviated from its normal trajectory when the bounding box of the hand collides with the one of the head. Under such circumstances, the estimated position of the hand is adjusted using a simple continuity/discontinuity algorithm described in [57].

- **Orientation of the hand**

  Define the intermediate variables a, b, and c,

$$\begin{aligned} a &= \frac{M_{20}}{M_{00}} - \bar{x}^2 \\ b &= 2(\frac{M_{11}}{M_{00}} - \bar{x}\bar{y}) \\ c &= \frac{M_{02}}{M_{00}} - \bar{y}^2 \end{aligned}$$

  where M_{20} and M_{02} are the second order moments with respect to the pixel at i and j, respectively.

$$\begin{aligned} M_{20} &= \sum_i \sum_j i^2 I(i,j) \\ M_{02} &= \sum_i \sum_j j^2 I(i,j) \end{aligned}$$

  The orientation of the hand can be determined by

$$\varphi = \frac{arctan2(b, (a-c))}{2}$$

6.3 3D Estimation

Given 2D positions in two camera perspective system, we can estimate the 3D positions of the hand and the head.

T_1 : transformation matrix from global to local at camera 1

T_2 : transformation matrix from global to local at camera 2

f_1 : focal length of camera 1

f_2 : focal length of camera 2

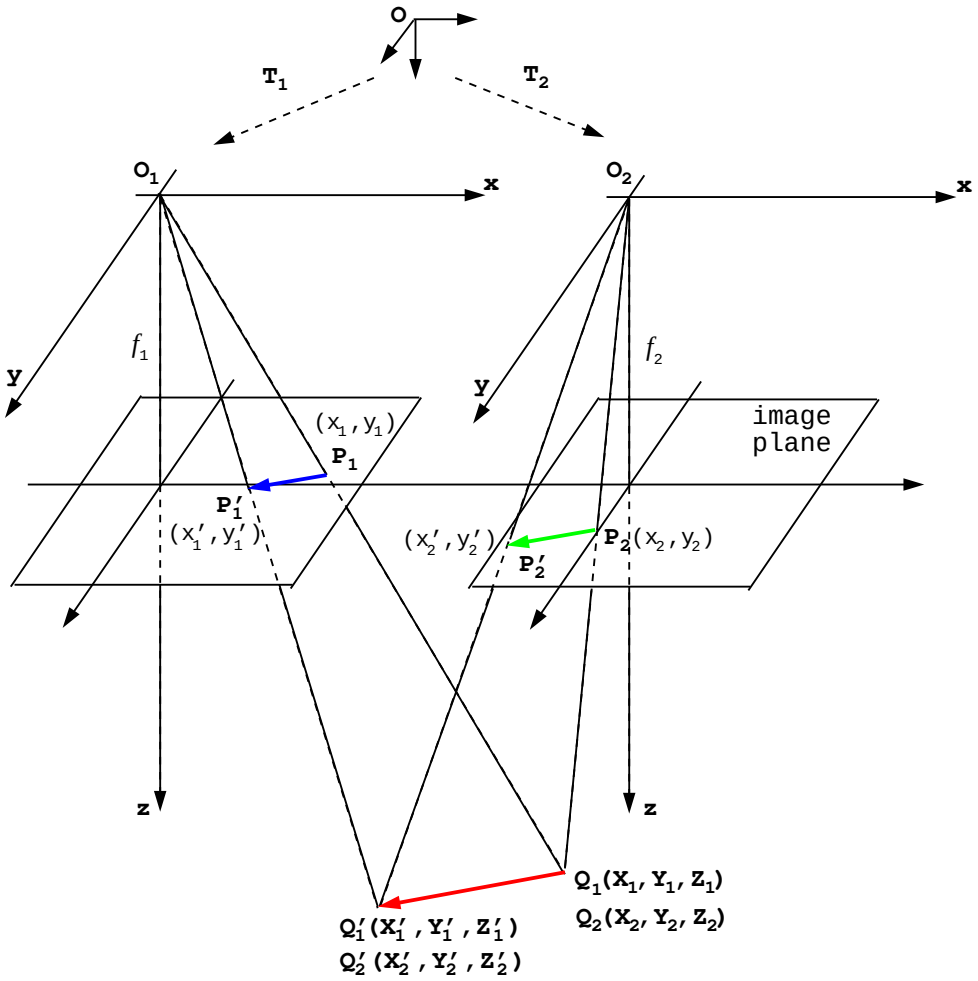

Figure 6.3: The two camera imaging geometry

O : origin of the world coordinate system

O_1 : origin of the coordinate system fixed at camera 1

O_2 : origin of the coordinate system fixed at camera 2

Q= $(X, Y, Z)^T$: spatial vector Q w.r.t O at time t_1

Q_1= $(X_1, Y_1, Z_1)^T$: spatial vector Q w.r.t O_1 at time t_1

Q_1'= $(X_1', Y_1', Z_1')^T$: spatial vector Q w.r.t O_1 at time t_2

Q_2= $(X_2, Y_2, Z_2)^T$: spatial vector Q w.r.t O_2 at time t_1

Q_2'= $(X_2', Y_2', Z_2')^T$: spatial vector Q w.r.t O_2 at time t_2

P_1= $(x_1, y_1)^T$: image vector P_1 w.r.t O_1 at time t_1

P_2= $(x_2, y_2)^T$: image vector P_2 w.r.t O_2 at time t_1

P_1'= $(x_1', y_1')^T$: image vector P_1 w.r.t O_1 at time t_2

P_2'= $(x_2', y_2')^T$: image vector P_2 w.r.t O_2 at time t_2

Assuming that the projection planes are parallel to the xy-plane at $z = f_1$ (for camera 1) and $z = f_2$ (for camera 2), the coordinates of points P_1 and P_2 in the camera coordinate system are given by perspective transformation:

$$\begin{cases} (x_1, y_1) &= \left(\frac{f_1 X_1}{Z_1}, \frac{f_1 Y_1}{Z_1}\right) \\ (x_2, y_2) &= \left(\frac{f_2 X_2}{Z_2}, \frac{f_2 Y_2}{Z_2}\right) \end{cases} \tag{6.1}$$

The points Q_1 and Q_2 are two different representations of the same point Q under different camera coordinate systems.

$$\begin{cases} (X, Y, Z) &= (X_1, Y_1, Z_1)\mathbf{T_1}^{-1} \\ (X, Y, Z) &= (X_2, Y_2, Z_2)\mathbf{T_2}^{-1} \end{cases} \tag{6.2}$$

Combining Eq. 6.1 and Eq. 6.2 ends up with 7 equations in 6 unknowns, therefore, the determination of $(X, Y, Z)^T$ at time t_1 is to find the solution of a set of simultaneous linear equations [57, 118]. We can compute (X', Y', Z') at time t_2 in the same fashion.

6.4 Experimental Results

Compared with the motion capture model, where we have a fairly large sample space of motions that cover different spatial directions, planes, and dimensions and have different

forms, the video model has a rather limited sample space. So far 24 complete motions have been performed by our professional LMA notators and captured through two cameras. Although the sample space is not general enough to be used as a training data set, it might suffice to form a testing data set. On the other hand, the samples that were captured in the motion capture model can actually be re-used in the video model. Indeed, the two models merely differ in how the motions are actually acquired and constructed: the underlying structure and styles of the motion are essentially the same. Some motion factors available or computable through the motion capture system, such as the swivel angles, may not be readily computable through the video capture system. An important hypothesis in our research, however, is that the reduced data set of image motion factors still suffices to provide the essential triggers for the recognition of distinct EMOTE Effort parameters. Based on this assumption, we do not have to bother to capture many motion samples in video, which are often subject to noise and inaccuracies, rather, we re-use the samples that were already captured through the motion capture system, which can provide the good accuracy that a training algorithm requires. Thus, the combined approach enables us to take advantage of both motion capture and video capture system while avoiding some of their disadvantages.

All the motion samples captured through the two cameras are reconstructed in a real-time fashion based on a video capture system developed by Shan Lu [1] (see Figure 6.4). The recovered 3D motion trajectories are then retargeted to the human figure that is geometrically similar to the performing person in the *Jack Toolkit* environment (see Figure 6.5).

Derived from each 3D motion trajectory are ten motion features: the total traversing time (t), total traversing distance (d), average velocity (v), average acceleration (a), number of zero-crossings of the second derivative (nZC) (or the weaving rate), average path curvature ($p\kappa$), average corner curvature ($c\kappa$), average torsion (τ), number of zero-crossings of the first derivative of the wrist orientation ($w\theta$), and head height (hh). Comparing this motion feature set with the one used before in the motion capture model yields some minor discrepancies. However, the wrist orientation and head height can be used as an

[1] He is a researcher working in the Vision, Analysis and Simulation Technology Lab directed by Dimitris Metaxas. He is on a fellowship from Keihanna Human Info-Communication Research Center, Japan.

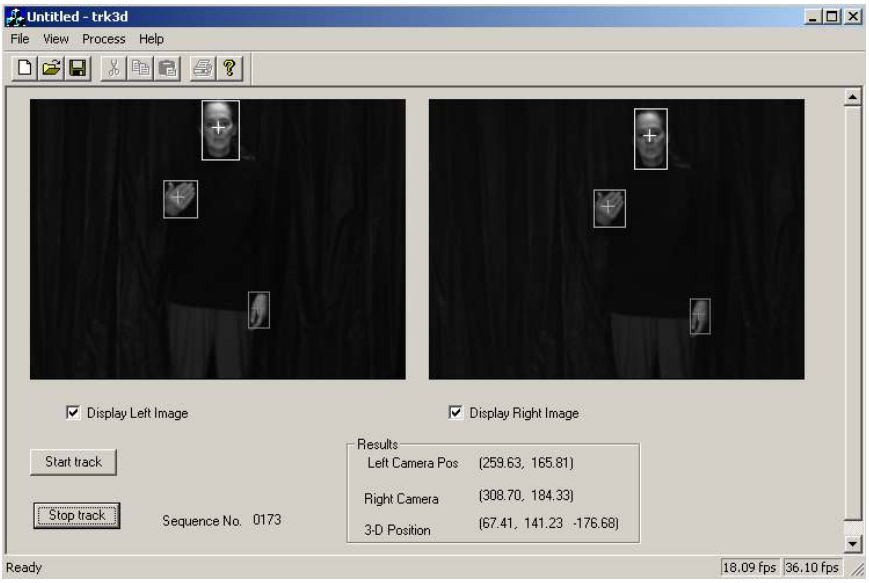

Figure 6.4: 3D estimation from two 2D images

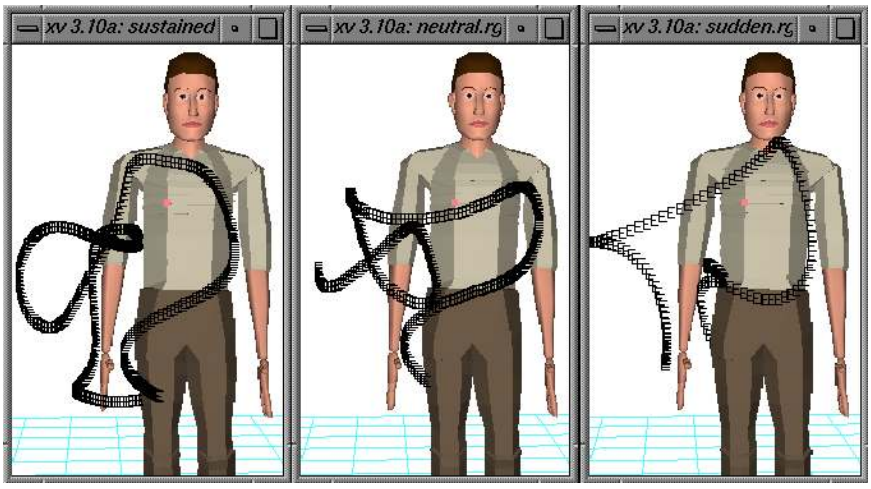

Figure 6.5: Trajectories of different motion styles (left: Sustained, middle: Neutral, right: Sudden)

approximation to the wrist angle and the sternum height, respectively, although they have to be first scaled before they can be actually used in the neural networks. The only neural network used in the motion capture model that has to be restructured to fit to the video capture data is the Space network, since some motion features such as swivel angles are unavailable from the video model. Therefore, the Space network has to be re-trained based on a subset of the motion features that it was previously trained upon. Other networks stay the same.

Although extensive testing with a large set of video captured motion samples has not yet been fully explored, we find that the experiments over the samples we currently have [2] give us quite satisfying results. Comparing the motion qualities automatically recognized by the neural networks with the ones manually specified by our LMA notators yields very few disagreements. Tables 6.1–6.4 show the experimental results. Note that the results are based on motion segments rather than whole motions.

Time Network		Actual S	N	Q
Predicted	S	8	0	0
	N	1	0	1
	Q	1	0	6

Table 6.1. Confusion matrix

Weight Network		Actual L	N	S
Predicted	L	12	0	0
	N	1	0	2
	S	1	0	10

Table 6.3. Confusion matrix

Flow Network		Actual F	N	B
Predicted	F	7	0	0
	N	2	0	2
	B	0	0	11

Table 6.2. Confusion matrix

Space Network		Actual I	N	D
Predicted	I	13	0	0
	N	2	0	1
	D	0	0	10

Table 6.4. Confusion matrix

Compared with the recognition rates we had in the motion capture model, the recognition rates of the video model are a little lower, however, the sample space is not

[2] We have one or two motion samples for each Effort factor captured on the video.

Figure 6.6: Original motion performed by our LMA notator

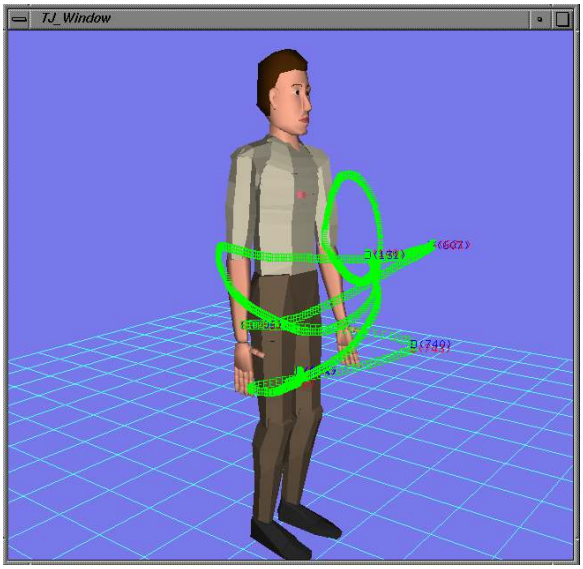

Figure 6.7: Motion trajectory recovered in the video capture system

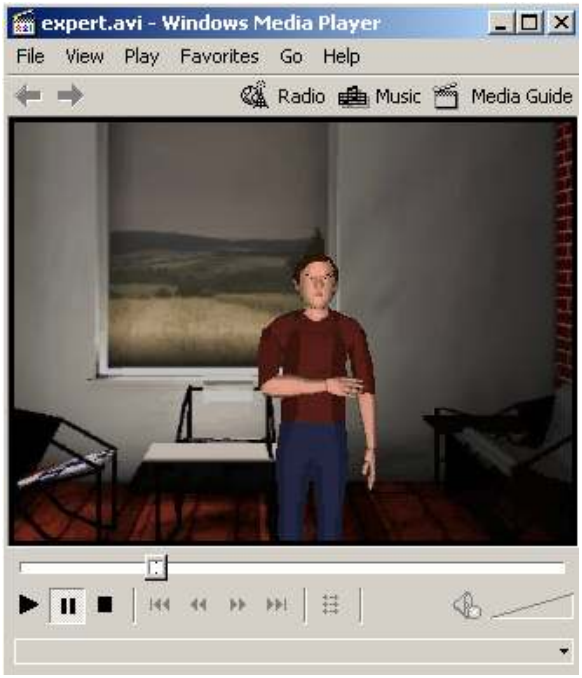

Figure 6.8: Animation generated using expert set qualities

Figure 6.9: Animation generated using learned qualities

big and diversified enough to conclude so. There are several possible reasons for the lower recognition rates: (1) we use some features that are actually approximation of the features that we used in the motion capture model, the estimated features may not be correspond very closely to the ones we used to train the networks; (2) the video data is more noisy than the motion capture data, for example, at some pionts the hand may be very briefly out of the capturing window, and at some other points the bounding box of the hand may collide temporarily with the bounding box of the head. Both cases may cause miscalculation of the actual positions of the hand. Although we used an inertia and coherence based estimation heuristic to smooth the trajectory, it still can not be as accurate and precise as the motion data.

On the other side, closer scrutiny at the disagreement points shows our networks sometimes work even better. For example, in a Sustained movement our LMA notators tried to perform Sustained Time to every segment of the movement, but they did not really do so (but just thought so) at some points, particularly when the motion starts, ends, and transits. Our neural networks successfully recognize the different (or additional) qualities encoded at these points. Examining the animations generated by expert set qualities and by learned qualities, with the original motions shows that the learned qualities make the animation more natural (see Figures 6.8 and 6.9 [3]).

Finally, we also carried out some experiments to determine Effort qualities from single-camera video projections. Although the depth information cannot be uniquely determined by monocular vision, our experiments show that the 3D trajectories projected onto 2D images (from the single camera view) still, in many cases, preserve the presence of many low level motion factors. Trajectories of motion samples in Weight and Space dimensions are shown in monocular and stereo views, respectively (see Figure 6.10 and Figure 6.11).

Comparing the 2D trajectories with the 3D trajectories reveals that the first-order (relative velocity) and second-order features (relative accelerations and zero-crossings) are preserved in most of the motion segments. This implies that single-camera video projections may suffice to provide the many factors that characterize Effort parameters. However, in cases when the depth information plays a crucial role, the first-order and

[3] All the animation files are available in the CD-ROM attached to this document.

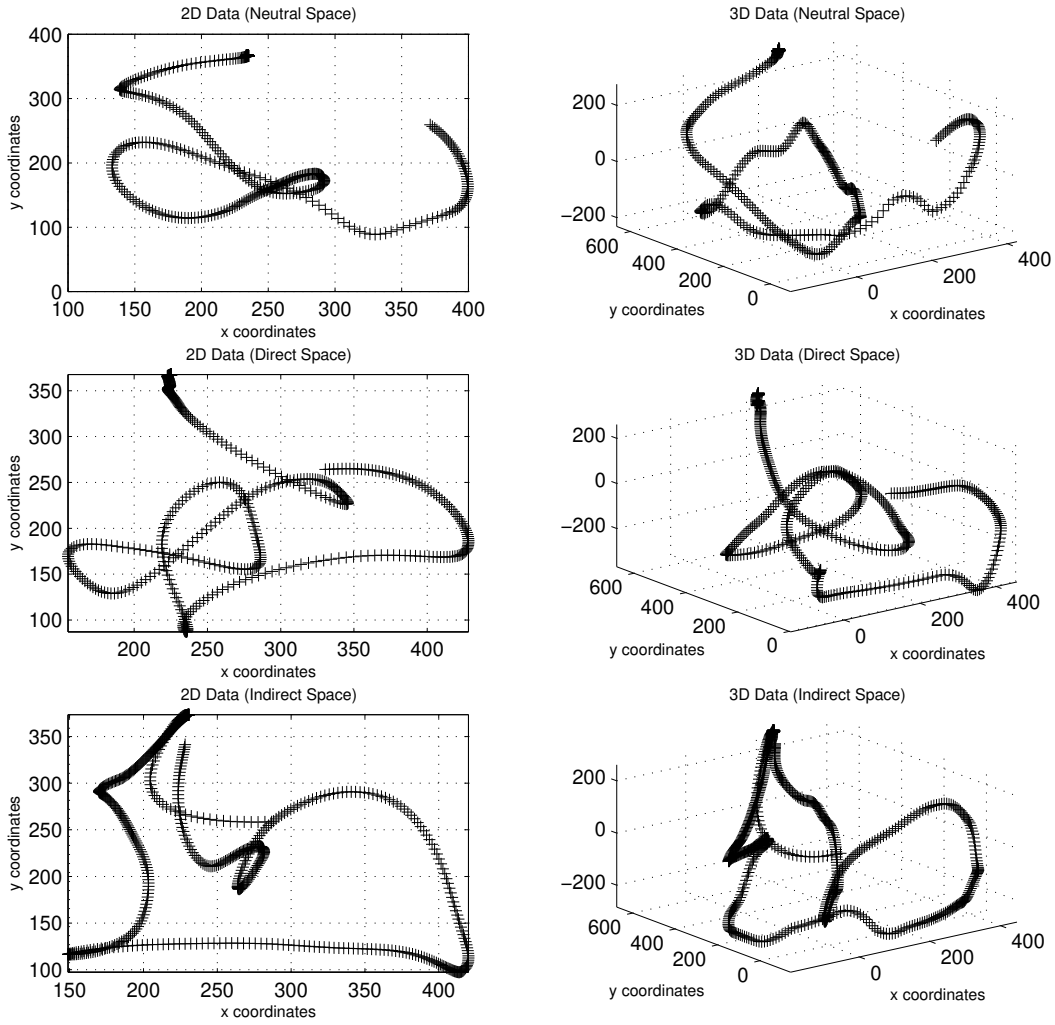

Figure 6.10: Motion trajectories in monocular and stereo views (Space dimension)

second-order features may be distorted in the 2D projections. The path of a Sudden-Direct motion towards the single camera, for instance, may be recovered as a very short, slow moving trajectory (or a single point in the worst case). Although the size of the bounding box (of the hand) may change, the camera we used is not so sensitive, making it hard to accurately measure the difference of the size of the bounding box.

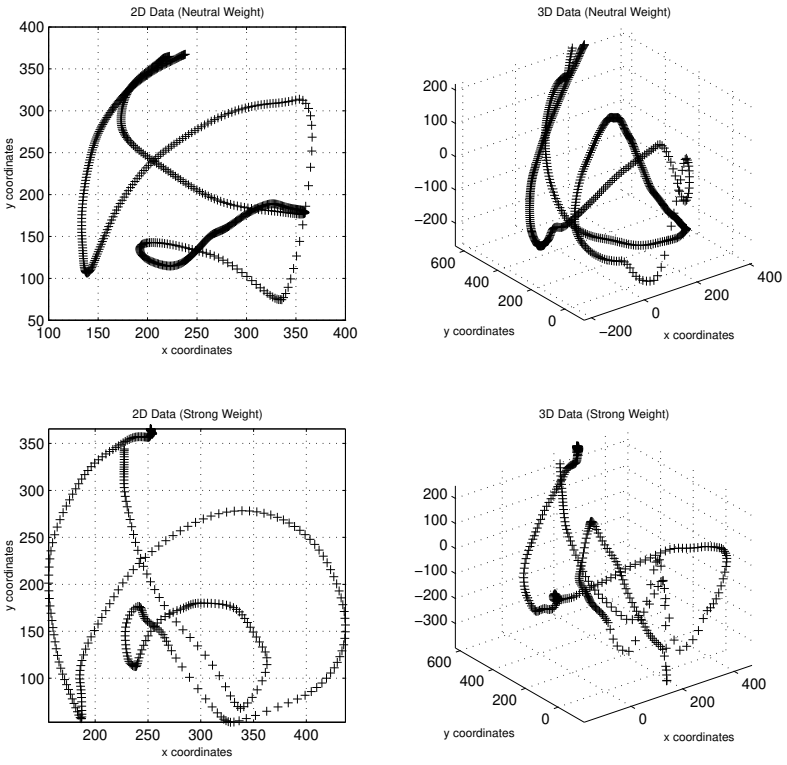

Figure 6.11: Motion trajectories in monocular and stereo views (Weight dimension)

Furthermore, the neural networks trained with 3D data are not directly applicable to single-camera projections. Instead, a separate data set comprising 2D video projections should be acquired from extensive experiments. To train the networks, a more careful approach should be considered, as the ground truth provided by the LMA notators are based on 3D data while the first-order and second-order features are computed based on 2D data. When the first-order and second-order features are distorted, the training data samples can become noisy. Therefore, additional information need to be employed. Possible considerations include the blurriness [107] and optic flow [37], or the redundancies

in a kinematic limb model [110, 56, 114, 49]. However, none of these approaches is without problems, and their computational complexity may prevent from a real-time implementation for the time being. Determining Effort qualities from a single-camera reliably and consistently is yet to be further explored.

Chapter 7

Conclusions and Future Work

In this thesis we have developed a framework for the procedural generation of expressive and natural-looking gestures for computerized communicating agents. This approach goes beyond the realm of psychology and linguistics based approaches by exploring the domain of movement observation science, specifically Laban Movement Analysis and its Effort and Shape components. This approach uncovers movement qualities which can be combined together to reveal different manners. We have also worked the opposite approach where the observable characteristics of gestures, including key poses, timing, and Effort parameters, can be extracted from live 3D and 2D data inputs. The two approaches combined give us the capability of automating the process and producing realistic and natural gestures for virtual agents from a sequence of video images.

7.1 Future Work

- Although Shape parameters have proven to be effective in animating expressive torso movements, further investigations should be carried out to identify how Effort qualities are manifested in the torso. A highly detailed, life-like human model with a deformable torso structure in particular should be used to further enhance realism.

- Recovering Shape parameters from live inputs has been essentially ignored in this work. There has recently been a series of efforts on the three-dimensional shape estimation of a moving human body [110, 56, 114, 49]. However, most of

the approaches used approximate rigid models of the human body, for instance, generalized cylinders. An immediate difficulty employing such models is that the rigid models can not easily adapt to different body sizes. To overcome problems that stem from using approximate models for the estimation of 3D human motion, Kakadiaris and Metaxas [63] have developed a method for the estimation of the parts of human body and their shape from multiple cameras based on a set of *controlled* motions, designed to reveal the body structure. This method allows the accurate estimation of the shape of the body parts of a particular subject and can be subsequently used for tracking the motion in 3D. DeCarlo and Metaxas [38] have developed a framework for the integration of edges and optic flow within a deformable model. The first step towards Shape parameter recovery would be to apply these aforementioned methods to the torso.

- Experiments of neural network based motion quality learning on subjects other than the training subjects should also be carried out to further evaluate the networks' generalization performance.

- Through Effort and Shape parameters and our agent model we suggest a plan for modeling the effects of agent mood, affect and personality. In the views of LMA researchers, an extroverted individual has a predisposition for some active Effort and Shape parameters—she uses the Effort and Shape exertions for affective functions and expressions more frequently, and perhaps to a larger degree, than an introverted person does. A key aspect of this approach is that the relationships between personality and LMA Effort and Shape qualities are not numerically fixed. A shy person may still yell a warning to a person in imminent danger. People without such ranges of expressive behavior may be (or appear to be) psychologically ill [12]. So it is more reasonable that personality defines certain set-points or statistical means for EMOTE parameters, and that the communicative context sets the variance and bias from the mean. During communicative acts, the actual EMOTE parameters used for gesture generation may vary within skew distributions defined by these means, biases, variances, and possibly *weights*. Agent mood may be represented by short duration repositionings of the means, but they gravitate to the personality means

over time and situations. The initial step toward this approach would be to identify the emotion/personality related variables described in the OCC model [101] and quantify these qualitative variables in terms of the EMOTE parameters.

- Finally we believe the video-based gesture acquisition system could be utilized in some behavioral or psychological experiments. The system could be further incorporated with an EMOTE based facial expression recognition system. The integrated system may prove to be effective in studies such as deception detection and discrepancy interpretation. According to the study done by Ekman [42], liars control the signals judged more informative and for which speakers are judged more responsible (mostly speech content and facial expression) but they pay little attention to vocal intonation and body movement. Thus, deception can be detected more often from bodily and vocal cues than from facial and verbal cues. For instance, due to the anxiety and worry of being caught, subjects may display more Indirect and Bound gestures and postural shifts than facial expressions while lying, and such discrepancies may not be very obvious while the subjects are telling the truth. Extracting EMOTE qualities from the gestural movements and correlating them with the ones extracted from the facial expressions may help to reveal the contrast between the "contrived" and the "spontaneous" in the subject's behavior.

7.2 Contributions

This thesis is not a simple elaboration of ongoing traditional work, but a new direction that may broaden our approaches to gestures, introduce a variety of complexities, and generate some new results. In the thesis, gesture synthesis is cast as a procedural animation problem, and gesture acquisition as pattern recognition. The two processes are then combined together through an agent model so that an engaging, expressive, believable virtual agent can be created. Most of the tools employed in this work can be found in textbooks on computer graphics and computer vision, however, very little work has been done so far to apply these tools to analyze patterns manifested in communicative gestures. In particular, very little is known about what kinds of observable patterns tend to be the "constant core" in communicative gestures. We believe our approach, based on Laban

Movement Analysis (LMA) and its Effort and Shape components, has made a significant contribution in that the lessons learned in human movement science over the past seventy years can be computerized to capture and analyze the patterns so that a higher level of understanding of communicative gestures can be achieved. Experiments of this sort have not been conducted before and should be of interest not only to the computer animation and computer vision community but would be a powerful and valuable methodological tool for creating personalized agents.

Appendix A

Experimental Data

The process of acquiring, processing, and analyzing the motion data to select a set of motion features is quite tedious. All the data is original, being acquired through the live performance of our professional LMA notators. We have implemented a *MotionCap* Plug-in in Jack Toolkit^{TM} to visualize, process, and analyze the data. Fig. A.1 shows the system GUI.

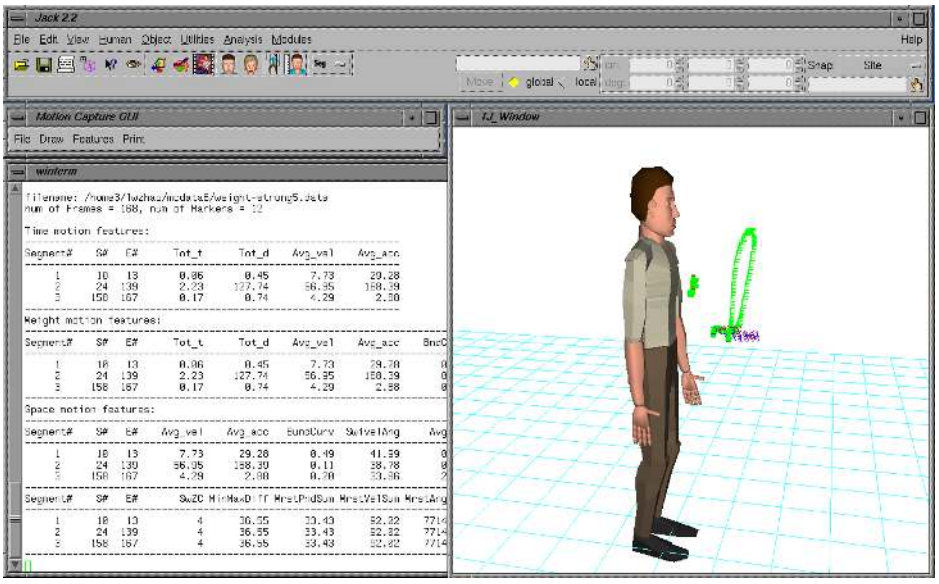

Figure A.1: Processing and analyzing the experimental data

The experimental data with basic Effort elements are listed in the following.

Q	F1 [a]	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
D	42.34	135.29	1.20	49.21	0.14	27.93	19.46	4	17.77	7.83	42.77	9.84	13.14	2
D	32.55	82.74	0.52	55.24	0.14	27.93	19.46	4	17.77	7.83	42.77	9.84	13.14	2
D	56.11	123.57	1.20	44.78	0.07	36.96	23.19	3	20.44	35.86	118.10	46.06	70.10	1
D	54.90	157.72	1.20	66.03	0.12	94.05	34.65	2	29.06	16.83	47.93	24.22	29.79	2
D	53.02	141.69	1.05	61.40	0.15	94.05	34.65	2	29.06	16.83	47.93	24.22	29.79	2
D	50.70	128.94	1.20	64.67	0.03	24.01	16.52	3	10.75	30.33	110.84	31.08	38.33	1
D	88.78	191.01	0.38	36.81	0.01	49.35	20.67	4	17.81	44.89	130.34	26.38	46.52	2
D	44.83	127.93	1.20	28.82	0.06	55.20	53.79	3	33.11	28.70	82.05	39.11	35.84	2
D	46.17	83.11	1.04	33.44	0.10	55.20	53.79	3	33.11	28.70	82.05	39.11	35.84	2
D	53.91	135.52	1.20	36.39	0.06	39.47	14.72	5	16.35	56.19	82.40	63.50	66.41	3
D	45.64	93.01	0.46	32.40	0.06	39.47	14.72	5	16.35	56.19	82.40	63.50	66.41	3
D	93.53	123.65	0.95	47.66	0.03	67.08	60.33	4	32.87	54.72	114.02	13.64	30.89	3
D	82.25	109.82	0.64	50.96	0.06	67.08	60.33	4	32.87	54.72	114.02	13.64	30.89	3
D	71.29	131.96	1.20	25.84	0.07	22.94	9.64	4	10.83	66.58	149.20	70.06	73.67	2
D	69.73	121.84	1.20	29.29	0.06	70.62	38.92	4	21.60	8.39	56.48	53.97	66.63	1
N	38.74	72.86	0.22	40.63	0.22	40.99	26.58	6	23.14	29.29	113.24	41.54	43.97	2
N	18.32	33.25	0.37	52.44	0.37	40.99	26.58	6	23.14	29.29	113.24	41.54	43.97	2
N	32.17	64.31	0.30	45.57	0.10	61.47	36.98	6	30.51	80.08	164.20	44.73	49.93	3
N	39.12	59.73	0.70	57.94	0.05	108.05	67.23	4	37.14	24.40	38.43	67.47	42.09	2

Table A.1: Experimental data used for training, validating, and testing the Space network (to be continued)

[a]Motion features used for the Space network are, F1: average velocity, F2: average acceleration, F3: average corner curvature, F4: average swivel angle, F5: average torsion, F6: swivel angle velocity, F7: high frequency of swivel angles, F8: swivel angle zero-crossings, F9: min/max swivel difference, F10: wrist pendulum, F11: wrist angle velocity, F12: sum of wrist angles, F13: max wrist angle, F14: wrist angle zero-crossings. D, N, and I represent Direct, Neutral and Indirect Space respectively.

Q	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
N	47.55	73.22	0.40	53.96	0.04	113.34	62.67	3	57.37	28.34	29.50	64.46	37.50	2
N	27.94	40.09	0.08	33.32	0.26	97.02	32.41	5	28.71	14.38	103.03	57.79	81.24	2
N	42.06	79.31	0.36	31.35	0.04	66.13	35.27	5	27.66	27.77	155.03	66.59	70.34	2
N	32.40	58.11	0.36	29.39	0.13	66.13	35.27	5	27.66	27.77	155.03	66.59	70.34	2
N	43.75	76.20	0.37	27.05	0.04	49.23	23.46	5	15.16	37.57	119.36	64.53	76.95	2
N	31.37	81.98	0.19	42.35	0.02	55.41	31.18	5	27.81	88.14	137.24	42.31	42.64	4
N	46.40	88.52	0.43	25.59	0.04	48.79	47.44	5	24.45	14.30	121.37	98.43	68.36	1
N	27.65	41.58	0.13	35.49	0.14	99.07	72.33	6	47.63	84.32	158.37	99.22	61.76	4
I	26.99	45.76	0.14	41.21	0.25	89.10	73.55	9	46.17	119.65	313.95	71.64	72.93	6
I	22.58	42.33	0.08	46.26	0.08	106.23	55.90	7	46.88	125.43	252.91	80.82	65.54	3
I	36.89	65.17	0.21	65.17	0.08	123.43	95.73	6	62.79	76.68	243.71	87.62	87.43	5
I	33.41	56.42	0.14	33.46	0.12	156.32	105.06	10	58.83	164.30	430.71	89.50	87.21	5
I	35.35	62.61	0.20	36.42	0.05	77.94	46.22	6	33.43	107.60	202.28	87.12	83.32	4
I	25.80	44.29	0.23	32.60	0.16	123.72	101.65	9	55.02	139.96	281.39	117.65	87.41	3
I	27.34	50.94	0.30	39.76	0.10	123.72	101.65	9	55.02	139.96	281.39	117.65	87.41	3
I	38.45	70.65	0.11	34.16	0.04	87.72	36.59	5	35.05	99.49	288.17	90.16	79.56	3
I	44.64	64.89	0.08	60.33	0.08	154.56	81.90	5	67.67	152.71	431.16	68.35	77.18	4
I	39.39	72.30	0.24	31.59	0.10	47.35	28.38	10	18.50	300.22	389.59	92.19	89.46	9
I	32.54	65.33	0.25	45.43	0.09	157.15	93.86	5	87.10	252.92	534.38	110.45	86.93	8

Table A.2: Experimental data used for training, validating and testing the Space network (continued)

Q	F1	F2	F3	F4
N	1.67	44.44	26.42	46.31
N	1.65	47.61	28.66	50.77
S	2.23	41.66	18.60	24.64
S	2.59	45.43	17.42	21.02
Q	0.48	38.55	76.90	303.98
Q	0.46	36.04	74.98	287.27
N	1.74	65.64	37.31	67.37
N	1.63	63.99	38.96	77.25
Q	0.41	35.47	83.65	352.50
Q	0.39	37.16	92.28	347.22
S	4.26	68.88	16.12	13.24
S	4.41	59.55	13.45	9.58
N	3.08	122.22	39.49	57.08
Q	0.81	92.40	111.43	394.18
S	4.55	110.55	24.21	21.86
N	1.90	54.85	28.73	38.56
N	2.15	63.21	29.22	36.08
Q	0.62	56.37	88.54	334.89
Q	0.56	47.57	82.53	297.41
S	4.67	70.34	15.03	12.87
S	4.92	63.64	12.91	8.74
N	1.22	34.37	27.98	42.95
N	1.80	51.94	28.62	47.55
Q	0.39	28.19	69.73	280.00
Q	0.46	33.70	70.06	284.08

Q	F1	F2	F3	F4
S	3.21	51.58	15.99	17.90
S	3.14	53.40	16.95	15.53
N	3.56	125.75	35.14	68.97
Q	0.50	33.62	65.83	297.98
Q	0.74	51.91	69.07	280.24
Q	0.77	64.93	82.32	341.65
Q	0.58	34.66	58.97	261.24
N	1.49	41.18	27.46	58.15
Q	0.41	27.32	66.43	203.03
S	3.15	44.69	14.12	15.34
N	2.46	78.45	31.70	57.27
S	4.49	86.24	19.13	22.99
N	1.70	51.42	29.93	45.21
Q	0.60	47.55	77.09	256.21
S	3.06	51.65	16.81	14.33
N	2.03	43.23	21.23	17.95
N	2.19	50.55	22.94	30.46
Q	0.56	46.23	79.96	300.92
Q	0.54	42.48	76.26	267.42
S	3.56	70.02	19.58	21.63
S	3.85	68.11	17.61	17.06
S	2.96	50.98	17.13	19.37
S	3.02	52.52	17.31	17.27
S	3.93	67.76	17.18	16.16
S	5.09	70.32	13.78	12.76

Table A.3: Experimental data used for training, validating and testing the Time network (S, N, Q are Effort Time quality Sustained, Neutral, and Sudden (Quick), respectively; Motion features used are, F1: total time, F2: total distance, F3: average velocity, and F4: average acceleration)

Q	F1	F2	F3	F4	F5	F6
L	2.03	54.17	26.46	37.38	0.39	1.61
L	2.21	68.11	30.67	42.94	0.13	1.03
L	0.54	7.43	13.49	34.67	0.52	1.60
N	2.03	43.82	21.41	31.20	0.32	0.24
S	1.97	63.02	31.73	53.38	0.26	4.70
S	1.72	71.85	41.38	51.36	0.33	6.41
L	1.82	56.50	30.82	42.96	0.57	2.68
L	1.90	54.93	28.74	35.30	0.48	3.57
N	1.67	63.15	37.57	68.50	0.35	0.21
N	1.51	66.90	43.87	83.67	0.20	0.35
S	1.34	69.13	51.20	129.04	0.25	4.11
S	1.49	64.81	43.08	96.46	0.34	3.77
L	0.17	1.65	9.37	30.08	0.41	1.65
L	0.21	1.35	6.27	15.19	0.69	1.50
L	4.38	157.31	35.83	35.49	0.08	5.96
N	2.69	118.25	43.66	71.83	0.33	0.90
S	1.92	133.72	69.14	143.75	0.54	9.08
L	2.09	53.25	25.42	22.84	0.19	2.56
L	3.41	79.50	23.27	20.76	0.12	4.16
N	1.30	27.44	21.15	28.13	0.11	0.42
N	2.30	65.23	28.12	37.12	0.51	0.41
S	1.16	52.63	45.14	70.68	1.20	3.36
S	1.78	70.05	39.02	64.86	0.13	3.46
L	1.51	36.24	23.94	25.06	1.20	1.08
L	2.17	58.80	26.92	37.05	0.49	1.44
N	1.82	42.38	23.18	26.92	0.41	0.17
N	1.86	58.91	31.43	53.44	0.49	0.04
S	2.34	66.15	28.07	35.69	1.20	9.01

Table A.4: Experimental data used for training, validating and testing the Weight network (L, N, S are Effort Weight quality Light, Neutral, and Strong, respectively; Motion features used are, F1: total time, F2: total distance, F3: average velocity, F4: average acceleration, F5: corner curvature, F6: sternum height)

Q	F1	F2	F3	F4	F5	F6
S	1.92	65.56	33.94	48.60	0.46	8.77
L	6.10	125.59	20.55	25.95	0.07	6.58
N	3.04	108.17	35.50	62.24	0.05	0.61
S	2.23	127.74	56.95	168.39	0.11	6.22
L	3.47	83.63	24.05	40.86	0.06	7.97
N	2.28	56.87	24.77	38.65	0.15	0.20
S	1.72	113.59	65.26	230.02	0.15	10.96
L	2.19	88.98	20.61	21.36	0.03	2.17
N	2.05	73.17	35.53	46.67	0.03	0.25
S	1.88	93.14	49.21	143.65	0.17	2.51
L	2.40	55.24	22.91	23.18	0.03	1.65
N	1.90	53.19	27.81	41.96	0.41	0.42
S	1.16	57.84	49.13	103.43	0.69	10.47
L	3.06	92.45	30.09	31.67	0.32	8.73
L	2.71	94.94	34.91	29.98	0.13	9.91
N	2.34	55.10	23.38	27.62	0.44	0.27
N	1.76	55.63	31.32	44.76	0.67	0.82
S	1.53	64.66	42.17	51.01	0.24	2.61
S	1.97	84.87	42.66	57.55	0.55	2.13
L	3.04	54.88	17.97	21.07	0.42	1.54
L	3.00	52.06	17.27	21.57	0.38	1.64
L	4.12	101.73	24.61	38.93	0.09	5.28
L	2.34	74.43	31.60	39.17	0.35	4.58
L	2.23	71.69	32.00	38.90	0.55	4.08
L	3.50	107.43	30.56	57.46	0.07	1.13
N	2.42	42.40	17.41	24.04	0.53	0.31
N	2.52	44.51	17.59	22.82	0.44	0.20
S	1.99	58.09	28.95	45.09	0.33	3.49
S	1.92	55.29	28.66	31.87	0.36	2.89
S	0.50	45.14	87.01	271.65	1.20	5.53

Table A.5: Experimental data used for training, validating and testing the Weight network

Q	F1	F2	F3	F4	F5	F6	F7
B	2.32	46.87	20.04	26.99	0.73	0.02	0
B	2.23	50.21	22.41	24.91	0.43	0.01	0
F	1.01	46.39	45.43	126.90	0.32	0.75	2
F	1.01	52.13	51.09	126.43	0.22	0.70	2
N	2.17	45.51	20.85	28.47	0.52	0.12	0
N	2.15	46.75	21.62	27.20	0.53	0.12	0
B	3.17	68.10	21.34	25.35	0.42	0.08	0
B	2.79	67.46	24.07	29.55	0.42	0.10	0
F	1.12	79.23	69.59	185.67	0.27	0.63	1
F	1.06	77.30	71.78	159.20	0.10	0.82	1
B	4.51	126.78	28.00	33.26	0.33	0.05	1
N	2.83	57.17	20.11	23.58	1.20	0.08	0
N	2.46	59.68	24.12	29.45	0.48	0.12	0
N	3.39	134.55	39.53	50.98	0.09	0.08	1
B	4.74	59.41	12.49	9.66	0.30	0.00	0
B	3.17	62.31	19.55	17.80	0.50	0.05	0
F	1.36	62.42	45.70	84.60	0.16	0.77	1
F	1.47	63.42	42.77	59.95	0.10	0.40	1
N	3.17	59.49	18.65	19.11	0.29	0.13	0
N	3.54	60.11	16.90	15.91	0.45	0.11	0
B	2.46	49.08	19.88	19.72	0.36	0.05	0
B	2.48	56.45	22.65	30.59	0.23	0.02	0
F	1.14	65.10	56.21	147.09	0.53	0.88	1
N	1.70	43.20	25.27	27.87	0.54	0.14	0
N	2.01	58.04	28.60	43.00	0.44	0.14	0
B	4.22	133.44	31.50	51.95	0.13	0.03	0
F	2.25	119.35	52.86	140.71	0.13	0.46	2

Table A.6: Experimental data used for training, validating and testing the Flow network (F, N, B are Free, Neutral, and Bound Flow Effort, respectively; Motion features used are, F1: total time, F2: total distance, F3: average velocity, F4: average acceleration, F5: corner curvature, F6: PAD (percentage of accelerations and decelerations), F7: number of wrist angle zero-crossings)

Q	F1	F2	F3	F4	F5	F6	F7
N	4.92	153.03	31.02	46.06	0.26	0.11	1
B	1.82	29.92	16.43	16.80	0.44	0.07	0
N	2.32	47.03	20.14	31.45	0.32	0.10	0
B	2.92	82.97	28.23	48.50	0.23	0.04	0
F	1.32	87.20	65.91	140.60	0.04	0.62	3
B	2.11	47.39	22.31	24.69	0.32	0.01	0
F	0.74	41.16	55.70	103.92	0.40	0.79	1
N	2.03	50.37	24.63	24.45	0.07	0.20	0
B	2.77	45.49	16.41	11.13	0.32	0.01	0
B	3.52	62.34	17.63	15.45	0.55	0.01	0
F	1.49	69.12	45.90	95.97	0.35	0.59	2
F	1.51	79.40	52.06	101.17	0.23	0.76	2
N	1.68	32.74	19.42	17.88	0.25	0.11	0
N	2.96	59.75	20.07	22.74	0.51	0.11	0
B	3.16	41.97	13.25	13.47	0.47	0.01	0
B	2.88	46.14	15.92	15.08	0.28	0.03	0
F	1.39	49.09	34.86	75.53	0.42	0.52	1
F	1.49	56.31	37.39	71.51	0.39	0.33	1
F	0.72	32.69	45.49	104.71	0.24	0.68	3
F	1.61	65.06	40.16	55.65	0.48	0.45	3
F	1.69	105.35	62.12	163.36	0.04	0.76	2
F	0.85	56.54	65.30	163.08	0.45	0.89	1
F	0.47	14.46	30.95	93.39	0.45	0.52	2
F	1.74	64.98	37.01	54.58	0.45	0.23	2
N	2.59	40.54	15.54	20.27	0.57	0.11	0
N	2.17	44.36	20.33	27.11	0.28	0.16	0
N	1.61	38.76	23.99	31.37	0.26	0.20	0

Table A.7: Experimental data used for training, validating and testing the Flow network

Bibliography

[1] Alias|Wavefront, Toronto, Canada. *Using Maya Character Setup*, 3rd edition, 2000.

[2] 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 Human-Computer Communications Society, May 1996.

[3] Norman Badler. A computational alternative to Effort notation. In J. A. Gray, editor, *Dance Technology: Current Applications and Future Trends*. National Dance Association, VA, 1989.

[4] Norman Badler, Rama Bindiganavale, Jan Allbeck, William Schuler, Liwei Zhao, and Martha Palmer. A parameterized action representation for virtual human agents. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied Conversational Agents*, pages 256–284. MIT Press, 2000.

[5] Norman Badler, Diane Chi, and Sonu Chopra. Virtual human animation based on movement observation and cognitive behavior models. In *Proceedings of Computer Animation 99*, pages 128–137. IEEE Computer Society Press, 1999.

[6] Norman Badler, Martha Palmer, and Rama Bindiganavale. Animation control for real-time virtual humans. *Communications of the ACM*, 42(8):64–73, August 1999.

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

[8] Norman Badler and Stephen Smoliar. Digital representation of human movement. *ACM Computing Surveys*, 11:19–38, March 1979.

[9] Norman I. Badler. *Temporal Scene Analysis: Conceptual descriptions of object movements.* PhD thesis, CS Dept., University of Toronto, 1975.

[10] Gene Ball and Jack Breese. Emotion and personality in a conversational agent. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied Conversational Agents*, pages 189–219. MIT Press, 2000.

[11] 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.

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

[13] Joseph Bates. The rote of emotion in believable agents. *Communications of the ACM*, 37(7):122–125, July 1994.

[14] Thomas Baudel and Michael Beaudouin-Lafon. Charade: Remote control of objects using free-hand gestures. *Communications of the ACM*, 36(7):28–35, July 1993.

[15] R. Beale and A. Edwards. Recognizing postures and gestures using neural networks. In R. Beale and J. Finlay, editors, *Neural Networks and Pattern Recognition in Human Computer Interaction*, 1992.

[16] Pascal Becheiraz and Daniel Thalmann. The use of nonverbal communication elements and dynamic interpersonal relationship for virtual actors. In *Proceedings of Computer Animation 96*, pages 58–67. IEEE Computer Society Press, June 1996.

[17] Pascal Béicheiraz and Daniel Thalmann. A behavioral animation system for autonomous actors personified by emotions. In *Proceedings of First Workshop on Embodied Conversational Characters (WECC 1998)*, pages 57–65, October 1998.

[18] Rama Bindiganavale. *Building Parameterized Action Representations from Observation.* PhD thesis, CIS Dept., University of Pennsylvania, 2000.

[19] Rama Bindiganavale, William Schuler, Jan Allbeck, Norman Badler, Aravind Joshi, and Martha Palmer. Dynamically altering agent behaviors using natural language instructions. In *Proceedings of Autonomous Agents 2000*, pages 293–300, 2000.

[20] Leslie Bishko. Relationships between Laban Movement Analysis and computer animation. In *Proceedings of the Dance and Technology Conference*, 1993. University of Wisconsin-Madison.

[21] Bruce M. Blumberg and Tinsley A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 47–54. ACM SIGGRAPH, Addison Wesley, August 1995.

[22] Richard Bolt. Put-That-There: Voice and gesture at the graphics interface. In *Proceedings of SIGGRAPH '80*, volume 14(3), pages 262–270. ACM SIGGRAPH, ACM Press, July 1980.

[23] Martin Brooks. The DataGlove as a man-machine interface for robotics. In *The Second IARP Workshop on Medical and Healthcare Robotics*, pages 1–12, September 1989. Newcastle upon Tyne, UK.

[24] Armin Bruderlin and Lance Williams. Motion signal processing. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 97–104. ACM SIGGRAPH, Addison Wesley, August 1995.

[25] Brian Butterworth and Uri Hadar. Gesture, speech and computational stages: A reply to McNeill. *Psychological Review*, 96:168–174, 1989.

[26] Tolga Capin, Igor Pandzic, Nedia Magnenat-Thalmann, and Daniel Thalmann. *Avatars in Networked Virtual Environments.* Wiley, Chichester, England, 1999.

[27] Justine Cassell. A framework for gesture generation and interpretation. In R. Cipolla and A. Pentland, editors, *Computer Vision in Human-Machine Interaction*, New York, 1998. Cambridge University Press.

[28] Justine Cassell. More than just another pretty face: Embodied conversational interface agents. In *Communications of the ACM*, volume 43(4), pages 70–78, 2000.

[29] Justine Cassell. Nudge nudge wink wink: Elements of face-to-face conversation for embodied conversational agents. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied Conversational Agents*, pages 189–219. MIT Press, 2000.

[30] Justine Cassell, Catherine Pelachaud, Norman Badler, Mark Steedman, Brett Achorn, Tripp Becket, Brett Douville, Scott Prevost, and Matthew Stone. Animated conversation: Rule-based generation of facial expression gesture and spoken intonation for multiple conversational agents. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94*, Computer Graphics Proceedings, Annual Conference Series, pages 413–420. ACM SIGGRAPH, ACM Press, July 1994.

[31] Diane Chi. *A Motion Control Scheme for Animating Expressive Arm Movements*. PhD thesis, CIS Dept., University of Pennsylvania, 1999.

[32] Diane Chi, Monica Costa, Liwei Zhao, and Norman Badler. The EMOTE model for Effort and Shape. In *Proceedings of SIGGRAPH '00*, Computer Graphics Proceedings, Annual Conference Series, pages 173–182. ACM SIGGRAPH, ACM Press, July 2000.

[33] Michael F. Cohen. Interactive spacetime control of animation. In *SIGGRAPH 92 Conference Proceedings*, volume 26(2) of *Annual Conference Series*, pages 173–182. ACM SIGGRAPH, July 1992.

[34] Michael M. Cohen and Dominic W. Massaro. Modeling coarticulation in synthetic visual speech. In N.Magnenat-Thalmann and D. Thalmann, editors, *Models and Techniques in Computer Animation*, pages 139–156. Tokyo: Springer-Verlag, 1993.

[35] 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.

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

[37] Douglas DeCarlo and Dimitris Metaxas. Deformable model-based shape and motion analysis from images using motion residual error. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 113–119, January 1998. India.

[38] Douglas DeCarlo and Dimitris Metaxas. Combining information using hard constraints. In *Proceedings of the IEEE Computer Society on Computer Vision and Pattern Recognition*, pages 132–138, June 1999. Fort Collins, CO.

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

[40] John E. Dietrich. *Play Direction.* Prentice Hall Publishers, New York, 1953.

[41] David Efron. *Gesture and Environments.* King's Crown Press, Morningside Heights, New York, 1941.

[42] Paul Ekman. *Telling Lies.* W.W. Norton, New York, NY, 1985.

[43] Paul Ekman and Wallace Friesen. The repertoire of nonverbal behavior: Categories, origins, usage, and coding. *Semiotica*, 1:49–98, 1969.

[44] Engineering Animation, Inc., Philadelphia. *Jack 2.2 Toolkit Reference Guide*, 1999.

[45] Sidney S. Fels and Geoffrey E. Hinton. Glove-Talk: A neural network interface between a DataGlove and a speech synthesizer. *IEEE Transactions on Neural Networks*, 4:2–8, January 1993.

[46] Pierre Feyereisen and Jacques-Dominique de Lannoy. *Gestures and Speech: Psychological Investigations.* Cambridge University Press, New York, NY, 1991.

[47] William T. Freeman and Craig D. Weissman. Television control by hand gestures. In *IEEE Intl. Workshop on Automatic Face and Gesture Recognition*, June 1995. Zurich, Switzerland.

[48] John Funge, Xiaoyuan Tu, and Demetri Terzopoulos. Cognitive modeling: Knowledge, reasoning and planning for intelligent characters. In *Proceedings of SIGGRAPH '99*, Computer Graphics Proceedings, Annual Conference Series, pages 29–38. ACM SIGGRAPH, ACM Press, August 1999.

[49] Dariu Gavrilla and Larry Davis. 3D model based tracking of humans in action: A multi-view approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, 1996.

[50] Michael Gleicher. Motion editing with spacetime constraints. In *Proceedings of Symposium on Interactive 3D Graphics*, pages 139–148. ACM Press, 1997.

[51] J.A. Graham and S. Heywood. The effects of elimination of hand gestures and of verbal codability on speech performance interpersonal behavior. *European Journal of Social Psychology*, 5:185–195, 1982.

[52] Brian Guenter, Charles Rose, Bobby Bodenheimer, and Michael F. Cohen. Efficient generation of motion transitions using spacetime constraints. In *Proceedings of SIGGRAPH '96*, pages 147–154. ACM SIGGRAPH, ACM Press, August 1996.

[53] Uri Hadar and Brian Butterworth. Iconic gestures, imagery and word retrieval in speech. *Semiotica*, 115:147–172, 1997.

[54] Barbara Hayes-Roth, Robert Gent, and Daniel Huber. Acting in character. In R. Trappl and P. Petta, editors, *Creating Personalities for Synthetic Actors*, 1997. Lecture Notes in CS, No. 1195, Springer-Verlag: Berlin.

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

[56] David Hogg. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[57] Berthold Horn. *Robot Vision*. The MIT Press, 1986.

[58] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[59] Nicholas R. Howe, Michael E. Leventon, and William T. Freeman. Bayesian reconstruction of 3D human motion from single-camera video. *To appear in Advances in Neural Information Processing Systems*, 12, 2001.

[60] Ann Hutchinson. *Labanotation*. Theater Arts Books, New York, 3rd edition, 1977.

[61] Mahesh Iyer and Russell Rhinehart. A method to determine the required number of neural-network training repetitions. *IEEE Transactions on Neural Networks*, 10(2):427–432, 1999.

[62] Finn V. Jensen. *An Introduction to Bayesian Networks*. Springer-Verlag, New York, NY, 2nd edition, 1996.

[63] Ioannis A. Kakadiaris, Dimitris Metaxas, and Ruzena Bajcsy. Inferring object structure in 2D from the deformation of apparent contours. *Journal of Computer Vision and Image Understanding*, 65(2):129–147, February 1997.

[64] Adam Kendon. Movement coordination in social interaction: Some examples described. In S. Weitz, editor, *Nonverbal communication*, pages 150–168, Oxford University Press, New York, 1974.

[65] Adam Kendon. Gesticulation and speech: Two aspects of the process of utterance. In M. R. Key, editor, *The Relation Between Verbal and Nonverbal Communication*, pages 207–227, Mouton, 1980.

[66] Adam Kendon. How gestures can become like words. In F. Potyatos, editor, *Crosscultural perspectives in nonverbal communication*, pages 131–141, Toronto, Canada, 1988.

[67] Adam Kendon. Does gesture communicate? A Review. *Research on Language and Social Interaction*, 2(3):175–200, 1994.

[68] Edward Klima and Ursula Bellugi. *The Signs of Language*. Harvard University Press, 1979.

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

[70] Teuvo Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, New York, NY, 1984.

[71] Shozo Kondo. Synthesis of hand-arm gestures. In C. Suen, editor, *Computer Recognition and Human Production of Handwriting.* World Scientific, 1989.

[72] Robert Krauss and Uri Hadar. The role of speech-related arm/hand gestures in word retrieval. In Lynn Messing and Ruth Campbell, editors, *Gesture, Speech and Sign.* Oxford University Press, 1999.

[73] Robert Krauss, Palmer Morrel-Samuels, and Francis Rauscher. The communicative value of conversational hand gestures. *Journal of Experimental Social Psychology*, 31:533–552, 1995.

[74] Myron Krueger. *Artificial Reality II.* Addison-Wesley, May 1991.

[75] Myron Krueger. Environmental technology: Making the real world virtual. *Communications of the ACM*, 36(7):36–37, July 1993.

[76] Rudolf Laban. *The Mastery of Movement.* Boston: Plays, Inc., Boston, 1971.

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

[78] Warren Lamb. *Posture and Gesture: An introduction to the Study of Physical Behavior.* Duckworth & Co., London, 1965.

[79] Warren Lamb and David Turner. *Management Behavior.* International Universities Press, Inc., New York, 1969.

[80] James Landay and Brad Myers. Applications of face and gesture recognition for human-computer interaction. In *Proceedings of the Sixth ACM International Multimedia Conference on Face/Gesture Recognition and their Applications*, pages 20–27, 1998.

[81] John Lasseter. Principles of traditional animation applied to 3D computer animation. In Maureen C. Stone, editor, *Proceedings of SIGGRAPH '87*, volume 21, pages 35–44. ACM SIGGRAPH, ACM Press, July 1987.

[82] Wayne Lea. Trends in speech recongition. In Wayne Lea, editor, *Components of a Speech Recognizer: Past and Present*. Prentice-Hall, Inc., 1980.

[83] James S. Lipscomb. A trainable gesture recognizer. *Pattern Recognition*, 24(9):895–907, 1991.

[84] Zicheng Liu, Steven Gortler, and Michael F. Cohen. Hierarchical spacetime control. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94*, Computer Graphics Proceedings, Annual Conference Series, pages 35–42. ACM SIGGRAPH, ACM Press, July 1994.

[85] A. Lofqvist. Speech as audible gesture. In W.J. Hardcastle and A. Marchal, editors, *Speech Production and Speech Modeling*, pages 289–322. Dordrecht: Kluwer Academic Publishers, 1990.

[86] Pattie Maes, Trevor Darrell, Bruce Blumberg, and Alex Pentland. The ALIVE system: Wireless, full-body interaction with autonomous agents. *Multimedia Systems*, 5:105–112, 1997.

[87] C. Maggioni. GestureComputer - New ways of operating a computer. In *Proceedings of Intl. Conference on Automatic Face and Gesture Recognition*, pages 166–171, June 1995.

[88] Vera Maletic. *Body, Space and Expression: The Development of Rudolf Laban's Movement and Dance Concepts*. Mouton de Gruyte, New York, 1987.

[89] David McNeill. So you think gestures are nonverbal? *Psychological Review*, 92:350–371, 1985.

[90] David McNeill. *Hand and Mind: What Gestures Reveal About Thought*. University of Chicago Press, Chicago, 2nd edition, 1995.

[91] David McNeill and E. Levy. Conceptual representations in language activity and gesture. In R. Jarvella and Klein, editors, *Speech, Place, and Action: Study in Deixis and Gesture*. John Wiley & Sons Ltd., 1982.

[92] Tom M. Mitchell. *Machine Learning*. MIT Press and McGraw-Hill Companies, Inc., Boston, MA, 1997.

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

[94] 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.

[95] Kouichi Murakami and Hitomi Taguchi. Gesture recognition using recurrent neural networks. In *Proceedings of CHI'91*, pages 237–242. ACM Press, 1991.

[96] Yanghee Nam and Kwangyun Wohn. Recognition of hand gestures with 3D, nonlinear arm movements. *Pattern Recognition Letters*, 18(1):105–113, 1997.

[97] Tsukasa Noma, Liwei Zhao, and Norman Badler. Design of a virtual human presenter. *IEEE Joural of Computer Graphics and Applications*, 20(4):79–85, 2000.

[98] James O'Brien, Victor Zordan, and Jessica K. Hodgins. Combining active and passive simulations for secondary motion. *IEEE Computer Graphics and Applications*, 20(6):86–96, July - August 2000.

[99] Jimena Olveres, Mark Billinghurst, Jesus Savage, and Alistair Holden. Intelligent, expressive avatars. In S. Prevost J. Cassell and E. Churchill, editors, *Workshop on Embodied Conversational Agents*, pages 47–55, October 12-15, Tahoe City, CA, 1998.

[100] Alan Oppenheim and Ronald Schafer. *Discrete Time Signal Processing*. Englewood Cliffs, Prentice Hall, NJ, 1988.

[101] Andrew Ortony, Gerald L. Clore, and Alan Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1988.

[102] Sharon Oviatt. Mutual disambiguation of recognition errors in a multimodal architecture. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'99)*, pages 576–583. ACM Press, 1999.

[103] Sharon Oviatt and Philip Cohen. Perceptual user interfaces: Multimodal interfaces that process what comes naturally. *Communications of the ACM*, 43(3):45–53, 2000.

[104] Young Park, Thomas Murray, and Chung Chen. Predicting Sunspots using a layered perceptron neural network. *IEEE Transactions on Neural Networks*, 7(2):501–505, 1996.

[105] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible of Meaning*. Morgan Kaufmann, San Mateo, California, 2nd edition, 1991.

[106] Catherine Pelachaud, Norman Badler, and Mark Steedman. Generating facial expressions for speech. *Cognitive Science*, 20(1):1–46, 1996.

[107] Alex Pentland. A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):523–531, 1987.

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

[109] Ken Perlin and A. Goldberg. *Improv*: A system for scripting interactive actors in virtual world. In *Proceedings of SIGGRAPH '96*, pages 205–216. ACM SIGGRAPH, August 1996.

[110] James M. Rehg and Takeo Kanade. Visual tracking of high DOF articulated structures: An application to human hand tracking. In Jan-Olof Eklundth, editor, *Proceedings of Third European Conference on Computer Vision*, pages 35–46, May 1994. Stockholm, Sweden.

[111] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In Maureen C. Stone, editor, *Proceedings of SIGGRAPH '87*, volume 21, pages 25–34. ACM SIGGRAPH, ACM Press, July 1987.

[112] Bernard Rimé. The elimination of visible behavior from social interactions: Effects on verbal, nonverbal and interpersonal behavior. *European Journal of Social Psychology*, 12:113–129, 1982.

[113] Bernard Rimé and Loris Schiaratura. Gesture and speech. In R.S. Feldman & B. Rimé, editor, *Fundamentals of Nonverbal Behavior*, pages 239–281. Cambridge University Press, 1991.

[114] Karl Rohr. Towards model-based recognition of human movements in image sequences. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 59(1):94–115, 1994.

[115] Anya P. Royce. *Movement and Meaning: Creativity and Interpretation in Ballet and Mime*. Indiana University Press, Bloomington, 1984.

[116] Dean Rubine. *The Automatic Recognition of Gestures*. PhD thesis, Computer Science Department, Carnegie Mellon University, 1991.

[117] David Rumelhart, Bernard Widrow, and Michael Lehr. The basic ideas in neural networks. *Communications of the ACM*, 37(3):87–92, 1994.

[118] John C. Russ. *The Image Processing Handbook*. CRC Press, 3rd edition, 1998.

[119] Edward J. Rzempoluck. *Neural Network Data Analysis Using SimulNet*. Springer-Verlag, New York, NY, 1998.

[120] Jakub Segen and Senthil Kumar. GestureVR: Vision-based 3D hand interface for spatial interaction. In *Proceedings of the Sixth ACM International Conference on Multimedia*, pages 455–464, September 13-16, Bristol, UK, 1998.

[121] Scott Steketee and Norman Badler. Parametric keyframe interpolation incorporating kinetic adjustment and phasing control. In B. A. Barsky, editor, *Proceedings of SIGGRAPH '85*, volume 19, pages 255–262. ACM SIGGRAPH, ACM Press, July 1985.

[122] David Sturman and David Zelter. A survey of glove-based input. *IEEE Computer Graphics and Applications*, 14:30–39, January 1994.

[123] Scott Sutherland and Lucy Venable. LabanWriter 2.0: A computer dance notation software. Internal Report, Department of Dance, The Ohio State University, 1990.

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

[125] Kristinn Thórisson. Face-to-face communication with computer agents. In *AAAI Spring Symposium on Believable Agents*, pages 86–90, March 1994. Stanford University, CA.

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

[127] Deepak Tolani, Ambarish Goswami, and Norman Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical Models*, 62:353–388, 2000.

[128] Xiaoyuan Tu and Demetri Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94*, Computer Graphics Proceedings, Annual Conference Series, pages 43–50. ACM SIGGRAPH, ACM Press, July 1994.

[129] Munetoshi Unuma, Ken Anjyo, and Ryozo Takeuchi. Fourier principles for emotion-based human figure animation. In Robert Cook, editor, *Proceedings of SIGGRAPH '95*, Annual Conference Series, pages 91–96. ACM SIGGRAPH, Addison Wesley, August 1995.

[130] Hannes Vilhjalmsson and Justine Cassell. BodyChat: Autonomous communicative human figure animation. In *Proceedings of the 2nd International Conference on Autonomous Agents*, pages 269–277, May 1998.

[131] Christian Vogler and Dimitris Metaxas. Adapting Hidden Markov Models for ASL recognition by using three-dimensional computer vision methods. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 156–161, 1997.

[132] Christian Vogler and Dimitris Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *Proceedings of ICCV'98*, pages 363–369, 1998.

[133] Lodewyk Wessels and Etienne Barnard. Avoiding false local minima by proper initialization of connections. *IEEE Transactions on Neural Networks*, 3(6):899–905, 1992.

[134] Alan D. Wexelblat. A feature-based approach to continuous-gesture analysis. MS Thesis, MIT Media Arts and Sciences, 1994.

[135] Bernard Widrow, David E. Rumelhart, and Michael A. Lehr. Neural networks: Applications in industry, business and science. *Communications of the ACM*, 37(3):93–105, March 1994.

[136] Andrew Witkin and Michael Kass. Spacetime constraints. In *SIGGRAPH 88 Conference Proceedings*, volume 24(4) of *Annual Conference Series*, pages 159–168. ACM SIGGRAPH, August 1988.

[137] Andrew Witkin and Zoran Popović. Motion warping. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 105–108. ACM SIGGRAPH, August 1995.

[138] Liwei Zhao and Norman Badler. Gesticulation behaviors for virtual humans. In *Proceedings of The 6th Pacific Graphics on Computer Graphics and Application*, pages 161–168. IEEE Computer Society Press, October 1998.

[139] Liwei Zhao, Monica Costa, and Norman Badler. Interpreting movement manner. In *Proceedings of Computer Animation 2000*, pages 98–103. IEEE Computer Society Press, May 2000.

[140] Liwei Zhao, Karin Kipper, William Schuler, Christian Vogler, Norman Badler, and Martha Palmer. A machine translation system from English to American Sign Language. In John S. White, editor, *Proceedings of the Fourth Conference of the Association from Machine Translation in the Americas*, pages 54–67. Berlin: Springer-Verlag, October 2000.

[141] Thomas G. Zimmerman and Jaron Lanier. A hand gesture interface device. In *Proceedings of ACM SIGCHI/GI'87*, pages 189–192, 1987.