# A Feature-Based Approach to Continuous-Gesture Analysis

by
Alan Daniel Wexelblat

B.S.E Computer Science, B.A.S Philosophy and Science
University of Pennsylvania
May 1984

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in Partial Fulfillment of the requirements of the degree of

MASTER OF SCIENCE in Media Arts and Sciences
at the
Massachusetts Institute of Technology
June 1994

© Massachusetts Institute of Technology, 1994
All Rights Reserved

Signature of Author _____

Program in Media Arts and Sciences
May 6, 1994

Certified By _____

Dr. Richard Bolt
Senior Research Scientist
Program in Media Arts and Sciences

Accepted by _____

Stephen A. Benton
Chairperson
Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Rotch
MASSACHUSETTS INSTITUTE
OF TECHNOLOGY
JUL 13 1994

# A Feature-Based Approach to Continuous-Gesture Analysis

by
Alan Daniel Wexelblat

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning on May 6, 1994,
in Partial Fulfillment of the requirements of the degree of
MASTER OF SCIENCE in Media Arts and Sciences

**Abstract:** Multimodal interfaces use a variety of modes of input, such as speech and gesture, to get information which is hard to acquire in traditional modes such as mouse and keyboard. This thesis describes the rationale, research, and implementation of a program — a gesture analyzer — which takes input from a sensor-equipped model of the human body and which produces a higher-level representation of the gesture in the context of a multimodal interface. Major innovations of this thesis are (1) enabling users to make continuous unrestricted gestures, a step forward in the technology comparable to the step from discrete-speech recognizers to connected-speech recognizers; and (2) semantic separation of gestural analysis from specific devices and from specific requirements of the interpreter, allowing the analyzer to be connected to any application. Gestures are analyzed in this system as they happen and a high-level representation is made available for interpretation. An important part of this work is that the analyzer is based on features of the gestures rather than using the limited template-based approach currently popular. The analyzer is embedded in a complex demonstration system which allows users to make natural gestures to specify spatial and temporal relationships such as "next to" or "like this" in the manipulation of system objects.

Thesis Supervisor: Dr. Richard Bolt
Title: Senior Research Scientist

This work was supported in part by Thompson-CSF and by the Advanced Research Projects Agency (ARPA) under Rome Laboratory contract F30602-92-C-0141.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring agencies or any other person or organization mentioned within.

# A Feature-Based Approach to Continuous-Gesture Analysis

by
Alan Daniel Wexelblat

The following people served as readers for this thesis:

Reader

Christopher Schmandt
Principal Research Scientist
Program in Media Arts and Sciences

Reader

Dr. Brenda Laurel
Member of the Research Staff
Interval Corporation

This thesis is dedicated
to the memory of

Henry Wexelblat, grandfather and inspiration

and

Martha (Marti) Hall, friend and teacher

# Acknowledgments

Thanks first and most to M. Elizabeth Hunter, who was there when it counted. Thanks to Dr. Richard Bolt for his leadership of the Advanced Human Interface Group, to Dr. Nicholas Negroponte for his leadership of the Media Lab, and to Dr. Stephen Benton for his direction of the Program in Media Arts and Sciences. Thanks to my teachers in the Lab and to my colleagues in the human-computer interaction community who have been both instructive and inspirational. Thanks to my family for their life-long support in my quest for education and for being proud of me and loving me as well. Thanks to my colleagues in AHIG: especially to Dave Koons for his work on the interpreter, his help in understanding the results of the experiment and his help in doing the math for the spatial transformations, and to Josh Bers for his work on the body model. Thanks to our undergraduate research assistants — Juno Choe, Greg Spurrier, Alberto Castillo and Chris Wren — for their help in programming the underlying substrates and device interfaces. Thanks to Shawn Becker for additional math help. Thanks to Kris Thorisson and Dave Tames for help with the pictures used in the thesis illustrations. Thanks to Carlton Sparrell and Dave Koons for bibliographic help. Thanks to my thesis readers for their feedback and suggestions for improvement. Finally thanks to Linda Peterson and the other people who make the entire Media Lab function and without whose help I could not be here to do this work.

This document was prepared in FrameMaker 4.0 on the Macintosh and Unix platforms. Times Roman typeface is used for body text; documents are reproduced in Helvetica and code in Courier. The pictures used for illustrations were taken with the Apple Quicktake 100 camera and uploaded with Quicktake software. Image manipulation was done in Adobe Photoshop and the results recorded as EPS files. Annotations to the pictures and additional diagrams were done with FrameMaker's internal drawing tools.

## Table Of Contents

## List Of Figures

## List Of Tables

# Chapter 1 Introduction

*Imagine that you've been involved in a car accident and must go to court. You meet with your lawyer to begin mapping out a strategy for the trial. You start by explaining that the truck "hit me from behind." Your lawyer says "So he rear-ended you." You shake your head. "Not exactly, it was sort of like this..." and you demonstrate, using your hands to model the progress of your vehicle and the truck, showing how it struck the right rear panel of your car.*

*Your lawyer nods, then asks you to stand in front of a large screen and repeat your gestures. First you create the scene on the computer by describing its layout: placing roads, shaping the tree that blocked your view of the oncoming traffic. Then you repeat your description of the incident itself and as a response, the computer animates a simple car and truck model, each moving in perfect imitation of your description, showing the impact as you remember it.*

*You work with your lawyer, changing the speed of the animation to more closely reflect your memory and play it back, viewing it from any position. In the end, you have a computerized representation of exactly what you remember happened.*

The scenario above shows an application of a multi-modal interface to a real-world problem. Gestures play an important part in this scenario, providing key information about location, method and timing of movement, and about spatial relationships among the objects being described. In a multi-modal interface, users go about their task, giving information to the system in whatever way seems most natural to them. Operating behind the scenes is the system software that takes input from gestures and other modes, correlates those inputs with the objects in the system — cars, trees, roads, and so on — and acts appropriately in response to the user input.

This thesis describes work that is implemented within the context of a larger project to enable multi-modal interaction of just this kind. The general approach and overall theoretical foundations on which this system is built are common to the Advanced Human Interface Group (AHIG) as a whole, though they take different expression in each of our work. Specifically, we are working to enable voice, eye-gaze, and gestural modes of input as part of a project to make multi-modal natural dialog possible with computer systems.

Each mode, or form, of input provides both unique and confirmatory information. Unique information for a given mode is that which is not present in modes other than the one under examination. Confirmatory information serves to verify or disambiguate information present in other modes. The purpose of capturing this information is to enable a more natural dialog with computer systems where information is provided in the forms most convenient for users. These modes provide information which, in many applications, is extremely awkward to capture with conventional input modes such as mouse and keyboard.

For example, in a site-planning application, an architect might wish to instruct the system to "put the loading dock next to the warehouse." The relation 'next to' is a categorical shorthand for a potentially infinite set of spatial relations any of which would be called 'next to.' To know which specific spatial relation is desired, conventional systems require users to either place the loading dock themselves (direct manipulation) or to specify the

spatial relationship in textual form (e.g. loading-dock.lower-right-corner.x-value = warehouse1.lower-left-corner.x-value - 1).

The textual form is cumbersome and requires the user to think in programmatic terms; this is contrary to our desire to have the computer system adapt to the user's abilities. In addition, the use of system terms and concepts ("warehouse1", "x-value") rather than user terms and concepts ("that warehouse", "there") where the target object is directly indicated) is generally accepted as bad design practice in the Human-Computer Interaction (HCI) discipline. The direct manipulation approach (Hutchins et al 1986) requires that users give up instructing the system in order to do all the work themselves, which can be tedious if there are more than a few objects to manipulate, or if the same task needs to be performed several times.

## 1.1 Thesis Focus

The work described in this thesis focuses on gestural input, specifically on gestures made with the hands. Our motivation is to build a computer system that can understand the natural gestures made by people (sometimes called gesticulation) without restricting the gesturer to a narrow set of predefined motions and hand configurations. To that end, we need a model which will allow us to continuously recognize gestures as the gesturer makes them and transform them into something which supports later use. The purpose of my research has been to provide such a model and to embed it in a software system which will perform conversationally acceptable (near-real-time) understanding of any gestures made by users.

In general, people are quite proficient at making multi-modal presentations involving speech, gaze, sketching and gesture. We do this in everyday conversation, moving fluidly among the modes as needed. The gestures used in these presentations are rapid, relatively continuous, and highly free-form. For the purposes of this research we distinguish empty-hand gestures from the strokes made with a pen or pointer and this research is focussed on empty-handed gesticulation.

Gesticulation of this sort is quite rich and expressive. As Kendon (Kendon 1986) notes, spoken languages must be constrained to the structure of some grammar and any relationship between this grammar and the subject being spoken about is coincidental at best. Gesticulative motions, though, can have a direct relationship to the subject. Gestures can depict action sequences, they can create pictorial diagrams. They can directly define spatial and temporal relationships. In this sense, there are more degrees of freedom for expression in gestures than in spoken utterances.

Gesticulation is not a language per se. In this respect it differs from sign languages such as ASL (American Sign Language). ASL is *generative* — new signs can be created, used, and adopted into the vocabulary. It is also *combinatorial* — existing signs can be merged to form a new sign and existing signs can be decomposed into more primitive elements in a hierarchical fashion. Neither of these properties applies to gesticulation. Gesticulation is also highly idiosyncratic, unlike ASL which has a proper standard form for its gestures. In

my experiments, which is described later in the thesis, people used a wide variety of gestures. When two people described the same scene they did not use the same gestures. Gestures also varied over time when made by one person. This two-dimensional variability makes input of natural gestures a particular challenge.

Gesticulation is also not rule-based; except in very rare cases it is impossible to speak of a gestural "error." Indeed, it is often the case that gestures contain correct information, even when speech contains errors. For example, my girlfriend frequently says "right" when she means "left" and vice versa. However, even when she says the wrong word, she always gestures in the correct (intended) direction.

Gesticulation, though difficult, is an expressive medium, used by people all over the world to convey ideas and information that cannot easily be carried in other modes. Indeed, many people find it hard or impossible to speak without using their hands. This intuition about the utility of gesture is supported by research (Graham & Argyle 1975) which demonstrated that people who were restricted from gesturing as they pleased had increased time in speaking and their partners performed worse on shared tasks.

People who cannot speak a common language often manage to communicate via gesture. It may even be argued, as McNeill does (McNeill 1992), that gesture is a form of communication closer to our thoughts than language. Surely the evolutionary development of gestures preceded the development of language and, as noted above, gestures sometimes contain better information than speech. All this serves to make gesture a fascinating area of study.

## 1.2 Thesis Layout

The rest of this document is organized as follows: The next section describes the problem addressed by this thesis. The solution to this problem is described in Chapter 3 "Solution", beginning with a description of a research experiment undertaken to gain experience with natural gesture. I then describe the key elements of the theoretical solution. The implementation is then described in detail, as well as the system framework in which the thesis demonstration system operates. The final chapter discusses results and insights into the theory gained from building the demonstration system and an analysis of related issues highlighted during implementation and operation.

# Chapter 2 Background and Scope

This thesis addresses the problem of recognition of gestures. As noted in the preceding chapter, this thesis concerns only unencumbered hand and arm gestures. There is a significant literature on pen gestures and pen-based interfaces; however, little of that applies to this research. We are concerned with the gestures made in giving descriptions to multimodal interfaces, which bear little resemblance to writing and drawing gestures. In fact, we frequently find ourselves picking up a pen to make sketches when speech and gesticulation have failed to convey our messages.

Within the realm of multi-modal interaction, there is also a great deal of work to be done on gestures made with other parts of the body if we wish to create systems which can truly understand human dialog. Our body language is rich with shrugs, head motions, facial expressions and other similar gestures. Little research has been done on understanding this kind of communication. While certainly worthwhile, it is outside the scope of this thesis.

Lastly this work is restricted to those gestures which people of normal ability use in actual descriptions and conversation. The human body is a flexible and highly variable device. It is capable of assuming a wide variety of configurations and motions. However, in everyday descriptive (narrative) interactions, people tend to make a fairly conventional set of gestures, compared to the vast variety of possibilities they could produce. As will be explained in this chapter and the next, it will be important to restrict the range of recognized gestures; naturally, the important set of gestures to recognize is the set that people commonly make in our target environments.

Similarly, while the research and prototype system described in this thesis are applicable to people with average motor capacities, there are a whole range of research questions applicable to people with various forms of mobility restriction or physical impairment that are not addressed here. We assume that our users will be of various sizes, shapes and genders, but will all have two hands, ten fingers, and be fluent speakers of standard English.

The next sections of this chapter give an overview of what gesture is, talk about different kinds of gesture and explain the terminology used in referring to gestures, with emphasis on the gestural taxonomy used by AHIG. This is followed by an examination of the general problem of gesture understanding from a computer point of view and finally a specific review of the "Iconic" gesture understanding system. The purpose of these sections is to show how this thesis attempts to solve general problems in the research domain by improving on the best current work.

## 2.1 Definition of Gesture

It is interesting to consider what constitutes a *gesture* out of the handwaving we seem to do all the time. The question, as it turns out, contains its own answer. A gesture is that thing which distinguishes itself from the background motions by virtue of something — a movement, a shape, etc. — that catches our attention. Gestures are like the proverbial tree

falling in the forest: if no one sees the gesture it is lost. Kendon expresses it thus (Kendon 1986, p. 26):

> *"...for an action to be treated as a 'gesture' it must have features which make it stand out as such."*

This definition is what we have used in attempting to produce systems which pick gestures out of the stream of input generated by people making multi-modal presentations.

Although this thesis is concerned with gestures, we concentrate largely on what is called *coverbal gesture* — that which occurs in the same temporal context as speech. Most often the gesture happens at the same time as speech, though as noted below (see "Psychological/Linguistic Research" on page 21) there are gestures that occur precisely because speech has failed.

### 2.1.1 Coverbal Gesture

When everything is working as expected for the speaker, researchers tend to agree that the gesture happens at the same time as the speech to which it is related, or slightly preceding it. In particular the stroke phase of gestures (which is always present) comes at the same time — or slightly preceding — the semantic high point of the speech. McNeill asserts that the stroke highlights the "conceptual focal point" of the utterance. In our demonstration systems we use temporal proximity of gesture to speech as an aid in determining what the gestures mean.

Nespoulous & Lecours identify three separate uses for coverbal gesture (from "Gestures: Nature and Function" p. 60):

1. "to illustrate ongoing verbal behavior" — the speaker says something and the hands demonstrate or point at what is being spoken about.

2. "to express personal emotions related to verbal behavior" — the feelings and emotions that are often carried in tone of voice can also be seen in some gestures.

3. "to lay emphasis upon specific verbal elements within discourse" — timing of gesture helps listeners focus on what is most important in what we say.

In coverbal gesture, the semantic content of the utterance is split between the speech and gestural modes. As noted in the introduction, each mode may contain unique or confirmatory information. We assume a cognitive model where gesture and speech share the burden of carrying the message and operate in complementary fashion: when one channel does not carry the message, the other channel does.

In a situation such as a conversation where multiple utterances occur, it is entirely possible that a given utterance may not have all modes present. For example, the speaker may first say something like "Move the picture over a little" and simultaneously make a finger-waving gesture[1] like the one in Figure 1 to indicate which direction to move the picture.

If the picture needs to be moved a little more, the speaker might then simply repeat the gesture without speaking; in a dialog much of the meaning can also be carried by the context of a gesture.

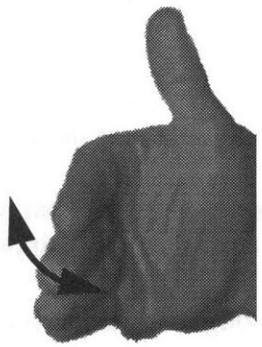

Figure 1. Finger-waving Gesture

## 2.2 State of the Art

The state of the art in gestural understanding is quite primitive. As noted in "Introduction" on page 15, gestures are rich, complex, and highly idiosyncratic. This makes gestural understanding by computers a particularly difficult problem. Three separate threads run through the field. In one thread (which is responsible for the taxonomic basis reviewed below), there is work by linguists, neurologists, therapists and others interested in the production of speech as it relates to brain function and thought processes. These researchers are unconcerned with computer implementation implications of their work; they are not systems builders. Their concern is largely with a performative understanding of gesture, especially as it relates to dysfunctional cases. Semiotic analyses of gesture have also been done (primarily by Nespoulous & Lecours), but again this is with an eye toward understanding the function of gesture in individual communication, not in a system-oriented context.

Although, as noted in Section 1.1 "Thesis Focus", gesticulation is not language-like, they are amenable to classification. A number of classification schemes have been proposed in the psychological and linguistic literature, all of which seem to be derived from Efron's initial work (Efron 1940). For our work we have found it necessary to rethink gestural classification issues and reorganize information into a form which is more suitable to computer processing. Efron's classification system and our modifications of it are covered in detail in Section 2.3.1 "Efron".

The second two threads of research on gesture operate within the HCI community. The first involves the use of gestures, frequently written gestures made while the person holds a stylus, as a portion of a specific interface to a specific system (e.g. Kurtenbach & Buxton 1991). Most of the pen-based systems available today fall in this category. These inter-

---

1. The reader is urged to put down the paper or take her hands off the keyboard and try this motion and others described throughout the thesis. Static pictures can give only a very poor approximation of what is meant by a particular gesture, whereas trying the gesture and seeing the result will give a much clearer idea of what is intended with the gestural examples. This problem seems to be common to all books which try to describe gestures.

faces are often called "multi-modal" and they usually use "modes" to provide additional channels of input, but often without deep thought having been given to the appropriateness of particular modes or with no deep understanding of how multi-modal input must be analyzed and interpreted in a dialog context.

The other research thread within the HCI community involves hand-based gestures as the primary form of interface to a system. This trend, which is most often seen in virtual worlds interfaces (VR), involves the use of a series of gestural commands or symbols. These commands are recognized by having the computer use templates — when the user's hand configuration matches a template, the associated command is invoked. This is analogous to using a set of function keys in that there is a one-to-one mapping done between gesture and meaning with no analysis or interpretation being applied. One popular application of this approach is the various attempts that have been made to capture ASL fingerspelling (Kramer & Leifer 1989)and similar rote gestures in aid of communication by deaf persons. The relevant computer research is reviewed in Section 2.4 "Computer/System Research".

The next two sections review the major literature in the relevant research areas. At the end of each section is a detailed discussion of the AHIG contribution to research in that area.

## 2.3 Psychological/Linguistic Research

Modern research on the use of gesture dates from Efron's work in the mid-thirties. Previous theorizing was not based on experiment or systematic observation and so is discounted by modern researchers.

In classical gesture theory (Efron 1940), each gesture consists of three phases: preparation, stroke, and retraction. In the preparation phase, the hands are raised to the location where the gesture will begin. In the stroke phase, the actual gesture is performed, and in the retraction phase, the hands are dropped to the sides or to other resting places. As with many classical models, this three-step description is only a very vague approximation of what happens in reality. It is easy to see from observation that the preparation and retraction phases may be missing. As will be discussed in Section 3.1 "Experiment", the variations people make from the ideal approximation gave important clues to my research.

Three main researchers have followed Efron's initial work in the general theory of speech and gesture — Kendon, McNeill & Levy and Rimé & Schiaratura[2]. Each has focussed on specific approaches while still considering the whole problem of coverbal gesture. The next sections discuss Efron's work and the people who followed him, including the AHIG gestural taxonomy developed by Koons and used in the present thesis.

---

2. There are also a number of specialized researchers whose work touches on gesture, but only in specific areas. For example, Miller (Miller 1982) investigated various theories of demonstrative reference, but his focus was on the underlying meaning rather than studying the gestures used. The specialized research is not reviewed in this thesis.

### 2.3.1 Efron

Efron's work began with observations of the conversational behaviors of Jewish (primarily from eastern European countries) and Italian immigrants in New York City. The most important feature that Efron brought to the categorization process was the orientation toward the referent of the gesture. This orientation has been retained in most of the literature that followed Efron (see "Psychological/Linguistic Research" on page 21).

Efron noted that although some gestures clearly had external referents (objects or events around the speaker), some had internal referents in the sense that they referred to something in the speaker's ideational process. He grouped gestures as enumerated below. Examples and illustrations of each of these gestures (as they correspond to the AHIG taxonomy) are shown in Section 2.3.6 "AHIG Categorization" and in Figure 3 on page 27 through Figure 5 on page 28.

- *batonlike*: these gestures are speech-marking; their referents are in the speech, and they either (1) stress some element of the speech, (2) introduce some new element into the utterance, or (3) divide the spoken sentence into chunks according to underlying reasoning.
- *ideographs*: these gestures traced the path of the speaker's thought; their referents were in the speakers ideational processes.
- *physiographic*: these gestures parallel the speech and present some kind of "figural representation" of the object(s) being spoken about. The referents of these gestures are in the content of the speech.
- *pantomimic*: these gestures also parallel speech, but serve to illustrate the function or manipulation of some object. The referents of these gestures are in the objects which are spoken about. These referents are sometimes indirect, as when one speaks about "cutting" and pantomimes the movement of a hand holding a knife.
- *symbolic/emblematic*: these gestures are representations which have one-to-one correspondence with meanings. They are devoid of any morphological relationship[3] with the thing being represented.

Identifying characteristics of these gestures categories are summarized in Table 1, "Summary of Gestural Classification Systems," on page 26.

### 2.3.2 Kendon

Kendon began his research by attempting to determine what people saw when they viewed gestures. His subjects viewed tapes of people speaking in a language they did not understand; however, the viewers felt that they had no trouble picking out gestures. Kendon determined that they were looking at significant "excursions" where the speaker's hands moved in certain ways that were perceived as deliberate.

---

3. That is, Efron asserted that symbolic gestures such as the circled thumb and forefinger ("OK") were simply arbitrary symbols and not derived from properties of the word they substitute for.

Kendon also theorized that (Kendon 1986, p. 30):

> *"...participants perceive each other's behavior in terms of a number of different systems of action — the deliberately communicative or gestural, the postural, the practical, the incidental and, perhaps... the emotional."*

The importance of this perception for Kendon was that gesture and speech were intimately related in that gestures appeared purposeful in that they accompanied and appeared to amplify speech, whereas similar movements in the absence of speech were not seen as gestures.

Kendon went on to investigate the relationship between what he called a "gesture phrase" and a "tone unit" of speech. A gesture phrase is a:

> *"... 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." (ibid, p. 34)*

Tone units are:

> *"...phonologically defined syllabic grouping with a single intonation tune." (ibid, p. 34)*

In Kendon's terms, the stroke of the gesture phrase occurred simultaneously with (or slightly preceding) the nucleus of the tone unit. His concern in drawing this point was the "dethronement" of spoken language from its position of uniqueness and primacy. Kendon strongly asserts that utterances are planned as integral units with gestural and verbal components. Kendon's classification of gestures is included in the summary table (Table 1) below. Note that Kendon uses the term gesticulation to refer only to rhythmic marking gestures, whereas this thesis uses that term as a synonym for all natural gestures.

### 2.3.3 McNeill & Levy

McNeill and Levy (and later McNeill alone) based their work around a theory that concrete sensory-motor models of reality are used by people making communications. Gestures are used to convey the forms contained in these models. Their experiments involved having subjects watch a cartoon and then narrate the action of the cartoon to other subjects who have not seen it. The experiment described in Section 3.1 "Experiment" is patterned after this work.

They also studied the relationship of gesture and narrative more closely than other researchers. They assert that narrative statements tend to be accompanied by iconic statements; this directly contradicts findings by Rimé & Schiaratura (discussed below). They agree with Kendon that speech and gesture are part of a coherent whole (McNeill & Levy 1982, p. 275):

> *"Contrary to the assumption which seems widespread, that gestures are part of a separate system of 'non-verbal communication,' only incidentally connected to speech, we find that gestures correlate closely with meaning on several levels[4] of language organization."*

McNeill and Levy are also unique among the psychological researchers in being the only ones to discuss gestural features (or other sub-gestural components) in their work. Their approach to features is similar to ours in some ways, but different in a number of important respects; this is discussed at length in Section 3.2.1 "Categorization Problems".

The categorization summary in Table 1 contains the more complete and detailed categories used by McNeill in his most recent work (McNeill 1992).

### 2.3.4 Nespoulous & Lecours

Nespoulous and Lecours looked at gestures from a semiotic point of view — that is, they took gestures to be linguistic signs. They looked at gestures as being divisible into four levels: Substance of Content, Form of Content, Form of Expression, and Substance of Expression. Each of these levels can be used to decompose and help understand gestures.

They also point out that the categorization of gestures can be done at any of these levels. The categorization schemes summarized in Table 1 are mostly at their substance of content/form of content level in that gestures are divided up according to the work they do.

Nespoulous & Lecours do not provide a comparable taxonomy; instead they classify gestures according to (Nespoulous & Lecours 1986, p. 55):

> *"the intrinsic nature of gestures within the context of a given semiotic system,"*

and on a second scale, according to:

> *"semiotic uses rather than to intrinsic abstract characteristics of gestural segments." (ibid)*

Much of Nespoulous and Lecours' theorization is influenced by their work with aphasic patients, which makes it hard to know how their work applies to people without such disabilities. In addition, their semiotic concerns lead to a more rigorous examination of the nature of gestures, which are seen as signs, than of their function.

### 2.3.5 Rimé & Schiaratura

Rimé & Schiaratura base their work on a pair of related experimental conditions: in the first condition, subjects were videotaped performing a standard digit-repetition task; in the second, they were recorded as they related to the experimenters (Rimé & Schiaratura, p. 239):

---

4. Specifically, McNeill & Levy distinguish the sentence and discourse levels of language activity.

*"...noteworthy events that had occurred in their personal life during the preceding week."*[5]

Subjects for both experiments were placed in conditions where they could see their conversation partner and where they could not. Interesting immediate findings were that (a) people gestured, even in the simple digit-repetition task; and (b) for both tasks gesture frequency did *not* decrease when the speaker could not see his conversation partner.

This lends strong evidence to the hypotheses which say that the purpose of gesture is more than just communicative. It also can be interpreted to support theories like McNeill's which state that communicative utterances are formed at a level in the brain that is neither language nor gesture, but which places meaning in both channels as needed.

As noted in Section 1.1 "Thesis Focus", Graham & Argyle did work in 1975 showing that people who are restricted from gesturing use more words, and give poorer directions (as measured by the performance of their partners on a shared task). Rimé & Schiaratura repeated this restriction-of-movement condition in their experiments and found similar results. They also analyzed the semantic content of the speech, and found that the vividness of imagery was significantly *decreased* when gestures were restricted. This result is the opposite of what would be expected, and strongly suggests that the gestural motor activities of speakers are linked to their verbal encoding activities.

Rimé & Schiaratura call this "embodied thinking."[6] This concept of embodied thinking and treatment of gestures as an integral part of an overall communicative act are strongly part of the AHIG approach. We reflect this in our attempts to build integrated systems (see "System Framework/Interaction Experience" on page 54) which use each of the modes in appropriate manners.

Interestingly, their subjects reported feeling that speech was less clear and less fluid when the speaking person used many gestures. This reflects the fact that in free-flowing fast conversation we tend to speak in incomplete sentences and rely on the totality of our presentation to get the audience to understand what we mean. However, speakers who used more gestures were rated as "warmer" and "more relaxed" and generally elicited a more positive response from viewers.

Rimé & Schiaratura also provide a review of the literature on gesture. They show that several conclusions about gesture are generally accepted by psycholinguistic researchers:

- The type of information communicated is an important determinant of ges-

---

5. Rimé & Schiaratura do not define "noteworthy," nor do they say what limits (if any) were put on the subjects' event descriptions.
6. They also take a very strong position against the behaviorist approach to language. They point out that if the motor effects of generating gestures affects speech content, then we lose the ability to distinguish between "stimulus" and "response" in the behaviorist sense. That is, part of the response (the gesture) becomes part of the stimulus for forming speech!

tural behavior. For example, when the speaker is attempting to communicate information that is largely pictorial in the mind, iconic (ideographic in Efron's terms) gestures are most likely to occur. In general, attempts to communicate visual, spatial or motoric information generate the most gestures and the most iconographic.

- Developmental data show that childrens' gestural abilities increase as they get older and their gestures grow in complexity and sophistication as their speech does.
- Gestures appear when there is some discrepancy between the unit of thought and the unit of speech. This is support for Kendon's assertion that gesture has more degrees of freedom than speech and that gesture can express different information than can speech.
- There is strong evidence for gestures preceding the relevant speech units in time —the amount of time precedence is variable, but no one has data showing gestures occur later than the related speech.

Table 1 shows a summary of the gesture classifications used by the major researchers in this field. As the table shows, all the psychologically-based researchers owe their classification systems to Efron's initial work.

**Table 1: Summary of Gestural Classification Systems**

| Kendon | McNeill & Levy | Rimé & Schiaratura | Efron | Identifying Characteristics |
|---|---|---|---|---|
| physiographic | iconic | physiographic | kineto-graphic | picture the content of speech |
| ideographic | metaphoric | iconic | ideographic | portray the speaker's ideas, but not directly the speech content |
| gesticulation | beats/Butterworths | speech-marking | baton | marking the rhythm of speech |
| autonomous gestures | symbolic | symbolic | symbolic/ emblematic | standardized gestures, complete within themselves, without speech. |
| — none — | deictic | deictic | — none — | pointing at thing/area; space around body used |

### 2.3.6 AHIG Categorization

Efron's research helps our general understanding of gesture; however, as discussed in "Experiment" on page 39, the orientation of this classification and of those who followed him toward a holistic human understanding of gestures can actually be detrimental. The

assumptions made within psychologically-oriented research do not necessarily apply to mechanical/computer systems, much as knowledge of how birds fly does not always aid our understanding of how to make aircraft fly.

This realization caused David Koons, another of the Research Assistants in the AHIG, to propose to me that we adopt a new categorization scheme for gestures[7]. After much discussion and debate, I agreed and have adopted his idea with relatively few modifications. We divide gestures into five major groups:

*Symbolic/Modalizing*

Symbolic gestures are the most familiar — V-for-victory, or the thumbs-up-OK gesture pictured in Figure 2. These gestures are also sometimes called "rote" because they are memorized and taught by rote repetition. Their connection to meanings may be well-known or lost in time, but they are always directly mapped to meanings, almost regardless of context. Symbolic gestures are the closest to templates.

Figure 2. Thumbs-up Symbolic Gesture

Modalizing gestures are ones which change the meanings of communication, usually of verbal communication. They control the "mode" of interpretation that conversation partners should apply. For example, an upturned palm or shrugged shoulders may indicated that the speaker is uncertain about what she is saying and does not want her words taken literally.

*Pantomimic*

As the name suggests, these gestures involve the person miming the use of objects and actions with objects. The hands form shapes which correspond to the shapes they take when manipulating the actual objects (see Figure 3), and make motions similar to those made when using the objects. Here the person's hands are curled as if gripping a steering wheel.

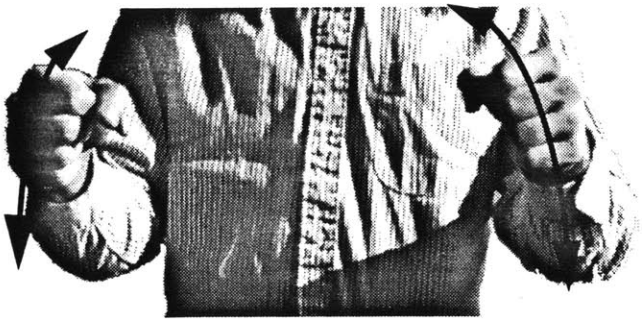

Figure 3. Driving Pantomimic Gesture

People making pantomimic gestures are demonstrating their knowledge of the affordances of objects. *Affordances* is a term used by Norman (Norman 1988) to describe the means

7. The precise content of this taxonomy is not yet recorded in published literature; however, the focus on computer interpretation is due to Koons' work on his Ph.D. dissertation.

of interaction — such as handles, buttons or shapes — that an object offers to the person interacting with it.

*Iconic (or Object)*

In these gestures, the hand "becomes" the object being demonstrated and is usually moved in such a way as to act out a scene or event, rather than a specific action within a scene, as with pantomimic gestures. An example of this is the use of the hands to show the relative positions of two colliding cars, as described in the scenario at the beginning of "Introduction" on page 15. A simpler example of this is shown in Figure 4 where the person is describing the fall of a tree; the left arm represents the ground and the right arm represents the tree (shown in mid-fall).

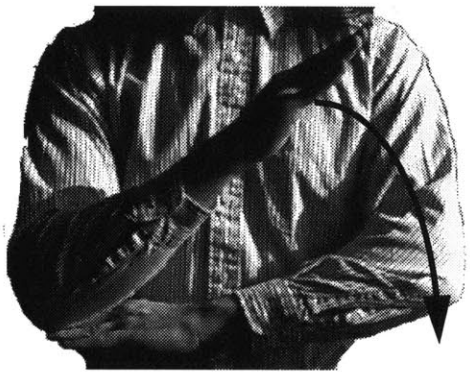

Figure 4. Tree-falling Iconic Gesture

*Deictic/Lakoff*

Deictic gestures are those gestures that point or indicate an object, location, direction, or group of objects. Deictics are the most commonly seen form of gestures used in interfaces, deriving initially from the "Put That There" system (Bolt 1980).

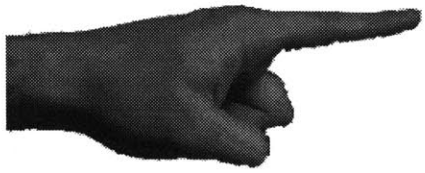

Figure 5. Pointing Deictic Gesture

Pointing gestures (as shown in Figure 5) which use the whole hand or arm are partly cultural; other cultures may use more subtle deictic gestures such as flicking one finger or tilting the head. In other cases, deictics differ based on the need for specificity. For example, if only two choices such as left or right are possible, a nod of the head may suffice. However, if more precision is needed — such as picking a person out of a crowd — an extended finger serves to sharpen the focus of attention.[8] In all cases, though, the body is forming a vector which intersects an object or marks out an area in the space around the gesturer.

Lakoff gestures, which do not appear in Efron's classification scheme, are named for the philosopher George Lakoff, who noted in his book *Metaphors We Live By* (Lakoff & Johnson 1980) that we frequently spatialize verbal metaphors (such as "I'm feeling down" to indicate an unhappy mood). These metaphoric utterances are often accompanied by gestures which show the directionality of the metaphor. In these cases —as with deictics — the gesturer is forming a relationship between his body and the surrounding space, though any object intersections are purely coincidental in these cases.

8. I am indebted to Dr. Bolt for providing and explaining this example

Lakoff gestures are conceptually more complex than conventional deictics; however, that complexity comes from post-hoc interpretation in which the metaphoric model suggested by Lakoff is applied to the gestural information. The important thing for classification, though, is that these are both *pointing* gestures. The application of Lakoff's model of metaphor spatialization can only be done based on speech or other context. For example, if you hear me say "I'm feeling down today," a downward-pointing gesture takes on a different meaning than if I say "That chair" or otherwise use some deictic reference.

We do not use the metaphoric gestures suggested by McNeill in his reading of Lakoff (and called ideographic by Kendon and by Efron). McNeill's metaphorics — such as the 'conduit' metaphor, in which motion of the hands is taken to be literally conveying an idea from gesturer to observer — are dependent on applying a further level of knowledge than we wish to admit. This objection is discussed in detail in Section 3.2.1 "Categorization Problems".

*Beat/Butterworths/Self-adjusters*

Beats are the strokes which mark the rhythm and pace of speech. They show which words the speaker thinks are important and help to direct the listener's attention. People who make beats which do not correspond to important thoughts are often seen as poor speakers; many of the lessons in public speaking involves helping people learn to control their beat gestures.

Butterworths are named for Brian Butterworth, the researcher who first studied them (Butterworth & Beattie 1978). They do not appear in Efron's classification scheme. Butterworth gestures are like beats, but they differ in that they mark not the important elements of speech, but rather their lack. Butterworths often look like grasping or reaching gestures, but they occur during non-semantic pauses in speech. Sometimes they consist of several beat gestures — classic "hand-waving" — and seem to be placeholders, used to tell conversation partners that we are not yet done speaking, even though we cannot think of the next word to say.

Self-adjusters is the name given to the various forms of gestural fidgeting we do: finger-tapping, hair-twirling, and so on. They are named for the fact that we appear to be consciously adjusting our appearance (straightening our ties or, as many fans have noted, the constant shirt-tugging that afflicts the characters on the television series *Star Trek: The Next Generation*). As with Butterworths the gestures do not indicate anything about the content of information present in other modes — such as speech — but rather serve as cues to the mental state (nervousness, determination) of the person making the gesture.

Beats and self-adjusters may also serve to help the speaker organize her thoughts; often our thoughts "run away" from us, and gestures can serve to focus us on what we are saying and keep our speech from wandering.[9]

---

9. This insight is also due to Dr. Bolt.

These five categories of gesture are most useful in an interpretive context; each class can be interpreted by different methods — applying a vector, or matching the hand configuration to pre-stored models of rote gestures. This interpretation of gestures in a multi-modal context is part of the work being done by my colleague Dave Koons. This thesis will not discuss interpretation in detail, but will refer to the classifications enumerated above. Koons' work is central to the system which was built to demonstrate our theses. The system is discussed in Chapter 4 "Implementation".

The different sorts of gesture in the AHIG taxonomy are summarized inTable 2; Efron's classification is repeated for comparison purposes.

**Table 2: Kinds of Gesture**

| AHIG Classification | Efron Classification | Identifying Characteristics |
|---|---|---|
| symbolic/modalizing | symbolic/emblematic | 'rote' gestures<br>1-1 mapping of gesture to meaning |
| pantomimic | kinetographic | use of object affordances to display interaction of hands and objects |
| iconic/object | iconographic | hand 'becomes' object to display action of scene |
| deictic/Lakoff | — none — | pointing at thing/area; space around body used |
| beat/Butterworth | baton | mark rhythm of speech |
| — none — | ideographic | portrays idea/metaphor |

## 2.4 Computer/System Research

As noted above, there is a significant amount of literature on pen-based and other hand-held-object gestures. Comparatively little work, though, has been done on open/empty handed gestures of the sort we are interested in. This section of the thesis treats only the empty-handed gesture work. This work can be divided into two different approaches — in one the hands are used as a form of three-dimensional mouse in that they provide a pointer/selector with depth and mobility in three-space. In the other approach, the hands are used to create a command language; the person makes gestures and they are matched (usually by a one-to-one mapping) to command verbs. This approach is often characterized by users making a gesture to fly around the virtual world, a technique which Myron Krueger disparages as *finger-flying* in his public talks.

In both cases, virtual environments (or VR) work has been the most prolific area for gestural interfaces to computer systems. Generally, VR systems use a computerized glove such as a VPL DataGlove (Zimmerman et al 1987) or Virtual Technologies Cyberglove for input.

### 2.4.1 Hand as 3-D Mouse

In cases where the hand is used as a mouse, the computerized glove worn by the user serves as an input source for a real-time animated computer-graphic model of the hand. The virtual hand moves around in response to the user moving his hand, and may intersect with objects or may project a selection ray from an extended finger. The selection-ray approach was used by Lewis et al (Lewis, Koved & Ling 1991) and their successors (Codella et al 1992) in the "Rubber Rocks" game.

In the game, users watch a computer monitor which has a representation of their (gloved) hand and that of an opponent. The hands move around a graphically-depicted cube in pursuit of a set of virtual objects — simple polygonal shapes with Newtonian physical properties. The objective is to "grab" an object by pointing your finger at it. The computer projects a ray from the extended finger; if the ray intersects the object, it is captured. It can then be "thrown" at the opponent's hand representation to explode. You win by destroying your opponent more times than he destroys you.

A similar approach using the hands as 3-D mice has been followed by Weimer and Ganapathy (Weimer & Ganapathy 1989 & 1992). They constructed a system using DataGloves which allowed users to generate three-dimensional curves for computer-integrated manufacturing applications. The user's extended index finger position was processed by a series of corner detection and decimation algorithms to produce a series of piecewise cubic bezier curves. In their earliest application, they treated the hand representation as a direct mouse equivalent — the user selected from control panels and from menus of b-splines by moving the hand analog until it intersected the desired choice.

Generally speaking, using the hands as 3-D mice is an extension of the direct-manipulation approach to interfaces; although we do not use this approach in our demonstration systems, our gesture recognition systems (including the one described in the next chapter) are compatible with using the hands in this fashion.

### 2.4.2 Gesture Command Languages

In this approach a hand configuration or specific motion is recognized and used as a direct command input.[10] Gestures form a command language used to give directions to applications. This approach fits especially well when using gestures from a sign language such as ASL, as was done by Murakami and Taguchi in their neural-network recognizer for Japanese (JSL) word gestures (Murakami & Taguchi 1991).

Their recognizer uses a set of neural networks to recognize 42 finger-alphabet gestures, each of which corresponds to a specific hand posture. They were able, with extensive training, to get the recognizer up to an accuracy of 92.9% for these static configurations. The user in their system is required to give a signal before and after the gesture so that the recognizer knows when to operate. They note this as a problem, but do not provide a solu-

10. In the "Rubber Rocks" game, only 'capture' and 'release' gestures are recognized. They are made by extending and then retracting the index finger.

tion. A similar problem is discussed below for AHIG's "Iconic" system, and this thesis proposes a solution.

Murakami and Taguchi then expanded the system to recognize ten JSL word gestures, all of which involved free hand movement. This worked much more poorly than the static recognition. Their neural nets were able to differentiate between two JSL gestures, but not to reliably identify an arbitrary gesture from a learned set. Murakami and Taguchi identify three general problems to be solved by gesture recognizers that are going to work with dynamic gesture information:

- How to process time-series data
- How to encode input data to improve the recognition rate
- How to detect delimiters between sign-language words

These issues appear in some form in all gesture-recognition systems. Even when recognizing natural gesticulation rather than sign-language gestures, we need solutions to the time problem and to the delimiter problem.

A typical example of the VR approach to gestures is provided by Väänänen and Böhm (Väänänen & Böhm 1993). Their system, called GIVEN (Gesture-driven Interactions in Virtual ENvironments), is a generalized virtual environment which serves as a testbed for their research group's work. Interaction with GIVEN is by means of a DataGlove. Their recognition software which is — like Murakami and Taguchi's — based on a neural network, captures any of roughly 10-20 hand configurations and matches them to system commands (Väänänen & Böhm 1993, p. 97):

> *"The program 'understands' different fist and finger positions (postures) received from the DataGlove as commands such as 'fly forward,' 'fly faster,' 'reverse,' 'grab,' 'release,' and 'go to the starting point.'*

New gesture commands can be generated interactively by the user; the neural net is put into a learning mode and the user makes the desired hand configuration which is then matched to a system command.

GIVEN allows what Väänänen and Böhm call "static" and "dynamic" gestures. A static gesture is just a hand "posture" (or configuration in the terminology of this thesis). A dynamic gesture is:

> *"...the movement of the fingers in addition to the position of the hand in a sequence of time steps." (ibid, p.101)*

Väänänen and Böhm use a fixed time window of five steps to determine if a dynamic gesture has occurred. The user must match finger and hand movements in each of the five time steps closely enough for the neural net to recognize the gesture. However, because their system memory is only five steps deep, the user must be careful not to make any movements after completing the dynamic gesture before the gesture is recognized.

## 2.5 AHIG Approach

Given the background of the preceding two sections, this section provides a critique of that work and shows how the research undertaken by the AHIG group differs from much of what has come before. This is followed by a specific critique of the "Iconic" system done in the AHIG group by Koons and Sparrell (Koons, Sparrell & Thorisson 1993) which is the primary predecessor to the present thesis.

The general problem with computer understanding of gestures can be seen as a problem of representation. In order to make gestures available to the computer they must be transformed into a programmatic representation. We explicitly reject the template-based approach which is currently popular because it fails to cover the wide variation in natural gesture both between people and within one person's gestures but between contexts. In addition, a template-based approach defeats the purpose of using the gesture mode which is, as noted above, to provide unique and confirmatory information. Templates provide exactly the same information available from a key press or mouse click.

It is important to realize that what we reject is not the *use* of templates per se. As will be discussed in Chapter 4 "Implementation", some matching must be done. It is possible that dynamically constructed and highly flexible templates might be useful for this kind of matching. Darrell and Pentland investigated construction of such templates by building a computer vision-based system (Darrell & Pentland 1993) which interactively learned templates for simple motion gestures such as "hello" and "good-bye."[11]

However, we reject the rigid one-to-one mapping of gesture to meaning in which you go directly from a detected pattern (a recognized gesture) to immediate assignment of command or function (an understood gesture).

The over-reliance on the rigid mapping approach seems to be rooted in a misunderstanding of the purpose and nature of gesture. For example, Väänänen and Böhm, in their discussion of gesture and its purpose say:

> *"Gestures are body movements which are used to convey some information from one person to another... In human-computer interaction, gestures... must be exact." (ibid, p. 94)*

This ignores the psychological research which shows that gesture is part of a complete utterance made by the speaker. We make gestures even when we cannot see the listener and we know the listener cannot see us.[12] To take gesture out of this communicative context is to miss their importance in the user interface; it leads people (erroneously) to think of gestures as necessarily being precise and rigorous. Of course, gestures are usually nei-

---

11. Unfortunately, the authors indicate that they have no plans at the time of this writing to follow up on the work reported.
12. Watch people on the phone sometime; they gesture just as if their listeners were present.

ther precise nor rigorous but if you think gestures require exacting precision, then templates might seem the only possible approach.

Not all computer researchers reject the implications of the psychological literature. For example, Weimer and Ganapathy are quite straightforward in noting that their work makes use only of symbolic gestures which are amenable to a one-to-one mapping of gesture to meaning. Similarly, Hauptmann (Hauptmann 1989; Rudnicky & Hauptmann 1992) performed a wizard-of-oz experiment[13] in which subjects were free to use voice and gesture as they chose in manipulating 3-D objects on a CRT screen. They found very similar results for this computer setup as the psychologists found for conversational environments (Rudnicky & Hauptmann 1992, p. 170):

> *"Subjects overwhelmingly... preferred the combined use of speech and gesture for the interface over each modality used alone."*

Rudnicky and Hauptmann also point out that there are no "expert" users of gestures, and argue for the use of gesture as a companion mode to speech and keyboard/mouse.

A similar argument for blended modalities is made by Hill et al (Hill 1992), who argue for the blending of multiple modes at the level of communication semantics, rather than waiting for information to be passed to the application. Their architectural advance is shown below in Figure 6. The diagram on the left is their view of how modes have been combined in the past; the diagram on the right is their suggested alternative and was implemented in their Human Interface Tool Suite (Hollan 1988).

This approach is very similar to that taken by the AHIG demonstration systems; the blending of semantic and pragmatic information from multiple input modes is done by the multi-modal interpreter created by David Koons (see "The Interpreter" on page 61). The primary importance of this blending style is the understanding that modal information cannot be treated separately. The GIVEN-style approach of requiring exactitude in gesture and of assigning meanings by a 1-1 mapping of gesture to meaning is compatible with the left-hand side. But the psychological literature strongly supports the newer (right-hand) interpretation, where the semantics of a given input from any mode can only be determined in a combined context which takes into account all the modes available. In addition, given the un-language-like nature of gesticulation, it seems highly doubtful that there even exists a differentiable "semantics" and "pragmatics" of gesture, as the old approach would require.

---

13. A wizard-of-oz experiment is one in which a human observes the user giving some input to the computer and provides the responses/reactions which the computer cannot do. The idea is to investigate peoples' interactions with more sophisticated computer systems than are generally implemented. The name comes from the movie and from the hope that the experimental subjects will not notice the "man behind the curtain."

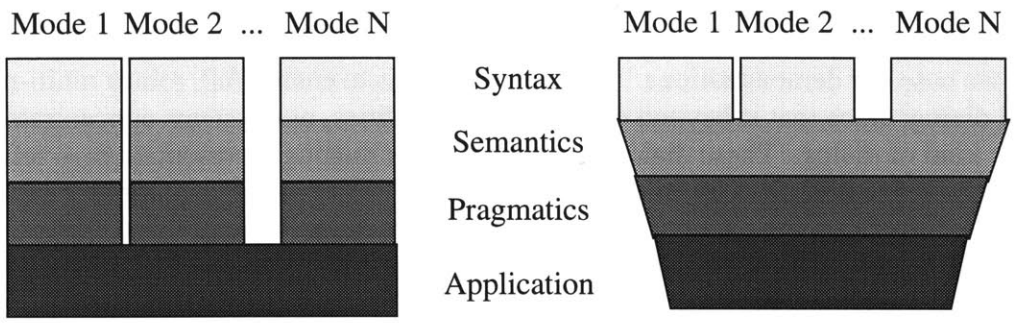


Figure 6. Old and New Ways of Blending Modes (from Hill 1992)

### 2.5.1 AHIG Work

Given that we reject a template (or whole-gesture) model, the problem can reformulated as identifying which parts of gestures to capture. The answer to this latter question is similar to the answer found in visual systems. We wish to capture and represent critical *features* of gestures. This approach has been common to AHIG applications for some time now, and represents one of the key differences between our approach and that of others doing research in related areas. The feature-based approach says that matching gestures at the whole-gesture level is programmatically very difficult for two reasons:

1. the number of possible gestures people make is very large. Even with the restrictions described in Chapter 2 "Background and Scope" on the types of gestures being studied, this number is still too large to allow a set of templates to be built.

2. the variability in gestures is also very large. There is variability not only between people on the same topic, but also within one person's gestures. This indicates that rules for matching gestures to templates would be complex, difficult, and very error-prone.

These assumptions were reinforced by the findings of my experiment on descriptive gesture (see "Experiment" on page 39).

We have implemented a number of feature-based gesture detectors over the years, deriving initially from the ideas found in the "Put That There" system. Initially research focussed on purely deictic gestural input, but in combination with other modes of input (eye-gaze and speech in particular). Most recently, the "Iconic" system built by Koons and Sparrell in conjunction with Sparrell's thesis attempted to tackle the problems of iconic/object and pantomimic gestures.

The "Iconic" system demonstrated several important points, but was deficient in a number of areas, which this thesis attempts to correct. Section 2.5.3 "The "Iconic" System" lays out some of the deficiencies and strengths on which the current thesis is built.

### 2.5.2 Directive Interfaces

One of the most important distinctive aspect of our research is the style of interface on which we base our demonstrations. Our eventual aim is to enable full, robust multi-modal natural dialog; since that is beyond our present capabilities, our systems concentrate on a specific kind of dialog. These dialogs involve the user making a presentation — telling a story in the sense described by Brenda Laurel (Laurel 1991) — to the computer and having the computer carry out actions based on the presentation. We call this a "directive" interface, since our current simplistic stories consist of little more than the person giving directions to the computer.

In a directive interface, the user makes a multi-modal presentation describing the task to be done, and the computer executes the task. Directive-style interfaces are also seen in the computer-human interaction work on learning interfaces (see for example Bos 1992). Learning interfaces generally do not require the user to make a specific presentation; rather, they observe the user's repetitive actions and offer to do the action sequence for the user. In addition, the ideas of computer agents which do the users' bidding also depend on humans being able to give direction in describing the tasks they delegate to agents to perform. Future implementations in the AHIG group will include an agent or "embodiment" of the system which will give the user a more informative dialog partner.

Since we do not yet understand how to have a full natural dialog with the computer, Laurel's storytelling approach seems a natural way to progress. Story telling is a form of dialog, albeit a restricted one in which the story teller does most of the presenting and the audience reacts. As we elaborate our stories in future AHIG demonstration systems, we expect the computer to be able to respond in more complex and interesting ways. The human will still guide the dialog, as happens in human storyteller/human audience cases, but the computer audience will interact in more sophisticated ways.

### 2.5.3 The "Iconic" System

The "Iconic" system (Sparrell 1993), illustrated in the block architecture diagram in Figure 7, took its input from a pair of VPL DataGloves and accumulated the data. When a completed gesture was detected, it would output the record describing that gesture.

A 'completed gesture' was defined to be one which had all three of the phases of a classical gesture. Users began with their hands at their sides, raised them up, made the gesture, then returned their hands to the resting position. The acts of raising and lowering the hands were detected by the gesture recognizer and used as key markers for the start and end of the gesture. In addition, the stroke phase of the gesture was marked by the gesturer stopping his motion, first at the beginning of the stroke (after the preparation) and then again at the end of the stroke. This pause-based scheme derives from research by Kendon (Kendon 1980), who asserted that gestures consisted of five phases: *preparation*, *pre-stroke hold*, *stroke*, *post-stroke hold*, and *retraction*.

The "Iconic" gesture recognizer took the data records produced by the gloves and by Polhemus six-degree-of-freedom position sensors (not shown in the diagram) and merged

these records to give significant intervals —the stop/start segmentation — detecting these intervals by observing changes and steady states in the data values.

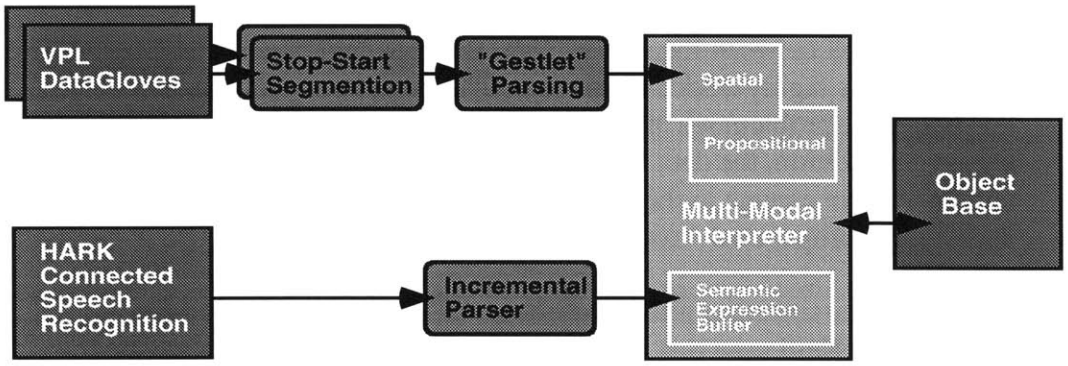


Figure 7. The "Iconic" System

Sparrell then attempted to categorize these intervals in terms of primitives or *primes*, derived from work on ASL (see Sparrell 1993, p. 20). The intervals were further assigned to one of a set of basic shapes, which were assumed to be primitive components of iconic or pantomimic gestures. The combined information was then merged with information about the place, movement, and orientation of the hands to produce what Sparrell called a *gestlet* — a sub-gesture.

As with the current thesis (see Section 4.2 "Implementation Architecture", specifically Figure 18 on page 57 for the comparable diagram), the "Iconic" system included a number of other components. The multi-modal interpreter accepted input from the gesture code and from an incremental speech parser. The information from these two modes was separated into spatio-temporal (concerning the times and locations) and propositional (concerning the relationships) representations inside the interpreter. The interpreter — which was built by Koons — then used these representations to control actions in an object base, which in turn showed its actions on a large-screen display.

The demonstration domain was a simulated room layout application which allowed designers to create and move furniture and related objects in a computer-graphic room. Architects could, for example, create a table and place objects on the table. Objects could be moved, and the room as a whole could be manipulated (tilted, or viewed from different locations).

*Critique of the "Iconic" Approach to Gesture*

The "Iconic" gesture recognition system suffered from some problems when implemented as a prototype system, which I enumerate here. In the next section, I will outline my proposed solutions to these problems and advances on this state of the art system.

1. The use of Kendon's divisions is questionable. Even Kendon asserts that only the stroke phase is necessary. However, the system as built required the user to insert artificial pauses into gestures, in large part because the

data devices could not produce reliable values fast enough. This meant that people had to artificially extend their holds which, in unrestricted gesturing, are often very short or omitted entirely.

2. The direct link between the gesture code and the input devices resulted in work that was highly device-dependent. Although not a problem in theory, this became a problem when new devices were acquired, as explained below. In addition, the data format produced by the devices was used almost without change in the prototype code. This meant that device-dependence propagated throughout the system.

3. The use of classification primes based on ASL is highly questionable. Although McNeill used this approach, it seems like this post-facto categorization is not generalizable for most gestural input. The configuration primes came to resemble templates and began to suffer from the weaknesses of templates noted above.

4. The "Iconic" approach is not capable of scaling by increasing integration of information. This advance — which is implemented in the current thesis — will be discussed in detail in later sections, but in general the idea is that a gestural analyzer should be able to produce increasingly-elaborated descriptions of the gesture by combining information as more becomes known (either over time or by reasoning over higher-level representations).

5. The use of classification primes revealed a subtler problem in the "Iconic" approach: there was no clean separation between the work being done by the gesture code and that being done by the multi-modal interpreter which took in the gestural input. The primes and features detected in gestures were precisely those that were needed by the interpreter to do a one-to-one matching with objects in the demonstration application.

   A related problem was that the "Iconic" gesture code assumed that the user was going to be making some form of iconic/object or pantomimic gesture. It attempted to use features of the gesture to determine what the user was doing. This violates the principle of separation of concerns and makes it impossible for the interpreter to properly apply multi-modal contextual information to the gestural input.

It is worth noting when entering into a critique of this work that Sparrell never intended to provide a complete solution to the problem of gestural input. Rather, he asserted that he was building an initial prototype on which more general solutions might be based. As is explained in detail in Chapter 3 "Solution", the present thesis is predicated on providing such a general solution; however, that work could not have proceeded without the invention and discovery embodied in the "Iconic" system.

# Chapter 3 Solution

This thesis emerged as a set of solutions to the problems seen in the "Iconic" implementation. The primary desire was to come up with a design that would allow gestures to be made more naturally, where that primarily meant "continuously." While pauses do occur in natural gesture and can be taken advantage of, users should not be required to insert holds into their gestures artificially, nor should they be required to return their hands to a pre-specified "rest" position between each gesture.

Adherence to these theoretical aspects of gestural models had come from our work being too long divorced from real gesturers. Each of us in AHIG has remarked on the feelings of having our gestures become stilted and rote after giving the same demonstration over and over. We no longer have a sense of how people would naturally make the gestures.[14]

To help overcome this, and to give us some raw input on how people make gestures I performed an observational experiment and analysis. This experiment and its results are described in detail in the next sections.

## 3.1 Experiment

The experiment involved subjects being videotaped giving descriptions of short scenes that they had watched. Two research questions needed to be answered:

- What sorts of gestures do people make in giving descriptions?
- How do we characterize these gestures so that a computer could understand them?

The first question addresses the issue of familiarity. I expected, going into the experiment, that I could tell when subjects would gesture and roughly what they would gesture about. The second question addresses the problem of identifying appropriate features to encode in a gesture-recognition system.

### 3.1.1 Experiment Setup

Subjects were seated in a room with a large-screen TV, videocamera, and tape player. They read a short description of the experiment and filled out a questionnaire and video release form. The chair in which the subjects were seated was positioned away from tables and other objects in the room. This left subjects' hands free for gesturing, if they so chose.

Subjects viewed eleven scenes from a movie — either THE TERMINATOR or CASABLANCA. Each scene was 20-45 seconds in length. After viewing all scenes, the subject

14. We have, however, all become highly sensitive to the gestures we see. Recently I found myself startled by seeing, in the videotape of the movie DEMOLITION MAN, that in one scene Sylvester Stallone repeatedly shook his head side to side when his dialog was saying affirmative things.

was asked to describe a specific scene (referred to by number). In order to facilitate recall and eliminate order effects, subjects re-watched the target clip immediately before giving their description. They were videotaped giving this description. This re-watch, describe, tape procedure was done twice for each subject.

Subjects were told that their videotaped descriptions would be viewed later by someone who had not seen the clips before and who would attempt to determine which clip the subject was describing based solely on the taped description. The instructions to the subjects did not mention gestures at all, as pre-experimental testing showed that mentioning gestures made the subjects "too self-conscious" (in the words of one volunteer) about their hands.

Subjects were told to be as complete and accurate as possible in their descriptions. The instructions (which are reproduced in Appendix A — Experiment Information) emphasized that there was no right answer and that what was important was to be as complete and accurate as possible. During the description, experimenters gave para-linguistic feedback (nods) but did not attempt to correct errors or elicit clarification from the subjects during the description. When the subjects indicated they were finished, they were asked "Anything else?" in order to give them a chance to reflect on their responses. Most subjects said "No;" a few added one or two sentences to their descriptions.

Subjects were paid US$5 for their participation, which took less than 30 minutes in all cases. Thirty-five subjects participated, 18 viewing THE TERMINATOR clips and 17 viewing CASABLANCA clips.

The scenes shown to subjects were selected to focus on four major characters from each film, and action sequences were duplicated wherever possible. For example, in THE TERMINATOR clips, subjects viewed four "chase" scenes; in the CASABLANCA clips, subjects saw four "drinking" scenes. Table 6 on page 87 contains a complete list of the clips used. The overlap of characters and actions was deliberately maximized to encourage subjects to use other differentiating information. That is, by having Rick appear in the majority of the CASABLANCA scenes, subjects could not simply say "Oh, this is the one scene with Rick in it."

This overlap worked surprisingly well. Thirty-one subjects gave what would be considered "complete" descriptions in that they narrated all the action of the clip. Four subjects found specific information which sufficed to distinguish their target clips and gave only that information. These four subjects were among the seven who made no gestures during their descriptions.

### 3.1.2 Experiment Output

The videotaped descriptions were transcribed for text and then annotated with gestural information using a style similar to McNeill's.[15] An example of a transcribed scene is

15. The appendix to *Hand and Mind* gives a very detailed description of the annotation procedure; a short description is on page 12.

shown in Figure 8 on page 42. In this transcript, the subject's text is in plain font and the annotations in italics. Words which coincided with the stroke phase of a gesture are in brackets. The hash mark (#) is used to identify a place where a gesture stroke occurred between words. Ellipses (...) are used to mark places of verbal hesitation and paralinguistic utterances. Numbers at the beginning of lines are for reference purposes.

Gestures are first identified as to category, using McNeill's distinction between character-viewpoint (C-VPT) and object-viewpoint (O-VPT) iconic gestures.[16] Then the gesture is described, using salient featural components. In cases of iconics, the intended object of representation is identified; in the case of deictics, the thing pointed to is identified.

### 3.2 Experiment Results

Several initial hypotheses were proposed in an attempt to generalize about the gestural information. Unfortunately, none of these hypotheses held up. However, performing the transcription and analysis led to several important insights.

There were 'many-gesture' subjects and 'few-gesture' subjects. A many-gesture subject produced 50-100% more gestures (usually 6-12 gestures, infrequently more) in describing a given scene than a few-gesture subject (usually 2-4 gestures). Several hypotheses might account for this difference.

I hypothesized that clips from THE TERMINATOR would produce more gestures than clips from CASABLANCA because the former movie is more "action-oriented" than the latter. In addition, the latter is filmed in a more modern style with many more camera cuts and shorter shots. This frequent change in viewpoint was expected to contribute to the gestural frequency. However, this turned out not to be the case.

Subjects' gender and native language were also not reliable predictors of gesture frequency. The effect of native language can not be precisely qualified, as all subjects spoke in relatively standard English. Studies have shown that speakers operating in a non-native language use more gestures than with their primary language (Grand et al 1977; Marcos 1979). It might be supposed that the use of English might have converted some few-gesture subjects into many-gesture subjects, but within this general categorization, the non-native speakers did not significantly differ from the native speakers.

This number-of-gestures property might also be hypothesized to be a function of the particular scenes viewed, with scenes having many moving objects producing more gestures. That turned out not to be the case; instead the raw number of gestures seen seems purely to be a function of the subject rather than the scene.

---

16. A character-viewpoint iconic is one which is made as though the speaker were the character performing the action. These correspond to the pantomimic gestures of Section 2.3.6 "AHIG Categorization". An object-viewpoint iconic is one which is made as though the speaker's hands/body was the object being talked about. These correspond to iconic gestures in our taxonomy.

Subject 4:
1) It starts with [one of the characters]
*Deictic: hand, palm up, finger points to Rees location in gesture space.*

2) [throwing off his jacket] and pulling out a shotgun.
*C-VPT iconic: hand, fingers curled, moves from center of body to over right shoulder (representation of pulling open trenchcoat).*

3) And [they're...] it [seems like] a nightclub because there are many women
*O-VPT iconic: hand, palm up, waves back & forth (representation of bodies dancing). Gesture repeated.*

4) dancing behind him. It then cuts to the [Terminator guy]
*Deictic: hand, palm up, fingers point to Terminator location in gesture space.*

5) [cocking] his gun and starting to bring it up.
*C-VPT iconic: hand moves back & up (representation of moving slide back to cock hammer).*

6) And then [... the other character] — I don't remember his name —
*Deictic: hand, palm up, finger points to Rees location in gesture space.*

7) [starts shoving people] out of his way.
*C-VPT iconic: hand, palm up, waves back & forth (representation of shoving people]*

8) It then cuts [to] the Terminator guy
*Deictic: points to Terminator location in gesture space.*

9) — [he had] a laser sight on his gun...
*C-VPT iconic: looks down at palm of hand as if at object in hand [representation of pistol in hand]*

10) It [cuts to] [#] what's her name sort of staring at him
*O-VPT iconic, then beats: two hands held up, palms facing [representation of Sarah's head from close-up shot]; beats during attempt to remember name.*

11) with the [little red dot] on her head from the sight. It cuts again, I think, to
*C-VPT iconic: index finger touches center of forehead [representation of laser sight circle]*

12) the Terminator and he's about to fire, then [Rees] fires
*Deictic: points to Rees location.*

13) and hits [him] with the shotgun. The [Terminator guy] stumbles back and Rees
*Deictic: two pointings to Terminator location.*

14) fires [three or four more times] with
*O-VPT iconic: hand, palm up, sweeps away from body [representation of Terminator falling (see below)]*

15) the Terminator guy [falling back.]
*O-VPT iconic: same gesture.*

16) Then it shows people [stampeding] out of the nightclub.
*O-VPT iconic: hand curled into cylinder sweeps across (l-to-r).*

17) They show [one with] [the woman] on the ground covering her head with
*???, deictic: two hands, one curled resting in palm of other held out; then points to lower center (where Sarah was lying on ground).*

18) people [running out] of the club around her.
*O-VPT iconic: hand curled into cylinder sweeps across (l-to-r) [representation of people running].*

Figure 8. Sample Transcription

More importantly, it quickly became clear that in general we could not predict *what* users would gesture about. An initial hypothesis had been that subjects' gestures would concentrate on action descriptions, showing the placements and movements of objects in the scenes. This is consonant with our approach of building descriptive interfaces. However,

this hypothesis turned out to be completely false and indicates that descriptive interfaces may not extend to dialog situations such as used in this experiment. Notably, one subject used only one gesture in describing an entire chase scene and that gesture served only to clarify the placement of lights on top of one of the vehicles in the chase.

However, we noted a number of gestures that were repeated by several subjects in describing the same or similar scenes. In addition, we noted that even with the sound turned off for the videotape playback, we could usually tell when the subjects made a significant gesture. Knowing when a significant gesture was occurring suggested that there were key elements in common between subjects. The subjects' gestures differed even when they described the same events within the same scenes, so clearly the commonality was not at a full-gesture analysis level. This supports our general approach of using sub-gestural features and indicated that Kendon's definition of what makes something a gesture (see "Background and Scope" on page 18) is essentially correct.

### 3.2.1 Categorization Problems

At the same time, the transcription of the videotapes shown above revealed a number of problems with gestural classification systems such as those used by McNeill (McNeill 1992). In particular, classification of gestures required that I use knowledge about the scene being described and about subjects' intentions. For example, in Line 10 of the transcript, the subject simply held his hands up. Only by knowing what is being described (a close-up shot of the character) can we categorize the gesture.

Information about the thing(s) being described would not be known by a computer system which was observing the user's hands and trying to understand them. Specifically, that information is not contained in the gesture. It seems that this information is probably not available to humans at the time gestures are made. It is only during post-facto analysis such as this experiment transcription that we have access to the content of the description as well as the object being described and can make inferences about the intention of gesture based on that combined knowledge.

This sort of "error" is present in much of the previous work on gestures. I concentrate on McNeill & Levy because their work is the only one to emphasize a feature-based approach to gesture. In *Speech, Place, and Action*, p. 280, they give examples of iconics which: "...reflect the speaker's concrete representational models of reality." Their subject has watched a Warner Brothers' cartoon and is narrating the events of the cartoon to another subject who has not seen the cartoon. The narrating subject says:

> *"She chases him out again." (ibid)*

They annotate this text with the gestural description:

> *"LH in fist moves down and to the right." (ibid)*

This is a feature-based description and records the kind of information which will be available to gestural input software. However, in classifying the gesture, they say:

> *"Here the gesture shows the hand of the protagonist grasping an umbrella... and simultaneously the pursuit of the target and/or the swinging of the umbrella." (ibid)*

This information is not available anywhere in the verbal narrative and while McNeill & Levy may be correct that this is the concrete mental representation that the speaker is using, it seems clear that this representation does not communicate itself to the listener/ observer.

It is also worth noting that even with all the information present in hindsight analysis, it is sometimes not possible to classify gestures. In the analysis following line 17 of the sample transcript, one of the subject's gestures was not classifiable. This seems to support the contention above that model information is applied post-facto to help human gesture understanding — sometimes people make gestures that we simply do not understand. A particularly clear case of this came from two of the subjects who seemed quite reluctant to take their hands away from their knees while giving their descriptions. Their hands moved, but only in a very restricted area. As a result their gestures were highly constricted and inexpressive. It was the gestural equivalent of verbal mumbling. During transcription it was impossible to tell what their gestures meant; looking at the videotapes, it was clear that their gestures did not add much to what they were saying.

The experiment transcript also shows some interesting results in the temporal relationship of speech and coverbal gesture. While many references, both deictic and iconic, happen at the same time as the spoken target (e.g. lines 11 and 12), many do not. In some cases, as in lines 14 and 15, the gesture comes several words before the thing to which it refers.

The tapes show that subjects who spoke faster most often had gestures which preceded the referent words. This does lend support to McNeill's theory that expressions are formed in a central locus in the brain/mind and then expressed along different paths for speech and gesture. It seems that the rapid speakers have trouble getting their words to keep up with their thoughts, while their (often very rapid) gestures can easily keep pace.

McNeill & Levy also make extensive use of the positioning of the hands relative to the body when a gesture is made. In later work (McNeill 1992), an attempt is made to subdivide a "gesture space" in which the meaning of gestures is associated in some way with their body-relative placement. No support for this hypothesis appeared with the experimental subjects; gestures meant the same thing no matter where they were performed in relation to the speaker's body. Therefore, all the features selected (discussed below) are treated in abstraction from the body. Gestures that involve *using* the body (such as shrugs or chest-tapping) are not part of this thesis.

The final leg on which a categorical approach to gesture might stand is the commonality of categories across speech instances. As noted above, my subjects yielded no statistically-reliably predictors of what actions, event, objects, or people would cause gestures to occur. McNeill & Levy attempted to correlate features in gestures associated with verbs.

However, their best correlations occur only 50-60% of the time.[17] Thus, it seems like a categorizing approach is not going to be widely successful.

### 3.3 Features Selected — Thesis Basis

Key to the success of this thesis is the identification and elaboration of a set of features to be used in characterizing movement; this is a basic problem for any sort of representation system. This section describes the set of features selected for this thesis and explains what kind of information can be ascertained about each kind of feature. The features selected are largely motivated by the experimental observations described in the preceding section.

As noted in Section 2.5.3 "The "Iconic" System", Sparrell used a trio of place, movement, and orientation in combination with categorized primitive shapes. McNeill & Levy appear to have a different approach — they report 44 features just for gestures used in conjunction with verbs. However, an examination of their list[18] (reproduced below in Figure 9) shows that many of their features are duplications ("at start" versus "at end"), and others are multiple ways of looking at a single feature ("orientation" versus "change in orientation").

However, there are key elements both in this list and in Sparrell's gesture recognizer. This thesis draws from both these precedents and concentrates on three kinds of features: *configuration-related*, *orientation-related*, and *motion-related.*

For each kind of feature, two kinds of information can be collected: spatio-temporal and categorical. Spatio-temporal information relates to the precise location and timing of gestural features. It describes precisely where and when something happened. Categorical information classifies information into one of a set of values. All the features listed in Figure 9 are (apparently) categorical features. However, as shown in the "Iconic" system, spatio-temporal information is one of the most important things available from the gestural mode.

Given that we do not want to have an extensive list of features, there is still an important question as to which features are essential to capture in gesture recognition. Section 2.5 "AHIG Approach", noted that our use of features in gesture-recognition systems is based on intuitive understanding of how visual systems decompose scenes. However, we are faced with a problem: in the human visual system, evolution appears to have "programmed" for us a set of features (edge detectors, motion detectors, and the like can be observed experimentally) that we use. There are many features that might be captured out of a visual scene but because of this programming, the set that we detect is fixed. Part of

---

17. Fifty percent is almost precisely the definition of "some do, some don't." In addition, McNeill and Levy reported on only six subjects; with a larger subject base even greater variation appeared.
18. The list is abbreviated by the authors in the original article. They state that "in the complete list 44 gesture features appear" but they do not show the complete list. Nevertheless, I think my point is still valid.

the core of this thesis is the solution to the question of which features should be captured. This question is discussed again in Chapter 5 "Analysis & Conclusion".

1. Phase (preparatory, iconic, or retraction)
2. Spatial location (centre of periphery)

AT START –

3. Hand configuration (fingers curled, fingers extended, fist,...)
4. Tension (tense, relaxed)
5. Orientation of palm (palm up, palm down, to left, to right,...)
6. Position relative to other hand (tips of fingers touching, palms together,...)

AT END –

7. Change in configuration (fingers extended, fingers spread,...)
8. Change in orientation of palm (palm rotates clockwise, counterclockwise,...)
9. Absolute direction of movement (up down, to left, to right,...)
10. Relative movement (toward other hand, toward self,...)
11. Path of movement (straight line, arc,...)
12. Quality of movement
    (a) Rate (fast, slow)
    (b) Evenness (continuous, jerky)
    (c) Cyclicity (reduplicated)
    (d) Length (elongated, abbreviated)
    (e) End marking (+/- end marked)

Figure 9. McNeill & Levy's Gesture Features

The features chosen for this thesis were selected as the most visually salient features seen on the experiment tapes, within the three categories and two types noted above. The features are summarized in Table 3.

**Table 3: Features Used**

| | Position | Orientation | Configuration |
|---|---|---|---|
| Spatial | start & end Positions | start & end palm Normal and Approach vectors | — none — |
| Categorical | Coordinated (y/n)<br>Move type (line, arc) | Facing direction (in, down, out, up, right, left)<br>Coordinated (y/n) | Palm curve (flat, curled, cupped, closed, fist, pinched)<br>Extended fingers<br>Thumb position (out, beside, over)<br>Finger spread (together, spread, apart) |

Positions and directions are expressed as triples (X, Y, Z) in global coordinates. Movement of the hands through space is recorded as changes in position; start and end values are recorded for features which change in this way. Directions are labeled based on the system of approach, orientation, and normal vectors commonly used in robotics.

*Approach* is the direction the fingers extend in, *normal* is a vector perpendicular to the palm, and *orientation* is the axis that runs laterally through the side of the hand. This is illustrated in Figure 10, using two pictures; in each view the vector which would project directly out of the paper/screen is shown as a dot.

### 3.3.1 Position, Orientation[19], Configuration

As enumerated in Table 3 above, the features used in this thesis fall into three major categories. Information is collected to give both instantaneous (static) information, and cumulative (motion/change) information. This process of collection is similar to what Sparrell did in "Iconic" (see Sparrell 1993, pp. 42-44).

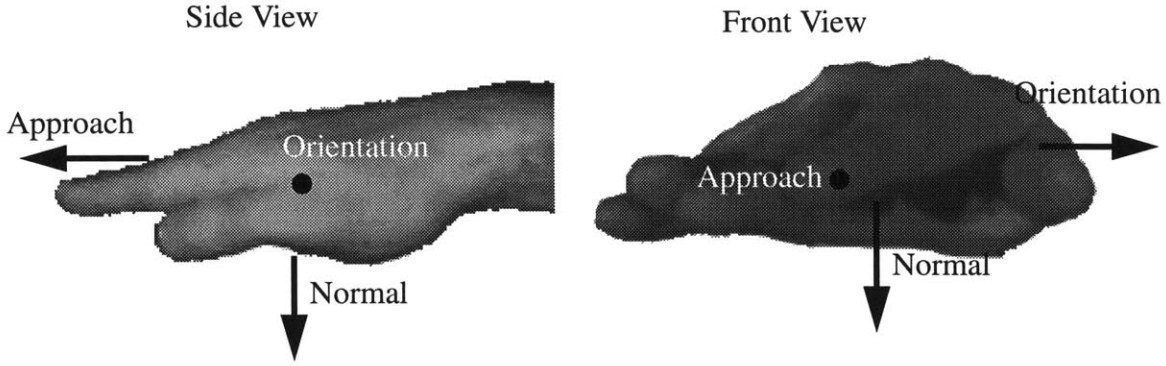


Figure 10. Vector Labeling

The position of the hands is taken to be a point in the center of the palm; motion of the hands is three-dimensional change in this point. In order to handle complex motions, the movement of the hand is broken into a series of segments, each with a start and end point (obviously the starting point of one segment is the ending point of the preceding segment). Segments are computed as needed for each particular motion; the number of segments is determined dynamically, based on the complexity of the motion. This is an improvement of the present thesis over previous work.

Another improvement is the consideration of the change(s) in both hands simultaneously. Previous work treated the left and right hands in isolation. However, the experiment tapes showed that the hands frequently made motions which clearly described a single item or scene. The key observation used for determining when the simultaneously-moving hands could be seen to be part of a single two-handed gesture was when they were making a coordinated presentation.

"Coordinated" here means literal spatial coordination. There are two general forms that this coordination can take:

- Mirrored or reflected motion. That is, when translated into the context of one quarter-hemisphere, do the motions appear to be the same.

19. Note that orientation in reference to hand features combines changes in normal, approach and orientation axes

- Aligned motion. That is, one hand provides an axis or point around which the second hand moves. Note that in aligned motion, it is frequently the case that one hand does not move.

Figure 11. "Traveling" penalty motion

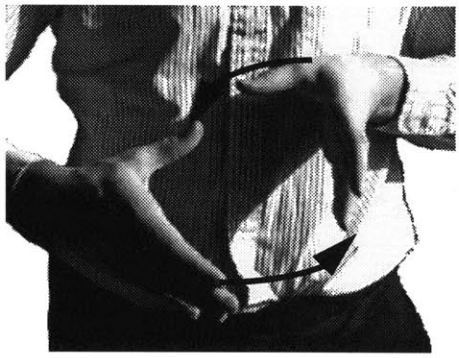

Figure 12. Turning the teapot motion

Examples of mirrored motion are shown in Figure 11 and Figure 12 above. In the first case, the hands are orbiting each other, but their motions reflect each other — one moves down and forward as the other is moving up and backward. In the second case, they are moving around a common center; again, considering only the directions of movement shows the similarity of motion and their coordination.

Examples of aligned motion are shown in Figure 13 and Figure 14 below. In the first case, the motion of the right hand is centered around the left hand, which provides a point of reference. In the second case, the moving right hand and forearm pivot around a point which is on the axis determined by the second hand.

Orientation changes are handled in much the same way as position changes. As noted above, orientations can be described with two vectors; changes can be recorded as a pair of start vectors and a pair of end vectors.

Figure 13. Earth orbiting motion

Figure 14. Tree-falling motion

In theory, it might be necessary to record changes in orientation via a series of steps, similar to what is necessary for capturing motion. However, this turns out not to be necessary in practice. Observations indicate that complex changes in orientation are invariably accompanied by movement of the hand; this seems to be dictated by our physiology, which makes rotations around the three major axes (approach, orientation, and normal) of the palm difficult to achieve without accompanying motion of the hand.

Interestingly, coordination of orientation changes can be determined in just the same way as with position changes. Since there is no movement, per se, of the hands in a pure orientation change there is only one case possible: the reflected case from above. If the hands change orientation in the same reflected manner, then they are making a coordinated presentation about a single object.

Configuration information is not kept in spatial form, only in categorical. The reason for this is observation from the experiment and from experience that precise spatial features in configuration are not semantically meaningful. What is important in recognizing hand shapes is the general information about approximately how the hand is shaped. For example, we do not care about the precise angle of elevation of the thumb in our familiar "thumbs up" gesture (see Figure 2 on page 27) — what we care about is that the thumb is extended and that it is out away from the rest of the hand.

The particular configuration features chosen for this thesis give an exhaustive list of interesting things about the hand's shape. Here 'interesting' is taken from the point of view of someone trying to describe the general status of the hand's shape by enumerating orthogonal dimensions. That is, a feature such as McNeill & Levy's "tension" could be added, but tension can only be expressed by changing something else such as the curvature of the palm.

It is also possible that additional categorizations could be added, specifying particular configurations — as McNeill & Levy (1982) did — or comparing the hand configuration to specific shapes — as Sparrell (1993) did. As discussed below, this is an error. A better approach is to reduce the hand configuration to its elemental components.

The features of interest are the curvature of the palm/fingers, the set of extended fingers, the placement of the thumb, and the spread of the fingers. Any given hand configuration can be described in terms of the complete set of these features. Figure 15 shows the hand in a common configuration used to show a gun.

This would be described as:

> *"hand curled closed, index finger extended, fingers together, thumb out from the hand."*

Figure 15. "Gun" hand configuration

The gun-hand configuration is a particularly apt example because it is a shape that people make several different ways. The approach of using features to abstract the description works for all the different ways because we can match on key features — in this case, the extension of the index finger and the orientation of the hand. The variable position of the thumb (some people curl it along with the fingers) is less important.

The hand curvature categories (enumerated below) correspond to the six hand curvatures most frequently seen in subjects' hands while they were making descriptions. Readers should understand that the examples given below of specific uses for each particular category do not embody any prejudgment about what hand configuration features "mean." Rather, they are illustrations intended to allow the reader to more easily imagine the configuration feature value being described:

- 'flat' is absolutely no curvature of either the fingers or palm. It seems to occur in situations where the speaker is representing a flat surface or (with palm down) an object such as a car which is basically rectilinear.
- 'curled' is a slight curvature of the palm, sometimes used in emphatic gestures when the speaker is not worried about keeping her hand perfectly flat.
- 'cupped' is a significant curvature of the palm and fingers. As the name suggests it is commonly seen when the user is miming holding or gripping things.
- 'closed' is a further curvature of the fingers, usually to the point where they touch the palm. However, the palm is always less curved than in the fist configuration and sometimes less than in the cupped configuration. As noted in the example above, closed configurations occur in cases where the hand is representing a complex shape; this configuration also corresponds to holding small objects.
- 'fist' is a closed curvature of the fingers, and a significant curvature of the palm. This is, as the name suggests, what happens when the hand is tightened into a fist. It corresponds often to object gestures where the hand is representing a spherical or tubular object.
- 'pinched' is a strong bend in the fingers, but with almost no palm curvature. This configuration often corresponds to holding small manipulator objects such as pencils or chopsticks.

The thumb-position categories are:

- 'over' corresponding to the thumb being in front of the palm.
- 'beside' corresponding to the thumb being next to or close to the palm.
- 'out' corresponding to the thumb being away from the palm.

Finger spread is categorized as:

- 'together' corresponding to the fingers being next to each other.

- 'spread' corresponding to the fingers being slightly apart — relaxed hands usually fall into this category.
- 'apart' corresponding to situations where the user has stretched his hand to move the fingers apart (a form of hand tension). This is usually seen when the user is miming holding a large object, or making a claw/rake shape.

### 3.4 Analogy with Speech Recognizer

Given the set of features described in the preceding section, there is a need to specify as concisely as possible the theoretically-based reasons for including certain information and excluding other information, specifically certain kinds of features used in the work done by McNeill & Levy and by Sparrell. At the same time the author underwent the common experience of being required to describe this work to people of widely varying degrees of interest and education. It quickly became necessary to develop a simple way of explaining the major thesis ideas in an easily-graspable form.

The key insight in this process came from reading Schmandt's (Schmandt 1994) description of speech technology. In his book, he gives a complete overview of the uses of speech in computer systems. One of the themes emphasized in the book is the idea that speech can be treated as a data type — like text or graphics — to be used in different ways in different applications.

Taking that view of modal input, it became clear that the work described in this thesis is similar in many ways to what a speech recognizer does. Abstractly viewed, a speech recognizer takes the movements of the air produced by speech and transforms them into a computational representation. The representation is governed by some rules embedded in the software; usually the rules specify a grammar similar to English (or other native language) syntax. Similarly, the purpose of gesture recognition software is to take the movements of the hands produced while gesturing and transform them into a computational representation. In further analogy, much of the improvement this thesis makes over the "Iconic" system is the same improvement that was made in going from discrete-speech recognizers to continuous-speech recognizers.

Schmandt points out that speech applications are actually much broader than generally recognized. He argues for an understanding of speech as a *data type*, in the way that text or graphics are. Data type seems to be just another way of saying 'mode.' That is, it should be possible to conceive of gesture as a data type which could be used in a variety of applications as appropriate to the needs of the particular application.

A speech recognizer is stupid in a particularly interesting way: given a recognizer with the rules of English syntax, it is perfectly possible to get the recognizer to correctly hear the sentence:

> *"Green ideas sleep furiously."*[20]

This sentence is syntactically plausible, but we immediately identify it as "wrong," in the sense that it is not semantically meaningful. However, that implausibility is a context-dependent judgement. In normal everyday conversation between people it is likely a correct judgement. However, in an application context, it might be the passphrase that I use to unlock my secret key. If I was using privacy-enhanced email[21] (PEM), it would be perfectly sensible for me to speak this phrase to the PEM application.

The point is that the speech recognizer does not make any judgement about whether or not the sentence makes sense. As long as the sentence is correct according to the rules built into the recognizer, it is up to the PEM (or other) application to determine if the phrase is meaningful. Similarly, what makes something a worthwhile gestural feature is just that it allows the gesture recognizer to know about gestures without knowing about the eventual end use of the gestures.

This leads to the final conceptual improvement of this thesis over past work: this thesis specifies a detailed method for decomposing the process of gesture recognition into two stages, with a clean division between stages:

- *analysis*, where data is captured, key features are detected and extracted from the data, and the abstracted features are combined into a higher-level representation;
- *interpretation*, where meaning is assigned to the representation based on multiple modes of input and the current dialog and application context.

The original idea of dividing gesture recognition into these two stages is common to AHIG work in general, dating from Bolt's initial conception (Bolt 1984). However, no concrete method for enabling this portion of the gesture recognition process has successfully been enumerated before this work.

The problems with the theoretical work described above and with Sparrell's approach in "Iconic" (both discussed above) can be seen to be related to a mixing of analysis information with interpretation information. Similarly, a fundamental flaw of template-based models is that they combine these two stages, doing both the analysis and the interpretation at the same time. A properly-implemented feature-based approach avoids this problem and allows the interpreter to use gestural input in the context of the application as needed. By avoiding semantic "contamination" we also can make the gesture analyzer scalable.

---

20. I am indebted to Barry Arons for providing and explaining this classic nonsense sentence which originates in Chomsky's work on separating language syntax and semantics.
21. PEM is a kind of email which provides transparent end-to-end encryption of email messages; this form of email is becoming more prevalent as more sensitive information passes over public computer networks. Encryption is usually done by applying the user's secret key. Because the key itself is sensitive, it is usually stored in an encrypted form; a passphrase is needed to access the key. Of course, speaking your passphrase out loud in a situation where it can be overheard is unwise.

The next chapter describes the implementation of a gesture analyzer built along the theoretical lines described above and incorporating the major improvements of this thesis:

- It is *feature-based*, using real experimental data for feature determination;
- It is *continuous*, allowing users to make smoother, more natural gestures without requiring artificial pauses;
- It is *semantically clean*, allowing the resulting analyzer to be used in any appropriate application and to be internally scalable.

# Chapter 4 Implementation

The thesis described in the previous chapter was implemented in the form of a general, continuous-gesture analyzer as part of the AHIG demonstration system in May 1994. The analyzer and the accompanying system are described in this chapter. It is intended that the details in this chapter be specific enough to allow a knowledgeable reader to replicate the implementation if she wishes.

The analyzer module accepts data from a set of sensors, then segments the sensor data into significant intervals. *Significant* is a context-specific term; the way significance is determined is described in Section 4.4.1 "Segmenters". The intervals are analyzed for features according to the framework described previously and the features are combined into frames.

Frames are conceptually similar to frames in a movie. The analyzer produces a "movie" of the hands, where the frames show the significant changes in the hands' status. This cartoon-movie approach is a compromise in the major trade-off of a real-time system: producing information when it is needed versus producing information when enough data has been gathered to allow something meaningful to be produced. This trade-off and the accompanying design and implementation decisions are discussed in more detail below.

Assembled frames are sent to the interpreter module for possible action. Ideally, the user is never aware of the gesture analyzer's operation, though in cases of ambiguity or unresolved reference, the system may use simple synthesized speech to ask the user to repeat a gesture, or make a new gesture to clarify the ambiguity. This simple dialog repair strategy has been used in a number of places such as in the systems implementation of the "Put That There" program by Chris Schmandt and Eric Hulteen (Bolt 1984).

## 4.1 System Framework/Interaction Experience

The demonstration system we used involved two interaction scenarios:

- architects or interior decorators can create and arrange furniture and objects in a simulated room. This is an enhanced version of the scenario used in the "Iconic" system.
- witnesses or simulation creators can create and modify discrete event simulations. This is similar to the scenario described in Chapter 1 "Introduction". The simulator itself is general purpose; for the demonstration system we implemented a simulator that handles vehicles.

In each of these scenarios the user stands in front of a large-screen display and creates and manipulates the objects seen on the display.

The user wears a "data suit" which consists of a lightweight nylon windbreaker jacket with the cabling for the CyberGloves and a position sensor sewn into the lining of the jacket. This is augmented with a head-mounted eye-tracker (from ISCAN) and micro-

phone connected to a BBN HARK connected-speech recognizer. The user also wears the CyberGloves which have position sensors mounted on the forearm straps (see Figure 16). The position sensors used are Ascension Technologies Flock of Birds sensors. The system is able to sense the user's position, movements and gestures anywhere within a radius of eight feet from the screen. The complete outfit is shown in Figure 17.

Figure 16. Cyberglove and Forearm-mounted Sensor in Use

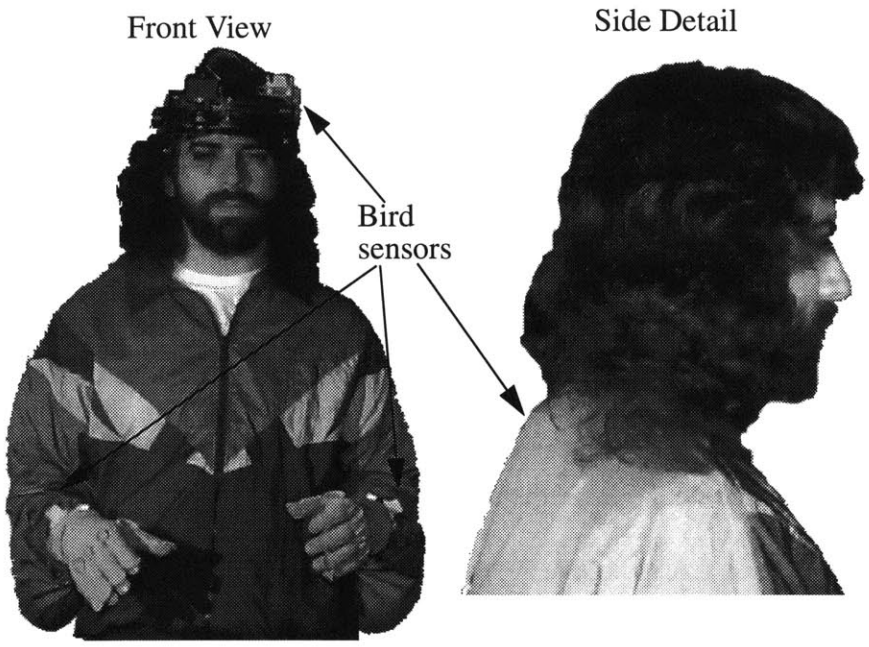


Figure 17. AHIG "Data Suit"

We recognize that this mobility restriction and requirement that the user wear specialized clothing in order to interact with the system is generally undesirable. Ultimately we expect that the instrumentation of the user's body will be replaced by unobtrusive cameras and computerized vision systems; however, that technology does not yet provide sufficiently precise information for our purposes. An interesting step forward in applying this non-intrusive technology was recently demonstrated by the "Alive" system (Maes 1993).

One of the key points of this thesis is that the gesture analysis is "technology transparent." The intervention of the body model (see Section 4.3.1 "The Body Model") allows replacement of low-level input components without requiring changes to the analyzer.

When the user first puts on the clothing, the system requires a calibration. The user is instructed to hold his hands in a series of specific positions. During this calibration, the system takes measurements of the user's body dimensions and range of movement. This

information is used both by the body model to more accurately compute information about the user's physiology, and by the gesture analyzer as detailed below. The entire configuration routine takes less than one minute to complete. This configuration is the only user-specific information required by the system; the speech analyzer is speaker-independent.

For the demonstration, we used a prepared script which exercised each of the important features. However, in normal operation, there is no prescribed sequence of actions that the user must perform. Generally, we have found that our systems perform quite well in unscripted situations. Visitors to the AHIG lab can operate the system themselves, once they are taught which command verbs are accepted. The system recognizes any of several different forms for each of its commands. The grammar is programmed into the HARK system and corresponds to the verb-based commands (such as "create," "move", "delete") which the interpreter understands. Obviously the user cannot give descriptions which are not supported by the system; however, beyond this, the only restrictions on user input are those which are dictated by the logic of the interaction (that is, you can say "Move the chair" if there is no chair, but the system will respond with an error).

As detailed in Section 2.5.2 "Directive Interfaces", the interaction style we use is directive. Each of the available commands involves the user giving a description to the system involving the objects in the system. These descriptions can be verbal, gestural, or a combination; for example, an interior designer working on a room layout might say "Turn the table like this" and make a gesture to indicate the direction and degree of rotation. The user's direction of gaze is also detected continuously and used as part of the input stream (Koons & Thorisson 1993). The user's descriptions cause changes in the state of the system which are reflected on the large display.[22]

## 4.2 Implementation Architecture

The basic architecture for the demonstration system is shown in Figure 18 below. The arrows represent the flow of information through the system. The gesture analysis module takes its inputs from a body model and sends output to the system interpreter. The body model samples the sensors and converts sensor-format information (absolute positions) into body-relative information (joint angles) before passing information on to the analysis layer.

The Interpreter module, designed and developed by Koons, is responsible for combining input from the analyzer (including information on the head, eye, hands, and body) with the output of the speech recognizer. The Interpreter uses an object/part knowledge base (OKB in the diagram) to model the world and to focus the collected user inputs into transformations of the system objects. The internal model uses a complementary three-part

22. In the implemented demonstration system, the analyzer is extended beyond the theory described in Chapter 3 "Solution"; it actually handles input from the eye tracker and information about the body position as well as the data needed for gesture analysis. This gives the interpreter a clean unified source for data about the user's body. However, no theoretical work has been done in this thesis to understand the basis of body or head gestures.

representation scheme that simultaneously coordinates spatio-temporal and categorical information. This representation, also developed by Koons, is described in his doctoral dissertation (to appear).

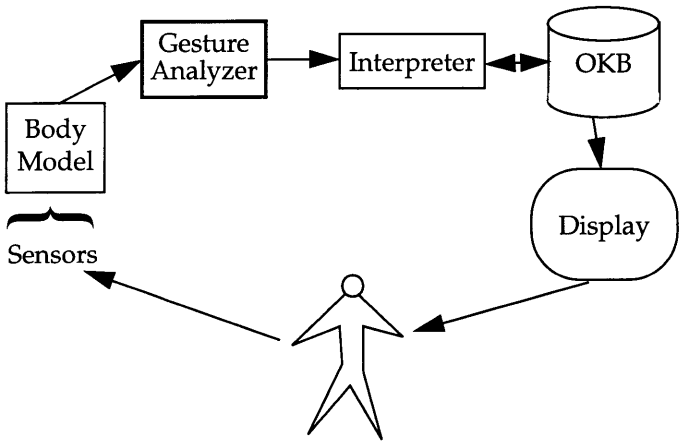


Figure 18. Data flows in the AHIG system

The transformation results (if any) are shown to the user via a large-screen display. In the current demonstration, the objects in the application are inert; in future implementations, the objects themselves will have behaviors and may initiate actions that transform the display. The display uses standard 2-D computer graphics to render a three-dimensional scene. Viewing transforms are applied to render the scene, but the transforms do not take into account the user's position relative to the display; all rendering is done as if the user's viewpoint was stationary in front of the midpoint of the display.

The effect of this is to give the appearance of viewing a large television screen rather than the appearance of gazing through a window which would be maintained if we rendered the scene using "true" 3-D graphics. In such a simulation it would be necessary to continuously track the user's head position and recompute the scene to be displayed based on the angle at which the screen was viewed. This is impractical in our demonstration environment because we do not choose to dedicate our computation resources to simple rendering tasks. In addition, our demonstrations are frequently viewed by more than one person at a time and such a projection display could properly accommodate only one viewer.

### 4.3 Conceptual Overview

Figure 19 shows the block-diagram view of how data is handled in the gesture analyzer. Subcomponents are responsible for:

- segmenting data from the body model into temporally significant states and transitions;
- extracting the key features (configuration, orientation, and position);
- motion analysis;
- path computation, which tracks the course of movement and feature

change over time and integrates the information into more complete descriptions.

The arrows into and out of the decomposition box are intended to show that each component of the analysis module operates independently, then a combined result — the frames mentioned above — is sent to the interpreter. The frames are reused in the path computation, as detailed below.

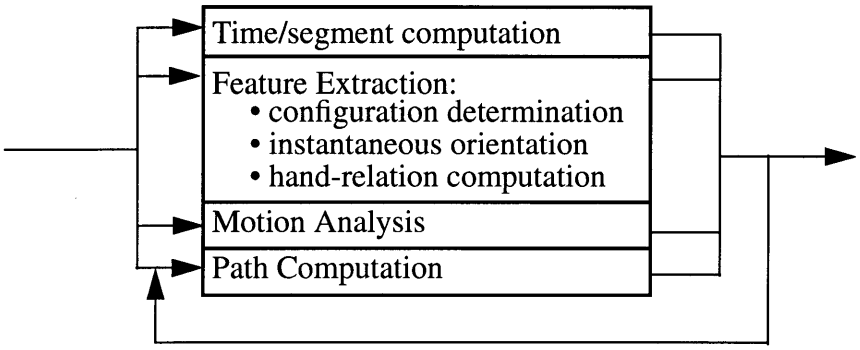


Figure 19. Conceptual Dataflow in the Gesture Analyzer

The number of frames generated for any gesture depends on how the user makes that gesture, since frames are generated only when significant featural changes occur. Each frame contains a set of slots which are filled in with the data accumulated during the time it took to generate that frame. This is a partial, incremental representation scheme, which allows some of the slots to be unfilled or filled with poor information which is improved as time passes.

This approach was adopted to allow the analyzer to deal directly with the need to adapt to different users and to differences within gestures. The analyzer has a strategy of not setting predetermined boundaries for the start and end or the duration of gestures.

It is useful to think of the analysis module as a set of special-purpose recognizers, each of which is looking for changes in the data stream that it considers significant. For example, the direction of movement of the hand may change from downward to sideways, or the hands may start a roll, as in the previous example. When such a significant change is detected, the recognizer which detected the change calls for a new frame to be generated. All the recognizers contribute their accumulated data into the new frame which is then output. All gestures result in sequences of one or more frames with the number of frames, and their specific contents, determined as a result of how the user made the gesture.

This approach allows the analyzer to deal with multi-stroke gestures, both pairs of gestures where the intervening retractions and preparation phases have been elided, as well as single gestures with complex stroke motions (such as illustrations of object paths). Because the number of frames is determined dynamically, the analyzer can continue to create frames as the gesture adds strokes.

As noted above, in order to make a continuous-analysis system useful, we must compromise between two competing desires:

- the need to be able to output a frame of information at any significant change in the feature space of the gesture, and
- our knowledge that gestures are complex and that waiting longer to output a frame of information will enable us to have a more complete and a more accurate description of the gesture.

The path computation module makes this compromise by allowing frames to be output as features change and by using frames that have been output in the past to build up an image of the gesture that is updated as more information becomes available. The path portion of the initial frames output for a given motion are impoverished and may contain incorrect information; for example, an arc motion may be initially interpreted as a line. However, as more complex data become available the path portion is filled in and corrected. In an ideal world, we would know ahead of time which features would be significant and would send only those to the interpreter. In the real world, we cannot know in advance what to send, so the analyzer compensates by making incremental improvements.

For example, imagine that an engineer explains a 360 degree rotation to the machine and makes a gesture indicating around which axis the object is to rotate. As the engineer's hand moves, key features such as the angle of orientation will change enough to be deemed significant, and a new frame showing the change in the gesture will be generated several times during the process. Ideally, we would know that a circle was being traced and could avoid generating the intermediate frames. In reality, it is not until the end of the circle that the analyzer can know what was being drawn. The details of how this is done are shown in Section 4.4.4 "Temporal Integration".

### 4.3.1 The Body Model

The body model allows the analyzer to be device-independent; new generations of sensors can be put in by modifying only the lowest level of the body model. An interface between the body model and the analyzer is defined which specifies (a) the communication protocol between the body model and the analyzer, and (b) the bytes of data to be provided by the body model and the order in which they appear. The analyzer does not need to be concerned with details of how the body model acquires the data. The interface specification is reproduced in Appendix B — Interfaces to the Gesture Analyzer.

The body model coordinates input from a number of sensor devices:

- *Virtual Technologies Cybergloves.* The cybergloves provide angle data on seventeen different angles associated with the hand and wrist. The angles measured are shown in Figure 20 on page 62. These angles give the bend and spread of the fingers, the curvature of the palm, and the pitch and yaw of the wrist.[23]
- *Ascension Flock of Birds position sensors.* These are magnetic six-degree of freedom position sensors. They provide the positional information for

each of the four measured points on the body (head, top of spine, and left and right forearms). They record the X, Y and Z positions relative to the transmitter, to an acceptable accuracy of +/-.5 cm, and the rotations around the X, Y, and Z axes (called xang, yang, and zang)[24]

- *ISCAN eye-tracking camera.* The eye-tracker uses an infra-red light and a half-silvered mirror to capture a picture of the user's eye. The image is analyzed to determine the vector of the user's gaze.

  Two artifacts in the picture are located: the pupil (a large dark circle), and the corneal reflection of an infrared LED (a smaller white circle). The relative distance in X and Y of these two artifacts vary systematically with the position of the observer's eye relative to the head. This measure of eye-to-head attitude is combined with the head location and orientation to get a vector of gaze.

The body model is responsible for sampling these sensors at appropriate data rates. In addition, the body model filters the data to provide "cleaner" data. In this case cleaner data means that the data values should not change unless a true movement or angle change is occurring. Because the analyzer uses data changes to detect feature onsets (as explained below), it is important that the changes it sees in the data streams should be valid.

To achieve this, the body model samples the gloves and cubes at 100 Hz and uses a Gaussian filter to produce 20 Hz samples with data spikes removed. The ISCAN is sampled at 60 Hz and a complex filter is applied to remove blinks and saccades (the involuntary motions made by our eyes). This filtering is possible because we know that the human eye cannot maintain and change fixations more than 5 times/second (Cumming 1978). Therefore, any fixation which lasts less than 200 milliseconds is discarded as a saccade.

The body model uses sensor data to compute joint information (positions and angles) before passing information on to the analyzer. This is necessary because we wish to minimize the instrumentation of the user's body. As shown in Figure 17 on page 55 we have four position sensors on the user's body, and angle sensors for the joints of the hand (see Figure 20) provided by the Cybergloves.

In an ideal environment we would like to have complete information on the user's body — for gestures this means getting the angular information on the elbow and shoulder as well as the wrist information provided by the Cybergloves. In the current system this

23. Roll, pitch, and yaw are ways of specifying movement around a set of axes such as the approach/normal/orientation axes defined in Figure 10 on page 47. In the case of the hand, roll is movement around the approach axis, pitch is movement around the orientation axis, and yaw is movement around the normal axis.
24. Note that the rotations of the Bird sensor cannot be directly translated into roll, pitch and yaw — they can only be determined with respect to the coordinate system of the cube doing the measurement. The mapping between coordinate systems and the creation of a unified spatial representation out of the varying sets of numbers available as input is one of the more complex computations carried out by the body model.

information is supplied by the body model, which computes the angular information via a process known as inverse kinematics.

Inverse kinematics — described in detail in *Simulating Humans* (Badler 1993) — is a constraint-satisfaction process. Knowing the fixed positions of certain points on the body — and knowing the constraints on motion imposed by the configurations of human body joints, it is possible to solve the constraint equations and determine the positions and angles of the body joints which are not directly instrumented.

In our demonstration system, the body model knows the fixed point at the top of the spine and the moving point on the back of the forearm. The position of the shoulder is computed during the configuration that the user does at system start-up, and then treated as a fixed point. The body model computes the position of the elbow and the shoulder and elbow angles.

Shoulder angles are:

- elevation — the degree to which the shoulder is raised above straight down;
- abduction — the angle of the raised shoulder away from the center of the body;
- twist — the turn of the shoulder inward/outward from the belly.[25]

Elbow angles are:

- closure — how much the elbow moves the wrist toward the shoulder;
- roll — the rotation of the elbow which produces a roll of the hand.[26]

The angles produced by the body model for the elbow and shoulder joints are used by the analyzer to decompose complex movements of the hand into recognizable segments for analysis.

### 4.3.2 The Interpreter

The Interpreter module, designed and developed by Dave Koons, accepts inputs from the gesture analyzer and from the HARK speech recognizer. It assigns semantic value to the collected input information by use of a using a multi-representation system. This system permits the interpreter to take advantage of the dual nature of much multi-modal input: categorical and spatio-temporal. The interpreter describes object parts in terms of a tree of relationships which change as the multi-modal description provided by the user evolves.

---

25. Imagine stabbing yourself in the belly with a knife held in your closed fist.
26. This roll actually occurs by the two long bones of the forearm (radius and ulna) crossing over each other, but it is treated as a property of the elbow joint because the wrist joint does not move in this angular rotation and all changes are assigned to the nearest moving joint.

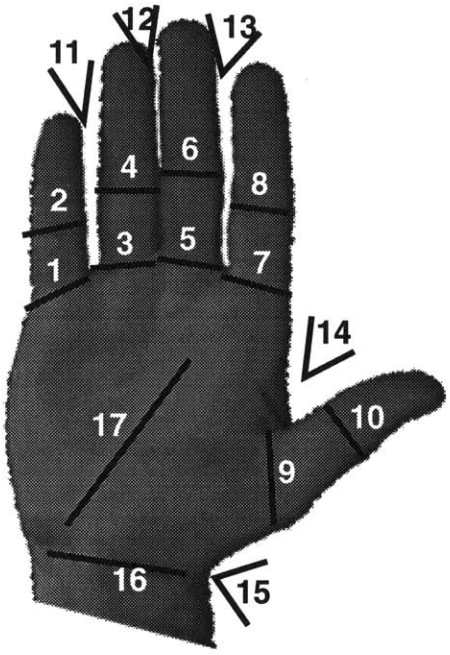


Figure 20. Joint Angles Measured Directly by Cybergloves

**Table 4: Key to Cyberglove Joints**

| # | Name |
|---|---|
| 1 | pinkie metacarpal |
| 2 | pinkie pip (2nd joint) |
| 3 | ring metacarpal |
| 4 | ring pip (2nd joint) |
| 5 | middle metacarpal |
| 6 | middle pip (2nd joint) |
| 7 | index metacarpal |
| 8 | index pip (2nd joint) |
| 9 | thumb metacarpal |
| 10 | thumb ip (1st joint) |
| 11 | pinkie-ring abduction |
| 12 | ring-middle abduction |
| 13 | middle-index abduction |
| 14 | index-thumb abduction |
| 15 | wrist yaw ("side-to-side" relative to forearm) |
| 16 | wrist pitch ("up-down" relative to forearm) |
| 17 | palm arch |

For the most part, this internal structure of the interpreter is invisible to the gesture analyzer. Semantically, the analyzer does not concern itself with the relationships and other knowledge-based parts of the interpreter. However, in the demonstration system, the analyzer and the interpreter are implemented in different languages and run on different hardware. Since we needed in any event to solve the problem of transferring data between the analyzer and the interpreter, we created a specification (shown in Appendix B — Interfaces to the Gesture Analyzer) which permits the analyzer to form its data into Lisp expressions which can be transferred to the interpreter and evaluated directly into forms the interpreter can use. The data transferred is described in "Path Analysis" on page 67.

In addition, the interpreter depends on the analyzer to do a number of computations over the basic featural information. This is, to some extent, a violation of the thesis principle of purely separating analysis and interpretation. However, it should be noted that the analyzer is capable of any of a variety of computations, summations, or higher-level formations over the basic featural data. It seems, though, that unless we wish to program an analyzer which performs all of these computations, only a subset can be present. For the demonstration system, naturally, the gesture analyzer supports the set of computations

which is most useful to the interpreter which Koons is implementing for his dissertation. In the current demonstration system, these computations involve:

- representing the data on arc segments as a center plus start and end angles;
- representing the data on orientation change as a pair of start and a pair of end vectors (and discarding the categorical information).

## 4.4 Detailed Analyzer Architecture

The details of the analyzer internal architecture are shown in Figure 21. Data flows from the top of the picture toward the bottom, as represented by the lines. Our current prototype is implemented in under 5,000 lines of C++ and runs on a DECstation 5000 under UNIX. Each layer is implemented as a separate class to allow easy overriding and extension of the model. This allowed us to extend the purely gestural portions of the analyzer to handle data about the eye and torso for the demonstration system.

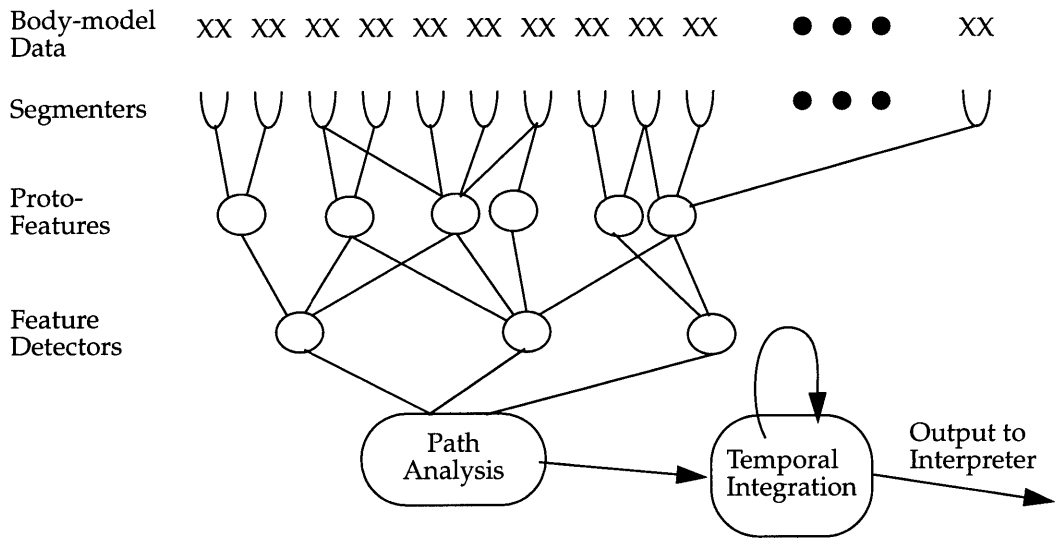


Figure 21. Internal Architecture of the Gesture Analyzer

### 4.4.1 Segmenters

At the top level of the analyzer architecture are segmenters which are responsible for receiving the synchronous 20 Hz data from the body model and dividing it up into periods of movement and inaction. Each data value produced by the body model is watched by one segmenter. Segmenters embody knowledge about what kind of data they are segmenting and use this to detect only significant changes.

Each segmenter has an epsilon value associated with it; only changes greater than epsilon are considered significant. Epsilon is a variable value which allows the analyzer to take advantage of two kinds of knowledge. First, the analyzer knows what each segmenter corresponds to and can adjust epsilon to be more sensitive for data values which are more important. For example, the change in angle of the thumb and forefinger is more impor-

tant than the changes in angle of the ring and pinkie finger. This is a condition of how we as humans make our gestures —we assign more importance to certain parts of the hand than to others.

Second, the epsilon value for segmenters can be dynamically adjusted in response to what is happening. For example, when the user is not making any movements, the value of epsilon can be set higher so that incidental motions are not picked up by the analyzer. Conversely, when the analyzer detects that a significant movement is taking place, the value of epsilon for the motion-detecting segmenters can be set lower so that fine-grained details of the motion such as changes in direction or pauses (which might show divisions in multi-stroke gestures) can be picked up.

Segmenters output data structures called (naturally) segments. These are diagrammed in Figure 22. A segment represents a steady state or continuous change in the data value that the segmenter is watching. This is shown graphically in Figure 23.

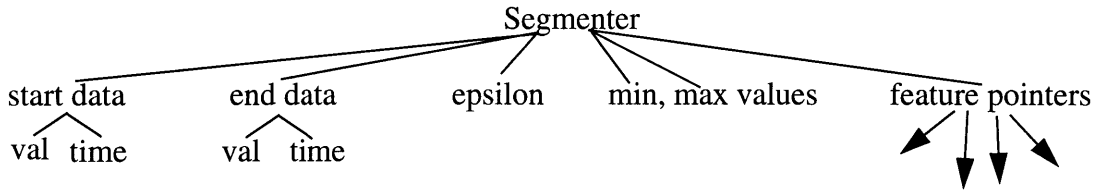


Figure 22. Segment Data Record

Segmenters have two value/time pairs which represent the data change for this segment. In steady-state segments (such as numbers 2 and 6 in Figure 23), the data values are the same; in segments which record changes, they are the way the analyzer knows that "a value changed from V1 at T1 to V2 at T2."[27]

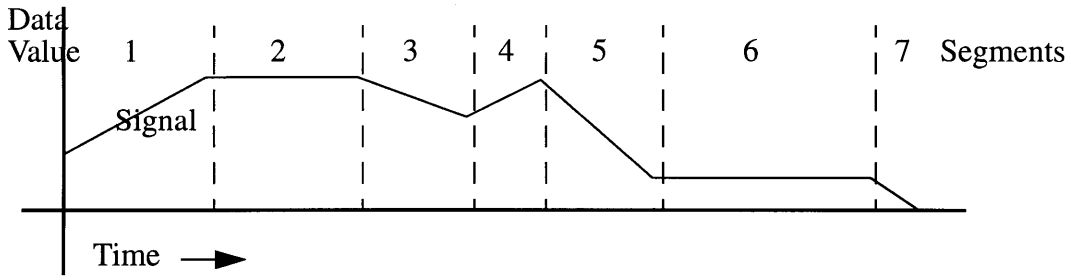


Figure 23. Conceptual View of Data Segmentation

Epsilon is the "fudge factor" which is used to smooth out noise not eliminated by the body model as described above. Note that in Figure 23, the segment boundaries appear slightly

27. Of course raw input data is not as smooth as the signal line shown in Figure 23; however, remember that the data received by the analyzer has been Gaussian filtered by the body model to eliminate spikes and "jitter" in the signal.

after the changes in the data value. This is a necessary consequence of anticipating an unreliable signal. The segmenters wait until a change exceeds epsilon before marking a change boundary, or until a changing value falls below epsilon before marking the end of the change. In practice, this is usually not a significant problem. Users report that they feel the computer presence and the data suit as constraints and do not move as much as they would in unrestricted circumstances

The segmenter *min* and *max* values are used to determine how significant a particular value is for a given body part. When a feature is attempting to assign categorical information, the min and max values of the segmenter are queried and used in the feature computation. For example, if the analyzer knows that the wrist pitch is 75 degrees it can say that the wrist is canted upward. However, if it also knows that the maximum value for that data is 80, then it can know that the wrist is canted upward "a lot." This kind of information is essential to determining categorical features, as it provides a correlation between the raw numbers given by the body model, and the range of possibilities that the human body can undergo.[28] The min and max values for each segment are provided initially by the body model, which records them as part of the calibration described in Section 4.1 "System Framework/Interaction Experience". The segmenters may also adjust these values. Since the calibration itself is not precise and since users might switch places without re-performing the calibration, the segmenters detect if a value is received that is outside the range of pre-determined values. If this happens, the affected boundary is adjusted.[29]

The feature pointers are links to the feature objects (see "Features" on page 66). The segmenters are maintained in an array whose size and composition is kept as parallel to the data values from the body model (shown in "Appendix B — Interfaces to the Gesture Analyzer" on page 89); that is, there is one segmenter in the array for each data value the analyzer is interested in. At software start-up time, the features "register" themselves with the segmenters. They do this by providing a C++ object pointer to the function object. This object pointer gives the segmenter a handle to use when a segment has been detected.

When a new segment has been detected, the segmenter loops through all the registered feature pointers and calls a "notify" method on each feature object. The notify method is passed pointers to the start and end data pairs; in this sense the segmenters "hand off" data segments to the features for operation.

---

28. This process is similar to what is used in *fuzzy logic*, which was developed by Lofti Zadeh (Zadeh 1982, but see Kosko & Isaka 1993 for a more readable summary). In fuzzy logic a variable or system can be more or less in some state: "mostly hot" "partly dry." This is determined by computation of percentage values and set intersections. Categorical features have a strong resemblance to fuzzy values, so a similar computation technique is appropriate.
29. Note that this takes care of the case where a more limber or larger person takes over; however, if a smaller or less mobile person puts on the data suit without redoing the calibration, there is nothing the analyzer can do.

### 4.4.2 Features

Beyond the segmenters are proto-feature detectors and feature detectors. These detectors work by monitoring one or more segmenters and receiving asynchronous reports of states and events appearing in the data. The proto-feature detectors look for simple things like extension of an individual finger. They in turn feed their data to higher-level feature detectors, which combine these reports into more intelligent features. For example, extension of all four fingers, each of which is detected by a single proto-feature, and lack of curvature in the palm (another proto feature) can be seen to be a flat hand configuration. Hand configuration is a feature. Generally, those things listed in Table 3, "Features Used," on page 46 are implemented as features; lower-level information needed for these features is implemented as proto-features.

The features and proto-features used in the implemented system are listed in Table 5 on page 67. The analyzer is designed to be both flexible and extensible. New features can be added as needed to extend the system into areas such as head and torso gestures and to integrate eye-tracking information — detecting eye-specific features such as fixations and smooth pursuits.

The data structures that make up the feature object class are shown in Figure 24. Each feature is initialized with a set of *needs* — those data that must be present in order for the feature to "fire." A certain number of data segments can be specified to be present, and a stack of specific types of data can also be specified. This information is initialized at start-up time and remains fixed throughout analyzer operation. Needs are always a strict superset of the segments in which the feature expresses interest.

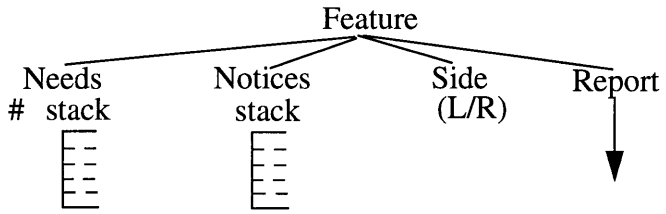


Figure 24. Feature Data Record

The *notices* stack keeps track of the segments which have reported in. At any time, segmenters can report asynchronously, depending on the changes they detect in the incoming data stream from the body model. If a segment reports a new value it is added to the stack; the stack is then compared to the needs to see if the requisite number of segments have reported and if all the required segments have notified the feature with data.

An advantage of this arrangement is that it allows the feature detectors to be less rigid about what constitutes a feature. Each feature object maintains its own stack of segmenter information and only declares a feature to have been detected when all the correct information is in place. However, what constitutes correct information can vary from feature to feature. For example, hand shape is determined by the curvature of the palm and the angular reports of the four fingers. But any three of the fingers are sufficient to determine

a configuration. This allows configurations to be reported even when one of the fingers is extended, without having to program specialized knowledge about each specific configuration. This flexibility shows that the feature-based approach provides significant savings over a conventional template-based approach.

A set of needs can be regarded as a kind of template or pattern detector but, as discussed in Section 2.5 "AHIG Approach", what is important is that (a) the matching being performed is malleable and dynamic and (b) that in this matching process we do not go directly from the template to meaning.

**Table 5: Features & Protofeatures Implemented**

| Type | Number |
|---|---|
| Extended | 10 — 1 per finger |
| Contracted | 10 — 1 per finger |
| Curvature | 2 — 1 per palm |
| Spread | 2 — 1 per hand |
| ThumbPos | 2 — 1 per thumb |
| Move | 2 — 1 per hand |
| Orientation | 2 — 1 per hand |

When the notices match up with the needs, the feature *reports* the presence of a feature to the current frame. The report slot is a function pointer to one of a set of functions. The specific function used for each feature is assigned at system start-up time, but this architecture allows similar features to share reporting functions.

For example, in the implemented system all the features which track finger extension use the same reporting function; however, each feature is notified by a different set of data segments, corresponding to changes in different fingers. In a hypothetical extension, the same reporting function could be used by a feature which tracked arm extension by monitoring shoulder, elbow and wrist angles at the segment level.

Lastly, each feature keeps a record of whether it is a left-side or right-side feature. This simple symmetry halves the required number of data structures.

Features are kept on a stack in the analyzer. As each feature is initialized at start-up time, it "registers" itself with one or more segmenters, as described above. It is then placed on the stack. At each time increment, each feature on the stack checks to see if it has all the required features and if so, it computes its particular value (for example, the degree of curvature for hand configuration) and reports that into the current frame.

### 4.4.3 Path Analysis

When a complete feature is ready to be reported, the current status of all the higher level feature detectors is sampled by the path module, which produces the completed description of the gesture in a new frame. The frame object is quite complex, as it records information about all of the features. This is shown in Figure 25.

Most of the information in each frame is duplicated for left and right. The only exceptions to this are boolean values which (a) record whether the information in a frame has been output yet or not and (b) record whether the information on left and right hands in this frame is coordinated.

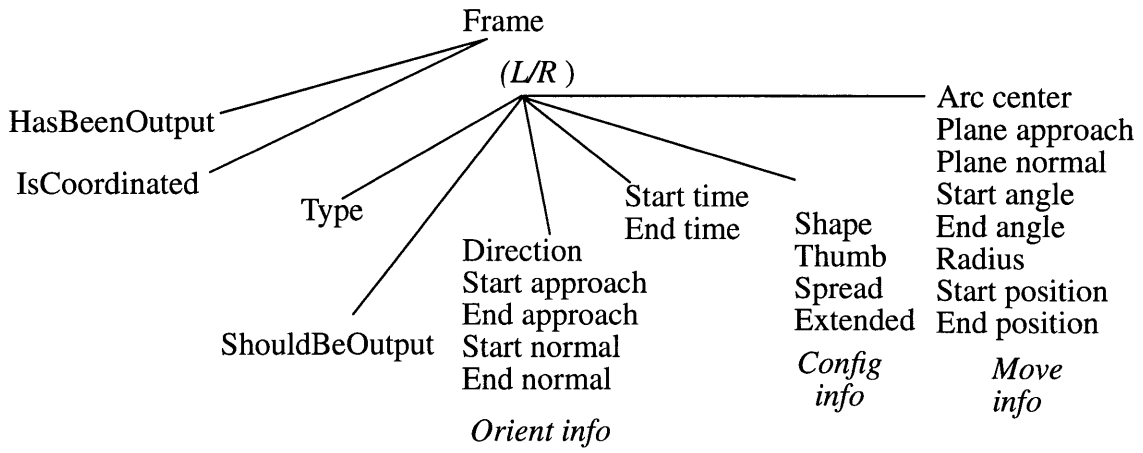


Figure 25. Frame Data Record

The first value is necessary in order to avoid duplicating information sent to the interpreter. Frames are output as requested — the interpreter sends a signal to the analyzer which immediately dumps out any completed frames which have not yet been output. However, temporal integration (described below) requires that frames be kept around so their information can be compared with new information. Therefore, the analyzer maintains a "history" stack of frames as they are generated; temporal integration and output to the interpreter happen asynchronously, so state information has to be kept.

The coordinated value is simply output to the interpreter along with the left/right information described below (see Figure 27 also).

Frames are *typed* based on the highest-level category of feature detecting in the current time-segment. Feature types are ordered as shown below in Figure 26. The ordering shown indicates that each succeeding type includes the information from the lower-level types. That is, ARC subsumes LINE subsumes ORIENTATION subsumes CONFIGURATION.


Figure 26. Ordering of Frame Types

It is possible that an ORIENTATION and/or a CONFIGURATION change might happen at the same time as a movement (LINE or ARC) change. If so, the frame that is produced contains the information on both (or all three) types of change that occurred. However, the primary information and computation for movement is considered to be the most important, so the frame is oriented toward the movement.

This is a logical consequence of the way we perceive the world. Generally, the larger the motion, the more of our attention it occupies. A movement of a line or arc type is going to be larger than the movement involved in changing orientation or changing the hand's con-

figuration. Therefore, the analyzer focuses on the largest motions; however, information about the other kinds of features is still passed along in the frame. The reason for this is that we cannot know in advance the meaning of the gesture. It is possible that a motion and/or orientation change may be the important part of the frame. For example, the user may be moving in order to point at something (an orientation change). In this case, the important thing for interpreting the gesture — knowing what the user is pointing at — is that the analyzer know the final position and orientation of the hand, and the fact that the finger is extended.

The frame also coordinates knowledge about physically separated parts of the body. For example, information about the left and right hands is compared (as described in Chapter 3 "Solution") to see if the user is making a coordinated presentation such as using two hands to show the movement of a table, or separate presentations such as showing the placement of a chair next to the table.

Interaction with the interpreter is done on a demand-driven basis. The analyzer works at its own pace detecting features and creating frames as they occur. These are stacked as they arrive and are integrated (as explained below). The integration process tags certain frames as being important for output. When the interpreter asks for output, the analyzer dumps the frames which are tagged for output, then marks them as having been output.

The output format is specified by the description in Appendix B — Interfaces to the Gesture Analyzer. As noted above, the analyzer creates text forms of Lisp expressions so the interpreter can directly evaluate the information and initialize its own representations from the analyzer's frames. Figure 27 shows a summary of this transfer format.

| TIME | POSITION/MOVEMENT | ORIENTATION |
|---|---|---|
| (tstart tend) | (POINT point)<br>(LINE pstart pend)<br>(ARC arc information) | (start normal & approach)<br>((start normal & approach)<br>(end normal & approach)) |

Figure 27. Format for Data Transfer to Interpreter

The time interval is sent first — this is the time over which the feature occurred. Then one of three formats is used:

- POINTS are for ORIENTATION and CONFIGURATION changes;
- LINES are for linear movements;
- ARCS are for non-linear movements.

If there is an orientation change, two pairs of vectors are used, otherwise a single pair suffices. Arc information is the center, radius, start and end angles, and two vectors which define the plane of the arc. Arc information and the determination of whether a movement is an arc or a line are done by a single computation based on a projected possible center of motion. Imagine that the analyzer receives the motion shown in Figure 28. This motion is

composed into two frames as shown in Figure 29. The division into two (or more) frames occurs, as explained above, because there is some featural change in the motion — for example, the user may make use of different muscles in creating the smooth motion.

The computation is then performed as shown in Figure 30. The start and end points of each frame are projected onto a plane, then taken as the end points of coplanar lines.

Any motion the user makes that takes only one frame is assumed to be a line; this may be corrected based on later information. When the second (or subsequent) frame is completed it is used to provide a second line for comparison. The perpendicular line to each line segment is computed and the intersection of the perpendicular lines is computed.

If the center is too far away, the line segments are assumed to be colinear. "Too far away" is 100 cm in the implemented system. This value was chosen based on observations of the human body: the largest circle humans can make by moving just their arms (i.e. rotating the shoulder only) has a radius in the range of 70-80 cm. To make an arc of larger radius than this requires moving the torso, which leads to additional frames being generated and to a non-continuous arc motion.

Overall, the analyzer attempts to err on the side of producing too much information, rather than too little. For example, if the input is a forward motion of one hand, it is possible that a scientist is simply commanding a robot to move forward. In this case, the precise geometry of the motion is unimportant; all that matters is that a "go-forward" command be relayed to the robot. However, in another instance, the scientist could be making the same gesture to select a particular object. In this case, the forward motion is relatively unimportant; what matters is the geometric information, which can be used by the interpreter to pick out the desired object. The analyzer cannot know a priori which case is occurring based only on the motion.

This over-information is a necessary consequence of the separation between analysis and interpretation. Since the analyzer cannot know what people "mean" by any given gesture, all the potentially relevant features of the gesture must be captured and passed on. Similarly, the requirement of storing all this information in the frame means that the object must be expanded as new features are added.

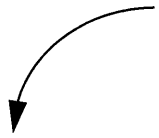

Figure 28. Example Motion

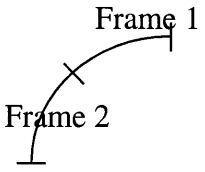


Figure 29. Motion Divided into Frames

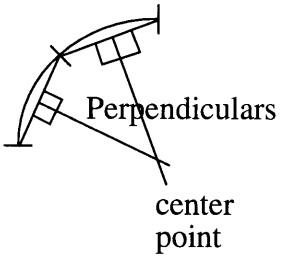


Figure 30. Center Computation over Frames

### 4.4.4 Temporal Integration

Finally, the generated frames are given to the history module for integration. The history module performs two major functions: it handles the interaction with the interpreter and performs temporal integration over the frames. The interaction with the interpreter is described above; this section explains temporal integration, which is one of the key advances of this implementation.

The generation of a frame involves summation over "space" — where space is thought of as the space of all possible features. Once a new frame is put together, it is then compared to past history to see if the information can be integrated over time. Temporal integration encompasses a number of different functions:

- *error correction*: when a movement is initially detected, the analyzer cannot tell what sort of movement is being performed; therefore, movements are initially all classified as LINE moves. However, a new frame may arrive showing a new movement. The analyzer determines if the movement is a continuation of the line, or if t here is a curve involved. If there is a continuous curve involved, the previous frame is edited to change its type to ARC. The arc information in the current frame is then integrated, as described below.
- *continuous-change integration*: When the analyzer has accumulated sequential information on a particular change, it merges the information to create the temporally largest possible descriptive frame. For example, part of a linear movement might be seen going from (X1, Y1, Z1) at time T1 to point (X2, Y2, Z2) at time T2 — then another part of the same line is seen going from (X2, Y2, Z2) at time T2 to (X3, Y3, Z3) at time T3. Assuming that the history module determines that the movements are in fact colinear, it merges the frames to report only a LINE movement going from (X1, Y1, Z1) at T1 to (X3, Y3, Z3) at time T3.
- *compatible-feature integration*: as described above, the analyzer keeps a hierarchy of features. However, since all features available for a given time segment are reported, the history module can integrate non-conflicting features which occupy non-conflicting time slots. For example, a configuration change might be recorded from time T1 to T2 and an orientation change recorded from T2 to T3. However, if no configuration changes happen in T2 to T3, the analyzer can know that the hand configuration is the same and report the paired configuration and orientation change over the interval T1 to T3.

  This has the nice effect of making the analyzer insensitive to a number of minor variations on the way people make gestures. For example, if you move your hand first and then point your finger, or point your finger first then move your hand you will get the same result from this gesture analyzer.

The process of temporal integration can be thought of as a further step in a scalable process. This process acts to remove artificial divisions in the data stream that are introduced by the digitization and analysis process. This is represented graphically in Figure 31.

At the top of the diagram is the initial gesture stream which is divided only by the nature of the user's movements. We then sample that stream at a fixed rate of 20 samples/second. We expect that this rate is fast enough to pick up the important divisions in the gesture stream.[30] However, it also introduces a large number of unnecessary divisions.

Most of these divisions are removed by the first layer, the segmenters, because nothing changes from one division to the next. Different segments are combined into features, which serves to remove further divisions. Temporally-related groups of features are combined into frames, again removing divisions. Then the history module does its temporal integration.

At the end of the process, the analyzer should have recaptured all the important divisions. This is what is implemented for the current demonstration system. However, there is no theoretical reason that the integration should stop at this level. The gesture analyzer is implemented with a scalable architecture which permits additional levels of integration to be achieved. In the demonstration system, this would mean implementing new C++ classes to handle the collected data. The advantage of this approach is that the analyzer can become increasingly sophisticated and create more elaborate and sophisticated descriptions of gestures as needed.

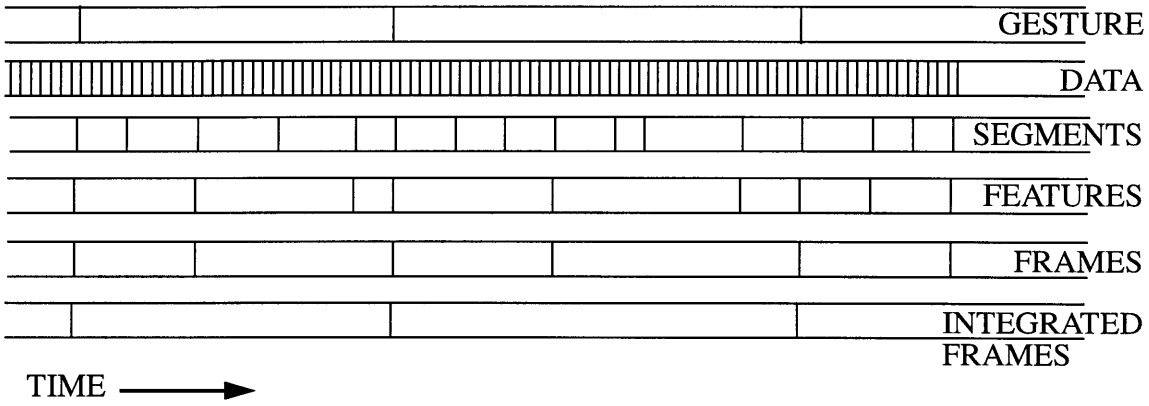


Figure 31. The Process of Temporal Integration

In practice, temporal integration looks like the process shown in Figure 32 through Figure 34. The user has made an arc motion which is divided up into 4 frames. As in the previous example, these frame divisions come about because our physiology causes variations in the way we make apparently-smooth motions. Once the computation shown in Figure

30. This is a consequence of the human anatomy. Try moving your finger as fast as you can — if you are like most people, you will have trouble moving it back and forth more than 5 times per second.

28 through Figure 30 (above) is done, the frame information looks like the middle figure (Figure 33). Both frames are on the history stack at this point.

The temporal integration module notes that these are two frames of the same type (ARC), with the same center. The information from the first frame is then merged into the second to give a new, integrated frame. In this example, the start angle and start position are changed so that after integration, Frame 2 effectively contains all the information from Frame 1. Frame 2 is then changed to have the start time of Frame 1. Finally, the temporal integration module marks Frame 1 as not needing output, since Frame 2 now contains more complete and correct information and covers the same time span.

This process continues through the integration of Frames 3 and 4 until at the end, Frame 4 contains all the information from — and covers the time-span of — previous frames and only Frame 4 needs to be output to the interpreter.

An interesting effect of performing the updating this way is that temporal integration becomes a very fast process. In theory, an arbitrary number of antecedent frames could be on the stack at any give point; therefore, the temporal integration module would have to look arbitrarily far back into history to determine which frames to merge. However, since each successive frame is merged as it is put onto the stack, the temporal integration module only needs to look back in time one frame. That is, when Frame 3 is put onto the stack, it only needs to be merged with the information from Frame 2; Frame 1 is never considered.

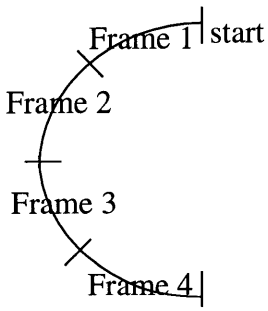


Figure 32. User's Motion (with Frame Divisions)

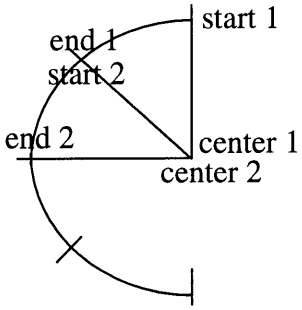


Figure 33. Frame Information Before Temporal Integration

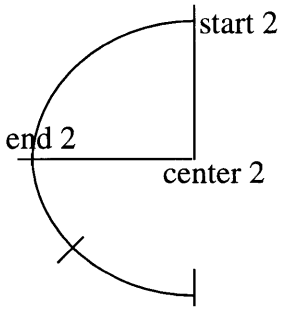


Figure 34. Frame Information After Temporal Integration

## 4.5 Analogy with Visual System

The reader may have noticed that Figure 21 bears some resemblance to diagrams of the human visual system. This is not accidental. Our low-level human visual processing operates in much the same way as the gesture analyzer: it receives a constant stream of input from the world and processes that input into key features (edges, texture gradients, motions). The structure of the feature objects (see "Features" on page 66) is roughly analogous to the structure of low-level neurons which receive input (photons) and report (fire) when the input matches specific criteria for that neuron. The reports of individual neurons

are combined into higher level representations in our visual cortex in a manner analogous to the way the gesture analyzer performs its temporal integration.

The features detected at this level are then combined to give us high-level internal representations of complex visual scenes. Our vision system is "programmed" by evolution; the features — such as edges, horizontal and vertical lines, and motions — it can detect and assimilate are pre-determined and the representation of those features in our brains is already set. By contrast, our efforts to find the key features of gestures and come up with appropriate computer representations is at best a first pass. We have some experimental and operational evidence to show that we are on the right track, but no conclusive proof.

This model seems to be appropriate for structuring prototype systems we build; however, no claim is made about possible similarities between our human perception of gestures and the computerized analysis done here.

# Chapter 5 Analysis & Conclusion

One of the advantages of building a demonstration system is the ability to validate the ideas inspiring the implementation. Similarly, an implementation exposes weaknesses and areas where the theory fails to meet with reality. This chapter reflects on the underlying theory, based on implementation experience and suggests ways in which the research might be moved forward from here.

The theoretical basis described in Chapter 3 "Solution" rests on high-level observation of gesturing. MacKenzie (Mackenzie 1994) and other researchers in the area of human prehension (grasping) have begun work at a much lower level to understand precisely how human hands interact with objects. Their research should be considered in revising the feature set used in future work.

## 5.1 Limits of the Implementation

In some ways the gesture analyzer in the current demonstration system does not use the full theoretical process outlined in Chapter 3 "Solution". Some information available from the analyzer is not passed on to the interpreter. For example, the categorical information about orientation (facing inward, right, and so on) is not used by the current interpreter, though it is used by the analyzer. Therefore it is not passed on. In addition, the output format is heavily tied to the needs of the interpreter. In the abstract, we would like to maintain a clean semantic separation between the analyzer and the interpreter. In practice, though, it seems unnecessary when there is only one client of the analyzer to output data in any format other than that required by the client. Future implementers may also find that theoretical firewalls cannot be efficiently maintained in the constraints of real implementation.

Some information that might be computed by the analyzer is not computed due to basic resource limitations. In "Position, Orientation, Configuration" on page 47 two different types of coordination were identified, one based on mirroring of movements and one based on an axis/motion pairing. In the current implementation only the first type of coordination is checked for. Mirrored motion is simple to detect; by contrast determining all the possible axes and orbits of which the hand is capable and checking against each of them is computationally very expensive. Therefore, this form of coordination checking is not done in the current implementation. With the advent of more powerful computers[31] it should be possible to complicate the mathematics performed by the analyzer and still be able to supply near-real-time response to the interpreter as needed.

One troubling issue that arose during the implementation was the question of control and synchrony versus asynchrony in the interaction between the body model and the gesture analyzer. In particular, the lowest level of temporal integration (shown in Figure 31 on page 72) done by the segmenters could theoretically have been done by the body model.

31. At the time of writing, AHIG was in the process of installing its first DEC Alpha processor, which promises to greatly increase the running speed of the analyzer code and of future implementations.

In this implementation, the segmenters were made part of the analyzer. The trade-off in this case was increased net traffic — as the body model must send out much more data — versus control. With the segmenters in the analyzer, they are able to be "in control" in the sense that the segmenters asynchronously signal the proto-feature detectors when a significant data segment has been recognized. With the segmenters in the same process as the feature detectors this is easy to implement. With them running on different machine architectures across a network it would have been significantly more difficult.

## 5.2 Limits of the Body Model

The decision to use joint angles for data representation was made within the AHIG group even before the "Iconic" system implementation was finished. Joint angles, which are not necessarily natural perceptual features, were expected to be a form of data representation much more amenable to computation and reasoning than raw positional information. Implementation experience has not completely borne this out. Unfortunately, the human body is not a simple construct and joint angles do not directly map onto smooth changes in the data. The complexity of the shoulder movements alone was the source of a great deal of headache.

However, while not perfect, joint angles are a simpler representation than raw positional information and significantly reduce the need to fully instrument the user's body. Without the inverse kinematics available from the body model in the demonstration system, the gesture analyzer would have performed significantly worse.

## 5.3 Analogy with Visual System

In Section 4.5 "Analogy with Visual System" the architecture of the gesture analyzer was compared to the superficial organization of the human visual system. In many respects, the gesture analyzer performs similar tasks as the visual system. However, it should be kept in mind that this is a conceptual analogy only. As noted, there is a whole set of evolutionary developments built into our visual system. Everything — from the fact that color perception drops off toward the periphery of the visual field to the fact that we seem better able to perceive horizontal and vertical lines than angled lines to the fact that human babies seem to have preferential identification of human faces — is determined by fixed and unchangeable structures in our bodies. In picking data to segment and features to detect for a computer, no such advantage could be used. As noted in "Features Selected — Thesis Basis" on page 45, the particular features used for this thesis were the ones which most succinctly described the gestures made by experimental subjects.

In creating a gesture recognizer the most influential principles taken from vision were the ideas of hierarchical organization and increasing abstraction and the idea of attempting to resolve the important divisions out of a world rendered with too many divisions (as described in Section 4.4.4 "Temporal Integration"), a process familiar to computer-vision researchers.

## 5.4 Analogy with Speech Recognizer

One of the inspirational thoughts for this thesis was the idea of making a gesture analyzer that would function in many ways like a speech recognizer. Interestingly, our gesture recognizers have suffered from some of the same problems as speech recognizers.

Speech recognizers necessarily must deal with any sound reaching the microphone. Ambient sounds such as slamming doors or colleagues conversing behind the speaker are picked up and may be "understood" as speech by the recognizer. Of course, the result is often gibberish. Similarly, the gesture analyzer functions continuously as long as it is receiving data from the body model. This allows us to avoid artificial demands of making the user signal a gesture's start and end (as was done by Murakami and Taguchi) or making the user insert pauses (as was done by Sparrell). However, it means that any time users moves their hands, the gesture analyzer treats that motion as a gesture to be encoded.

One simplistic solution to this problem may come from implementation of a user attention model (Thorisson 1993). If the system knows, perhaps by analysis of head and body postures, that the person is not addressing the system, then data from the analyzer can be effectively shut off and turned on at appropriate times.

Another shared problem of speech recognizers and this demonstration system comes from the existence of mumbling or slurring in both speech and gesture modes. As noted in "Experiment Results" on page 41, several subjects made gestures that were artificially constrained by their own choice (such as keeping their hands on their knees). In connected-speech recognition, a significant problem is differentiating one word from another:

- There is variability in single-word pronunciation when the word is in context (how many syllables in "chocolate" and how many in "chocolate drink").
- There is coarticulation where the end part of one word is elided or combined with the start of the next (try saying "The table" and "That table").

In gestural interaction, the first problem is not so serious; the gesture analyzer is built to handle relatively large variations in how gestures are made in context. Similarly, speaker-independent connected-speech recognizers can usually handle these problems.

However, coarticulation is a serious problem. In gestures, this can happen simply by people being imprecise in how they make their gestures. For example, let's say that the witness wishes to create two roads as part of her accident description. She says "Put a road here and another here" and makes the gestures shown in Figure 35. Note that the text of the utterance is broken up in the picture to show where gestures occur during speech. Note also that the analyzer sees three gestures, one of which is the movement of the hand from the position at the finish of the first "road" to the position at the start of the second.Now imagine the witness is a little less precise in her gestures: her initial stroke is not perfectly vertical and she does not pause at the end of that stroke. Her gestures might resemble those of Figure 36. If this is done in too much of a hurry and with sufficient cur-

vature of the downward stroke, the analyzer will recognize three different gestures, as shown in Figure 37.

What has happened here is that the analyzer has seen the downward stroke in several segments and at some point the segments have deviated sufficiently from colinearity with their predecessors that they are no longer recognized as part of the same line. The analyzer then begins thinking of them as an arc and (in this case erroneously) merges them with successive segments which can be approximated to the same curve.

This erroneous reconstruction of the input is precisely analogous to a speech recognizer mishearing "the table" for "that table" when the user does not sufficiently articulate both the T at the end of "that" and the T at the beginning of "table." However, it is a serious question whether improvement in the gesture analyzer in this case is necessary or appropriate.

For one thing, people generally do not gesture like the example in Figure 36. Only two experimental subjects were gestural "mumblers." Second, if we mumble our words, we learn to expect people to have trouble understanding us. People who make imprecise gestures end up expecting their gestures to be misunderstood. If they wish their gestures to be precisely meaningful, they make their gestures more carefully. In this respect the analyzer performs as well as a human observer, which is about all that can be expected.

In another respect, the advance in this thesis in accepting *continuous* gestures also has a parallel in speech technology. Early voice recognition technology allowed only discrete speech. These recognizers force users to put artificial pauses between words. Similarly, the previous gesture recognizer ("Iconic") required users to stop between gestures. While the difference does not seem large at first, the difference to anyone who has used both forms of technology is amazing. Connected-speech recognizers enable a much more natural form of interaction. From the user's point of view, they totally reshape the interaction with the system. This is also true for connected-gesture recognition.

From the surface appearance, not that many more gestures are recognized in this system than in the previous one. But the sense of interaction is more heightened, more smooth, and more natural.

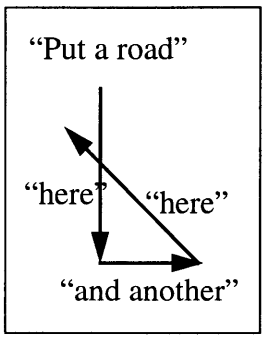


Figure 35. Precise Form of Gesture

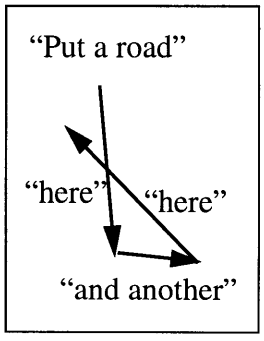


Figure 36. Less-precise Form of Gesture

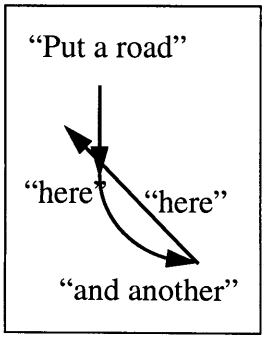


Figure 37. Analyzer's Output from Less-precise Form

The first generation of connected-speech recognizers also used a feature-based approach to analysis. The recognizers used hidden Markov models or similar algorithms to pick up on the proper features of the speech waveform and determine what words or phrases were being spoken. Modern research in speech technology appears to be moving away from this approach,[32] so it is reasonable to ask whether the use of features is an appropriate for gesture recognizers.

This is, unfortunately, an unanswerable question. Speech researchers spent ten or more years working on feature-based models and are only beginning to try new approaches. Perhaps when we have a similar body of research on gestural features we may find ourselves in a similar predicament. However, it is worth noting that speech recognizers are highly constrained and require very high levels of accuracy (well over 95% correct) for general usefulness. Gesture recognition is a much fuzzier science and it is unlikely that in a coverbal, conversation-oriented environment the same stringent level of accuracy would be demanded of the gesture analyzer.

## 5.5 Conclusion

This thesis set out to demonstrate two significant advances over current gesture recognition technology:

- enabling users to make continuous unrestricted gestures in the context of an interaction with a computer, allowing a more clean, natural interface with gestures as part of a multi-modal presentation; and
- semantic separation of gestural analysis from specific devices and from specific requirements of the interpreter, allowing the analyzer to be connected to any application and gestural input to be treated as a data type in its own right.

Both these advances have been proved out. A strong experimental grounding has been established for the feature set used and a limited demonstration system has embodied the ideas in practical, usable form.

It remains an open issue how frequently the sort of continuous gestures analyzed by this system will be used in arbitrary computer systems. As noted in Section 2.5.2 "Directive Interfaces", our style of system building is relatively distinct from the majority of applications being constructed today. It is unclear whether this approach could be useful in a non-directive style of interface. Directive interfaces are storytelling by nature; however, it is unclear how the narrative gestures involved in enacting a story will translate to the conversational realm of full multi-modal interaction.

It is also important to note that even as a data type gesture does not have a place in every application. Gestures convey important information about spatio-temporal relationships: placement, timing, and motion. If this kind of information is not important to a computer

32. I am again grateful to Barry Arons for discussion on this topic.

system, then gesture is probably inappropriate. This thesis does not advocate any attempt to shoehorn non-gestural interaction into a gestural mode. Natural interaction is key. To a large extent, the naturalness of the interaction and the amount of continuous gesturing done by users is affected by system response time. If the system responds rapidly enough, people are encouraged to make their gestures more fluid. If it responds too slowly, people are forced to wait and make their gestures correspondingly slower.

The ideas of this thesis and of the demonstration are consonant with the majority of the literature in psycholinguistics; however, it continues to be an unique application within the computer interface community. There is nothing magical or mysterious in the approach; this thesis exposes the important ideas of the theory and the relevant details of implementation. This should serve as a strong basis for future research and implementation.

# References


Badler, N. I., Phillips, C. B, and Webber, B. L. (1993) *Simulating Humans: Computer Graphics Animation and Control.* Oxford University Press, New York.

Bolt, R. A. (1980) "Put-That-There: Voice and Gesture at the Graphics Interface." *Proceedings of SIGGRAPH '80.* ACM Press, New York.

Bolt, R. A. (1984) *The Human Interface.* Van Nostrand Reinhold, New York.

Bolt, R. A. and Herranz, E. J. (1992) "Two-handed Gesture with Speech in Multi-Modal Natural Dialogue." *Proceedings of UIST '92.* ACM Press, New York.

Bos, Edwin (1992). Some Virtues and Limitations of Action Inferring Interfaces *Proceedings of UIST 92.* ACM Press, New York.

Butterworth, B. and Beattie, G. (1978) "Gesture and Silence as Indicators of Planning in Speech." *Recent Advances in the Psychology of Language*, Campbell & Smith (eds.) Plenum Press, New York.

Butterworth, J., Davidson, A., Hench, S. and Olano, T. M. (1992) "3DM: A Three Dimensional Modeler Using a Head-Mounted Display." *Proceedings 1992 Symposium on Interactive 3D Graphics.* ACM Press, New York.

Cohen, P. R., Sullivan, J. W., et. al. (1989) "Synergistic Use of Direct Manipulation and Natural Language." *CHI '89 Proceedings.* ACM Press, New York.

Codella, Jalil, Koved, Lewis, Ling, Lipscomb, Rabenhorst, Wang, Sweeney and Turk (1992) "Interactive Simulation in a Multi-Person Virtual World." *CHI '92 Proceedings*, ACM Press, New York.

Cumming, G. D. (1978) "Eye Movements and Visual Perception," *Handbook of Perception, Vol IX, Perceptual Processing*, E.C. Carterette & M. P. Friedman (eds.) Academic Press, New York.

Darrell, T. J. and Pentland, A. P. (1993) "Recognition of Space-Time Gestures using a Distributed Representation," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (also available as *MIT Media Laboratory Vision and Modeling Technical Report #197*).

Efron, D. (1941) *Gesture and Environments.* King's Crown Press, Morningside Heights, New York.

Fisher, S., McGreevy, M., Humphries, J. and Robinett, W. (1986) "Virtual Environment Display System." *Proceedings 1986 Workshop on Interactive 3D Graphics.* ACM Press, New York.

Graham, J. A. and Argyle, M. (1975) "A Cross-Cultural Study of the Communication of Extra-verbal Meaning by Gestures." *International Journal of Psychology, 10.*

Graham, J. A. and Heywood, S. (1976) "The Effects of Elimination of Hand Gestures and of Verbal Codability on Speech Performance." *European Journal of Social Psychology, 5.*

Grand, S., Marcos, L. R., Freedman, N., and Barroso, F. (1977) "Relation of Psychopathology and Bilingualism to Kinesic Aspects of Interview Behavior in Schizophrenia." *Journal of Abnormal Psychology*, 86(5).

Hauptman, A. G. (1989) "Speech and Gestures for Graphic Image Manipulation." *CHI '89 Proceedings*, ACM Press, New York.

Herranz, E. J. (1992) Giving Directions to Computers via Two-handed Gesture, Speech and Gaze. S.M. Thesis, MIT Media Arts and Sciences Section.

Hill, W., Wroblewksi, D., McCandless, T. and Cohen, R. (1992) "Architectural Qualities and Principles for multi-modal and Multimedia Interfaces." *Multimedia Interface Design*, Blattner, M. and Dannenberg, R. B. (eds.) ACM Press, New York.

Hollan, Rich, Hill, Wroblewski, Wilner, Wittenberg, Grudin (1988) "An Introduction to HITS: Human Interface Tool Suite." MCC Technical Report ACA-HI-406-88, Austin, TX.

Hutchins, E., Hollan, J., and Norman, D. (1986) "Direct Manipulation Interfaces." *User Centered Systems Design*, Norman and Drayer (eds.) Lawrence Erlbaum Associates, Hillsdale, N.J.

Kendon, A. (1980) "Gesticulations and Speech: Two Aspects of the Process of Utterance." *The Relation between Verbal and Non-verbal Communication*, M. R. Key (ed.) Mouton, The Hague.

Kendon, A. (1986) "Current Issues In The Study Of Gesture." *The Biological Foundations of Gestures: Motor and Semiotic Aspects*. Nespoulous, Perron, and Lecours (eds.) Lawrence Erlbaum Associates, Hillsday, N.J.

Klima, E. and Bellugi, U. (1979) *Signs of Language*. Harvard University Press, Cambridge, MA.

Koons, D. B., Sparrell, C. J., and Thorisson, K. R. (1993) "Integrating Simultaneous Input from Speech, Gaze, and Hand Gestures." *Intelligent Multi-Media Interfaces*, M. Maybury (ed.) AAAI Press, Menlo Park, CA.

Koons, D. B. (1994) "Capturing and Interpreting Multi-Modal Descriptions with Multiple Representation." working paper for *AAAI Spring Symposium*, Stanford, CA. (available from the author at dbk@media.mit.edu)

Koons, D. B. and Thorisson, K. R. (1993) "Estimating Direction of Gaze in Multi-Modal Context." *Proceedings of 3Cyberconf — the Third International Conference on Cyberspace*, Austin, TX.

Kosko, B. and Isaka, S. (1993) "Fuzzy Logic," *Scientific American*, July 1993.

Kramer, J. and Leifer, L. (1989) "The Talking Glove: An Expressive and Receptive "Verbal" Communication Aid for the Deaf, Deaf-Blind, and Nonvocal." Department of Electrical Engineering, Stanford University, Stanford, CA.

Kurtenbach, G. and Buxton W. (1991) "Issues in Combining Marking and Direct-Manipulation Techniques." Proceedings of UIST'91. ACM Press, New York.

Lakoff, G. and Johnson, M. (1980) *Metaphors We Live By*. University of Chicago Press, Chicago.

Laurel, B. (1991) *Computers as Theater*. Addison-Wesley, New York.

Lewis, J. B., Koved, L., and Ling, D. (1991) "Dialog Structures for Virtual Worlds," *CHI '91 Proceedings*, ACM Press, New York.

MacKenzie, C. L. (1994) *The Grasping Hand*. Elsevier Science, North Holland, Amsterdam.

Maes, P. (1993) "ALIVE: An Artificial Life Interactive Video Environment," *Computer Graphics Visual Processing*, ACM Press, New York.

Marcos, L. R. (1979) "Hand Movements and Nondominant Fluency in Bilinguals." *Perceptual and Motor Skills*, 48.

McNeill, D. and Levy, E. (1982) "Conceptual Representations in Language Activity and Gesture." *Speech, Place, and Action*, Jarvella and Klein (eds.) John Wiley & Sons Ltd.

McNeill, D. (1992) *Hand and Mind: What Gestures Reveal about Thought.* University of Chicago Press, Chicago.

Miller, G. (1982) "Some Problems in the Theory of Demonstrative Reference." *Speech, Place, and Action,* Jarvella and Klein (eds.) John Wiley & Sons Ltd.

Murakami, K. and Taguchi, H. (1991) "Gesture Recognition Using Recurrent Neural Networks." *CHI '91 Conference Proceedings.* ACM Press, New York.

Neal, J.G. and Shapiro, S.C. (1991) "Intelligent Multi-Media Interface Technology." *Intelligent User Interfaces,* J.W. Sullivan and S.W. Tyler (eds.) ACM Press, New York.

Nespoulous, J. and Lecours, A. R. (1986) "Gestures: Nature and Function." *Biological Foundations of Gestures: Motor and Semiotic Aspects,* Nespoulous, Perron, and Lecours (eds.) Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Norman, D. (1988) *The Design of Everyday Things,* Basic Books, New York.

Poizner, H., Klima, E. S. and Bellugi, U. (1987) *What the Hands Reveal about the Brain.* MIT Press, Cambridge, MA.

Rimé, B. and Schiaratura, L. (1991) "Gesture and Speech." *Fundamentals of Nonverbal Behavior,* Feldman & Rimé (eds.) Press Syndicate of the University of Cambridge, New York.

Rudnicky, A. and Hauptmann, A. (1992) "multi-modal Interaction in Speech Systems." Multimedia Interface Design, Blattner, M. and Dannenberg, R. B. (eds.) ACM Press, New York.

Rubine, D. (1991) The Automatic Recognition of Gestures. Ph.D. Dissertation, Carnegie-Mellon University.

Sims, K. (1987) Locomotion of Jointed Figures over Complex Terrain, S.M. Thesis, MIT Media Arts and Sciences Section.

Sparrell, C. J. (1993) Coverbal Iconic Gestures in Human-Computer Interaction. S.M. Thesis, MIT Media Arts and Sciences Section.

Starner, T. E. (1993) Unencumbered Virtual Environments, S.M. Thesis Proposal, MIT Media Arts and Sciences Section.

Sturman, D. J. (1992) Whole-hand Input. Ph.D. Dissertation, MIT Media Arts and Sciences Section.

Thorisson, K. R., Koons, D. B. and Bolt, R. A. (1992) "Multi-Modal Natural Dialogue." *CHI '92 Proceedings.* ACM Press, New York.

Thorisson, K. R. (1993) "Dialogue Control in Social Interface Agents." *INTERCHI Adjunct Proceedings '93.* ACM Press, New York.

Väänänen, K. & Böhm, K. (1993) "Gesture-Driven Interaction as a Human Factor in Virtual Environments — An Approach with Neural Networks." *Virtual Reality Systems,* Gigante, M.A. & Jones, H. (eds.), Academic Press, Ltd., London, UK.

Weimer, D. and Ganapathy, S. K. (1989) "A Synthetic Visual Environment with Hand Gesturing and Voice Input." *CHI '89 Proceedings.* ACM Press, New York.

Weimer, D. and Ganapathy, S. K. (1992) "Interaction Techniques Using Hand Tracking and Speech Recognition." *Multimedia Interface Design,* Blattner, M. and Dannenberg, R. B. (eds.) ACM Press, New York.

Whittaker, S. and Walker, M. A. (1991) "Toward a Theory of Multi-Modal Interaction." *AAAI '91 Workshop Notes*. AAAI Press, Menlo Park, CA.

Zadeh, L. (1982) "Fuzzy Probabilities and Their Role in Decision Analysis." *Proceedings of the Fourth MIT/ ONR Workshop on Distributed Information Systems*.

Zimmerman, Lanier, Blanchard, Bryson and Harvill (1987) "A Hand Gesture Interface Device" *CHI+GI '87 Proceedings*. ACM Press, New York.

# Appendix A — Experiment Information

Subjects were initially instructed by means of a script, in order to make sure that they all received the same instructions. The script was:

> *"You are participating in an experiment to help understand how people describe episodes such as scenes from a play or movie. The way we will do this is by asking you to watch some clips from a movie, as explained below, then describing one of the clips in such a way that a person who has not seen the clips or the movie will be able to tell which clip you mean.*
>
> *"You will be shown ten short clips from the film 'Casablanca'*[33] *Each clip is under 45 seconds long. You will watch all tend segments. Then I will give you the number of a particular segment and you will be asked to describe that segment. You will be videotaped giving that description; this videotape will be viewed later by the experimenter who will attempt to decide solely from your description which segment you are describing.*
>
> *"You are encouraged to be as complete and detailed in your descriptions as you can. You should remember that there is no right or wrong way to make descriptions and the purpose of this experiment is to learn what people naturally do when they make descriptions. The most important thing is to give a description which will uniquely distinguish your target clip from all the rest."*

After listening to these instructions, subjects read and signed an Informed Consent form which is reproduced below:

INFORMED CONSENT FORM

Title: Observation of multi-modal Reference

Participation in this study is voluntary, and you are free to withdraw your consent and to discontinue participation in the study at any time without prejudice. If you choose to withdraw you will be paid for the time you have spent.

The purpose of the study is to gather sample observations on how people spontaneously refer to items on a display, and describe manipulations to be done on those items. In the study, you will be shown sets of images or events on a display. These items may include animations or film clips. You will be asked to describe the actions in these items. There are no "right" or "wrong" ways for you to respond.

---

33. The names of the movies were changed in all forms between runs of the experiment.

You will be paid five (5) dollars for each half hour, or fraction thereof, of active participation in the study. Total participation time will ordinarily not be more than one hour and typically will not be more than one half hour.

Consent to be videotaped:
In order not to interrupt the course of the experiment, your responses will be videotaped for later analysis by the Principal Investigator of this study, Dr. Richard Bolt, and his graduate research assistants (their names to be furnished to you at your request). Only these researchers will have access to the tapes for post-session viewing. These tapes will not be used in any publication, made public, or circulated in any way. At the conclusion of this study, all tapes showing you participating in this study will be erased or destroyed.

In the unlikely event of physical injury resulting from participating in this research, I understand that medical treatment will be available from the M.I.T. Medical Department, including first aid, emergency treatment and follow-up care as needed, and that my insurance carrier may be billed for the cost of such treatment. However, no compensation can be provided for medical care apart from the foregoing. I further understand that making such medical treatment available, or providing it, does not imply that such injury is the Investigator's fault. I also understand that by my participation in this study I am not waiving any of my legal rights. Further information may be obtained by calling the Institute's Insurance and Legal Affairs Office at 253-2822.

I understand that I may also contact the Chairman of the Committee on the Use of Humans as Experimental Subjects, M.I.T. 253-6787, if I feel I have been treated unfairly as a subject.

**I have read and understand the above and agree to participate in this research effort. I consent to be videotaped as part of this study.**

NAME: DATE:

---

After reading and signing the consent form, subjects filled out a brief questionnaire, which is reproduced below. As noted in Section 3.2 "Experiment Results" no correlation was found between externally-distinguishable subject features (such as gender) and performance in the experiment. In the questionnaire, subjects were asked if they had previously seen the film used in the experiment. Both films — THE TERMINATOR and CASABLANCA — were selected in part for their popularity and likelihood that subjects would be at least somewhat familiar with the movies. In reviewing the transcripts no significant correlations appeared between subjects' knowledge of the movies and their gestures.

One indirect correlation that was noticed came through the medium of speech. Subjects with knowledge of the movies were more precise in their descriptions (using character's names, for example). This led them to make fewer modalizing gestures to indicate their uncertainty. However, more precise verbal descriptions did not lead to more precision in gesturing. None of the subjects who actually participated in the study labeled themselves as having experience describing scenes for any medium. An interesting research question remaining is what effect (if any) previous descriptive experience would have on the gestures people make.

Questionnaire for Subjects:

Title: Description of EpisodesSubject #_____
Name:
Age:
Gender: M F
Education level: . 1) Finished high school 2) Some college (undergraduate)
3) Undergraduate degree 4) Some college (graduate)
5) Graduate degree

Have you seen the movie "Casablanca" before today? Y N

If yes, how recently was the previous time you saw it?

Do you have any experience describing scenes for radio, television, film, or theater?

If yes, please briefly describe your experience:

Table 6 below shows the list of clips used in the experiment. As noted in "Experiment" on page 39, the scenes are all short and selected to feature a recurring set of characters and similar actions in several different scenes to encourage subjects to make more elaborate and involved descriptions.

In each set of scenes there are two "distractor" entries. These clips were used to help minimize subject complacency by occasionally switching subject matter. However, subjects were never asked to describe the distractor scenes because of their deliberate distinctness. In the table, these distractor scenes have an asterisk (*) after their numbers.

**Table 6: Experiment Clip List**

| Clip Description |
|---|
| 1: Rick is at his desk watching Peter Lorre drinking. |
| 2* Sam is at the piano; Rick walks by and hides the papers in the piano then goes to get a drink. |
| 3: Rick and the Captain are at a table outside the cafe. The plane to Lisbon flies overhead. |
| 4: Rick and the Captain are in Rick's office talking. |
| 5* Ms. Elton and Sam are at the piano. She asks him to play "As Time Goes By." |
| 6: Rick and Ms. Elton are at a table. Rick is drinking. |
| 7: Rick and Ms. Elton are at a table. Rick is drinking. She leaves. |
| 8: Rick and the Captain sit at a table inside the cafe. |
| 9: Rick, Captain and Ms. Elton are at the airport. She is preparing to leave. |
| 10: Rick and the Captain are at the airport. The plane carrying Ms. Elton flies away. |

**Table 6: Experiment Clip List**

| Clip Description |
|---|
| 1. Rees plants bomb in Terminator's tanker truck while Terminator is chasing Sarah |
| 2. Terminator chases Rees & Sarah in cop car, ends up crashing into wall. |
| 3* Police captain tells Lieutenant to call Sarah. |
| 4. Terminator chases Rees & Sarah on motorcycle, ends up crashing cycle and their truck flips. |
| 5. Rees shoots up Terminator in nightclub when Terminator tries to assassinate Sarah. |
| 6. Rees & Sarah discuss bomb-making over grocery bags of supplies. |
| 7. Police captain and psychologist tell Sarah Rees is crazy and Terminator was wearing vest. |
| 8. Sarah awakes in Rees' arms, they walk out of cave. |
| 9* Terminator uses scalpel to pluck out damaged eyeball. |
| 10. Rees battles a futuristic tank, throws bomb, escapes in car being chased by tank. |

# Appendix B — Interfaces to the Gesture Analyzer

There are two interfaces to the gesture analyzer, one from the lower end which supplies the data — the body model (see "The Body Model" on page 59) — and one from the higher end which accepts the data — the interpreter (see "The Interpreter" on page 61).

The body model and the analyzer share a common implementation language, so the interface is in the form of a C++ header file which defines the data byte order. It also defines a number of data types and manipulation routines; this allows the analyzer to maintain a higher level of data abstraction than would otherwise be possible.

The interpreter operates in a different language, so all data is transferred as text strings. The interface definition, therefore, is a text file describing the information to be transferred. As detailed in "Implementation Architecture" on page 56, the information is formatted as a Lisp structure which can be directly evaluated.

### Body-Model Interface

```
// This module defines the body model interface
#ifndef BM_IF
#define BM_IF

extern "C" {
#include <sys/types.h>
#include <netinet/in.h>
}

// Host on which the body model runs and initial connection port #
#define BM_HOST_ADDR "18.85.0.211"
#define BM_PORT 5194

// Defines for expressing interest in possible data values.
// If new blocks of data are added to the model, the block comment below
// must be expanded to define the values in order, and the defines below
// must be updated.
#define LA  0x00000001     // left arm (shoulder->wrist)
#define LH  0x00000002     // left hand (palm->fingers)
#define RA  0x00000004     // right arm
#define RH  0x00000008     // right hand
#define EYE 0x00000010     // eye
#define HED 0x00000020     // head
#define TOR 0x00000040     // torso

/* Data values in order within block order defined above (note count
defines below):
   Arm data is shoulder, elbow, wrist
         shoulder is X, Y, Z (of shoulder point), Elevation, Abduction,
Twist
         elbow is X, Y, Z, Flex, Twist
         Wrist is X, Y, Z, Pitch, Yaw
```

```
    X, Y, Z are in global coordinates, relative to imaginary origin
where center of cube would touch floor
    X is left/right with 0 at floor under left corner of screen
    Y is toward user, with 0 at screen face
    Z is up, with 0 at floor

    All angles in degrees

    Shoulder elevation 0 with arms relaxed at side, up is +
    Shoulder abduction 0 with upper arm perpendicular to shoulder, away
        from body is +, across body is -
    Shoulder twist 0 with arms hanging at sides, inside of elbow facing
        torso, inside of elbow forward is +
    Elbow flex 0 when arm straight, toward shoulder is +
    Elbow twist 0 when wrist axis is 90 degrees from elbow, thumb out is
        +, thumb in is -
    Wrist pitch 0 when back of hand level with forearm, hand up is +,
        hand down is -
    Wrist yaw 0 when palm aligned with forearm, fingers toward thumb
        is -, away from thumb is +

  Hand data is palm, thumb, fore, middle, ring, pinkie fingers, abductors
        palm is X, Y, Z (location), X, Y, Z (normal), X, Y, Z
        (approach), palm arch
        approach and normal vectors define orientation of hand in
                        global space
          palm arch 0 with palm flat on table, closed is +
        thumb is mcp, ip
          mcp 0 with thumb 90 degrees from palm
          ip 0 with thumb extended
        forefinger is mcp, pip
          mcp 0 with finger extended
          pip 0 with finger extended
        middle finger is mcp, pip
          mcp 0 with finger extended
          pip 0 with finger extended
        ring finger is mcp, pip
          mcp 0 with finger extended
          pip 0 with finger extended
        pinkie finger is mcp, pip
          mcp 0 with finger extended
          pip 0 with finger extended
        abductors are all 0 with fingers next to each other (thumb next
                        to palm)
          thumb abductor, middle-index abductor, ring-middle abductor,
          pinkie-ring abductor

  Eye is X, Y, Z, normal-roll, normal-pitch, normal-yaw
        X, Y, Z are in global coordinates, give center of eye
        normal-roll, normal-pitch, normal-yaw are as for hand, with 0
        at parallel to floor, normal to screen (X axis), roll is always
        0 (eye does not truly roll)

  Head is X, Y, Z, elevation, abduction, twist
```

```
        X, Y, Z are in global coordinates, give top of crown center
        Elevation 0 with spine straight, up/back is +, down is -
        Abduction 0 with head in line with axis of spine, right is +
        Twist 0 with nose in line with bellybutton, right is +

  Torso is X, Y, Z, normal-roll, normal-pitch, normal-yaw
        X, Y, Z are in global coordinates, give center of spine at
                        shoulder height
        normal is direction of spine, roll, pitch, yaw as hand above
*/

// These defines count the number of values in each block of data; they
// are used here to define the largest possible array for fast buffer
// manipulations.  They are also used in body model clients for defining
// smaller buffers and controlling loops.
#define LA_DATAVALS 16
#define LH_DATAVALS 24
#define RA_DATAVALS 16
#define RH_DATAVALS 24
#define EYE_DATAVALS 6
#define HED_DATAVALS 6
#define TOR_DATAVALS 6

typedef short BmData;       // The BodyModel Data type.  Joint values are
                            // 0..360 or -180..180.  Positions are in
                            // centimeters
typedef unsigned long Timestamp;// Time since start-up of BM

#define MINTIME 0x00000000 // BM time starts at 1 so MINTIME is
                            // always less than a time sent by
                            // the BM
#define MAXTIME 0xffffffff

// These routines are for modularity; they are defines for speed
#define BmDataton(x) htons((x))
#define ntoBmData(x) ntohs((x))
#define Timeton(x) htonl((x))
#define ntoTime(x) ntohl((x))

typedef BmData
BmBuf[LA_DATAVALS+LH_DATAVALS+RA_DATAVALS+RH_DATAVALS+EYE_DATAVALS+HED_
DATAVALS+TOR_DATAVALS];     // Buffer to transfer body model data

#endif   // BM_IF
```

## Interpreter Interface

This is a text file which specifies the interface between the analyzer and the interpreter.

1. Data Format
Frames are sent to the interpreter as Lisp expressions which can be eval'ed directly.

If there is no useful information for one side, nothing will be output for that side. If both sides have useful information, the left side is output first for a give time span.

Static and orientation frames are output as points. Static frames have one vector pair (normal & approach). Orientation frames have two vector pairs -- one for beginning and one for end

Movements are output as lines (with start and end points only) or arcs (with a center point, a radius, start/end angles in degrees, and a pair of orientation vectors.

General notes:
- triples are output as lists of three values (x y z)
- pairs are output as lists of two values (one two)
  [note that either value may itself be a triple]

In the following BNF description, all-caps are terminals, lowercase are nonterminals. The vertical bar (|) separates options, the colon (:) shows nonterminal expansion. All other punctuation (quotes, parentheses, semicolons) is output as part of the buffer stream. n_X, n_Y, n_Z refer to the x, y and z values of a point or vector.

```
buf:  '( info ) ;
   |  '( info ) buf

info: hdr pos orient config

hdr: side timepair coord

side: L | R

timepair: (start_time stop_time)

coord: T | NIL

pos: point
   | line
   | arc

point: (POINT apoint)

line: (LINE (start_point stop_point))

start_point: apoint
stop_point: apoint

arc: (ARC center_point radius ( start_ang stop_ang )
         ( arc_plane_normal arc_plane_approach )
         ( start_point stop_point ) )

center_point: apoint
radius: NUMBER
```

```
start_ang: NUMBER
stop_ang: NUMBER
arc_plane_normal: apoint
arc_plane_approach: apoint
start_point: apoint
stop_point: apoint

apoint: (P_X P_Y P_Z)

orient: ( static_orient )
      | ( moving_orient )

static_orient: ( hand_normal hand_orientation )

hand_normal: apoint
hand_orientation: apoint

moving_orient: ( start_orient stop_orient )

start_orient: static_orient
stop_orient: static_orient

config: ( shape thumb_position finger_spread extended_fingers )

shape: NO_SHAPE
     | FLAT
     | CURLED
     | CUPPED
     | CLOSED
     | FIST
     | PINCHED

thumb_position: NO_POS
              | OUT
              | BESIDE
              | OVER

finger_spread: NO_SPREAD
             | TOGETHER
             | SPREAD
             | APART

extended_fingers: NUMBER
```

[To prevent confusion, the numbers correspond to unique bit values: THUMB=0x01, INDEX=0x02, MIDDLE=0x04, RING=0x08, PINKIE=0x10. Thus, having the thumb and index fingers extended gives the number 3; having the index and middle fingers extended gives the number 6, etc.]

2. Interaction Protocol

The analyzer holds data in an internal buffer until the interpreter asks for data. When asked, the analyzer will dump out all data not yet sent.

Communication is via TCP socket opened at initialization time. The interpreter will send a SIGIO signal when it wants to receive data.

Information is delivered in a text buffer of length 4096. If the data to be delivered exceeds that length, multiple instances of the buffer are written to the socket. Arguments are separated by spaces. The analyzer will ensure that complete, space-terminated, arguments are sent, which may result in buffers of less than maximum size being sent.

# Index