9. Acknowledgements

The authors would like to thank Dave Peloso of GasTops Ltd. and Tim Taylor of Phalanx Research who were contracted to implement the user interface and the generalized knowledge browser respectively.

Chapter 11

Integrating Simultaneous Input from Speech, Gaze, and Hand Gestures

David B. Koons, Carlton J. Sparrell, and Kristinn R. Thorisson

Abstract

The focus of this chapter is the integration of information from speech, gestures, and gaze at the computer interface. We describe two prototype systems that accept simultaneous speech, gestural and eye movement input from a user. The three modes are processed to a common frame-based encoding and interpreted together to resolve references to objects in the map. In the first prototype, a user can interact with a simple two-dimensional map. The computer responds in synthesized speech and by manipulating the map display. The second system uses a three-dimensional blocks world and demonstrates a more flexible interpretation strategy for handling full-hand gestures. Speech-related hand movements are processed to an intermediate level of representation without automatically assigning a deictic or symbolic meaning. Interpretation occurs later and takes advantage of information from the other modes and from the context.

1. Introduction

With increasing computer and robot intelligence, it is becoming more desirable to *communicate* with machines rather than *operating* them. We have at our disposal a wealth of techniques to communicate our thoughts and intentions. To get ideas across quickly, our communication system relies on a very efficient mix of spatial and semantic knowledge. We can switch instantly between various modes for communicating the same ideas: speech, hand gestures, facial gestures, intonation, etc. Together, these features significantly increase the "bandwidth" between two communicating parties.

One of the problems encountered when interpreting simultaneous input from multiple modes is the timing of events. With a free mix of input modes, actions are not coordinated according to a script or a set sequence. This puts a higher burden on the interpretation methods used. A second problem is the level of abstraction. If the signals from the disjoint modes are separate pieces of a particular message, the question becomes how far we should process the given "evidence" in each mode before trying to integrate it with the information from the other modes. A related problem is *how* to combine the information once it has been extracted. In this paper, we focus on integrating information from speech, gestures, and gaze at the computer interface. We have built two prototype systems that accept speech, gestural and eye movement input from a user. The first one allows interaction with a simple two-dimensional map (for an overview and hardware description, see [Thorisson et al. 1992]). Through a free mixture of speech, deictic gestures and glances the user can request information or give commands to modify the contents of the map database. The second system extends the repertoire of free gestures to include iconic and pantomimic gestures and replaces the two-dimensional graphics with a three-dimensional blocks world.

2. Related Research

2.1. Multi-Modal Interfaces

The most widespread example of a computer interface that offers multiple input channels is the combination of keyboard and mouse found on most modern workstations. These systems offer the user a choice of modes for many tasks: entering text is best accomplished with the keyboard; moving the cursor or other objects around on the screen is easily carried out with the mouse or other pointing device. Interface designers must constrain the possible actions that a user may carry out and develop an unambiguous mapping between the context of an action and its interpretation. In addition to a highly-constrained interaction, these interfaces do not allow the use of the multiple input channels in parallel (except for a few cases such as the click-drag).

"Put That There" [Bolt 1980] used speech-recognition and a three-dimensional sensing device to simultaneously gather input from a user's speech and the location of a cursor on a wall-sized display. CUBRICON, a more recent multi-modal interface prototype, provides for simultaneous speech, keyboard and mouse input [Neal and Shapiro 1991]. Although these are important contributions to the development of multi-modal interfaces, both systems reduce gestures to the location of a simple two-dimensional cursor. Hauptman [1989] conducted a "Wizard of Oz" study (a person monitors the user's actions and translates them into computer commands) and found that, when given the task of manipulating objects displayed on a computer display, there is an advantage in allowing a person to communicate using a free mixture of speech and gestures. However, the study also showed that two-dimensional input devices severely restricted the types of gestures that could be made by the person.

2.2. Gestures

Human hands are powerful tools for directly interacting with the environment and exerting influence on it. This may be one of the reasons why research on hand gestures (both in three-dimensional interactive graphics and computer generated environments as well as telerobotics) has focused on *tool-level* manipulation (e.g., [Butterworth et al. 1992; Sheridan 1992; Sturman 1992; Weimer and Ganapathy 1989; Fisher et al. 1986]). In these interfaces, gestures either imitate real-world actions (like grasping and throwing) or are purely symbolic, with a single and complete meaning attached to a pre-defined motion, posture, or combination of the two. While tool-level interfaces can be very efficient and sometimes intuitive, they often become unwieldy when functionality is hidden under layers of hierarchical modes and menus.

Another characteristic of these interfaces is their limitation to a small set of gestures [Tyler et al. 1991b; Wahlster 1991] that the user sometimes has to learn, and even train for, to be able to apply. Wahlster [1991] has a good example of such a system using deictic gestures. In his interface the user selects the desired type of gesture from a menu of icons; the interpretation of the subsequent deictic gesture (a mouse click in a chosen region of the screen) is based on the type of icon selected. While the interpretation is context dependent, the interface still forces the user to learn specific rules for operating the computer.

In addition to tool-level manipulation our hands play a large role in communication because of their natural link to the spatio-temporal world in which we live [Rimé and Sachiaratura 1991]. People use their hands to show three-dimensional relationships between objects and temporal sequences of events: a hand can in one instance take the role of an object and in the next, serve as a pointer to fictional constructs created in the gesture space. These types of gestures at the interface have received very little attention.

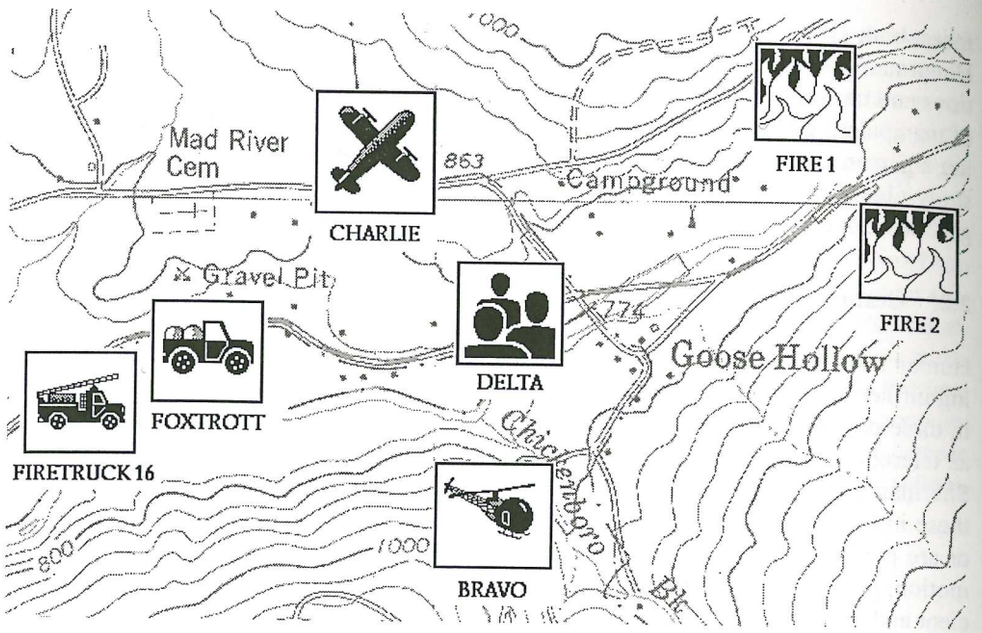

Figure 1. A section of the screen with icons representing helicopters, airplanes, trucks, fire crews and fire locations. The user can move, create and change these objects by referring to them in a free mixture of speech, gaze, and gestures.

2.3. Eyes

If an object of discussion is in someone's vicinity there is a natural tendency to glance in its direction, make gestures toward it, and to look directly at it, in coherence with the flow of the conversation [Kahneman 1973]. This feature of gaze can augment the interpretation of deictic references when the input from other modes is partial or segmented. Starker and Bolt [1990] describe an interface that bases its output on the data from a user's looking behavior. Whereas their system used only eyes as input, they made an effort to interpret the deictic behavior of the eye on more than one level, making the interface potentially more responsive to the user's looking. Jacob [1990] devised an eye tracking interface that allows a person to use gaze, or gaze and keyboard input in combination, to make selections on a computer screen. However, in this system, eye tracking plays a role similar to a mouse or other pointing device. In our system, the user is not required to use eyes as a pointer. We are specifically looking to incorporate eyes into the interaction process in a non-intrusive manner.

3. Prototype for Three Modes

To explore the issues related to multi-modal interpretation, we designed a prototype system that can gather input from speech, gestures and eye movements. The primary goal was to design a computer interface that could collect, process and interpret the different channels into a single integrated meaning.

A simple two-dimensional map is used as the subject of the interaction with the computer (Figure 1). Through a free mixture of speech, gestures and glances the user can request information or give commands to modify the contents of the map database. The prototype is composed of three major components: the input system, the map database, and the interpretation module (Figure 2).

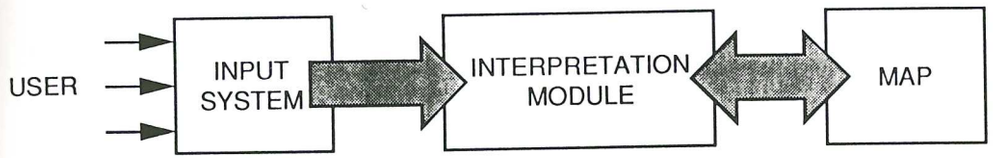

Figure 2. The interpretation module receives information from both the user's actions and the map database, which allows it to interpret the actions in the context of the map.

3.1. Input System

The user's speech is recognized using a PC-based discrete word recognition system. Hand and arm movements are sampled using full hand sensing hardware, and data on eye movements is collected using a corneal-reflection eye tracker. All three streams of data, the words from the speech recognizer, the position and posture of the hand and the

point of gaze, are collected on a central workstation (Figure 3). Each incoming data record is assigned a time stamp as it arrives on the host computer. This timing information is later used to realign data from the different sources.

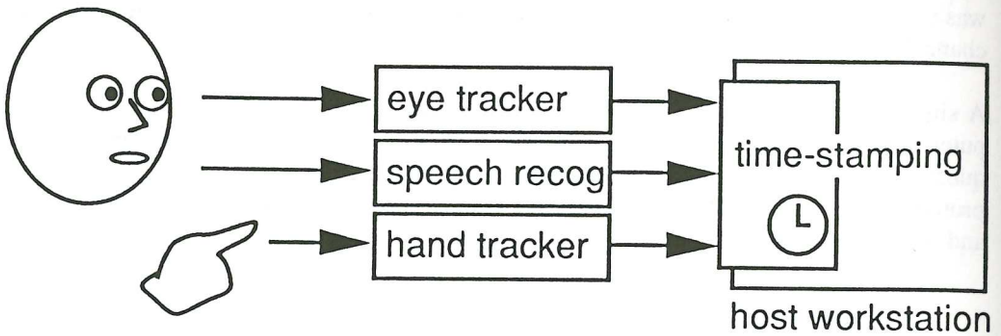

Figure 3. The data from speech, eye and hands are collected and sent to a real- time interface where they are time stamped. The time stamping allows for a later reconstruction of the input based on the exact time of occurrence.

3.2. Map Database

The map, displayed on the workstation display, serves as the shared subject for the human-computer interaction. The graphic presented is a simple two-dimensional color map and a number of colored icons representing objects. An object-oriented database manages the attributes and locations of the map objects. A command-language interface allows the interpretation module (and external simulation modules) to make queries to the database contents or to modify selected objects. When the system is initiated, the map database reads a stored configuration and displays the map for the user.

3.3. Interpretation Module

Based on the observation that a message is often composed of at least two very different kinds of information, the interpretation module includes two different representational systems that are interconnected: the first system is used to encode categorical information; the other is used to encode geometrical or spatial information. Information from the map database is used to build and maintain a knowledge base that spans the two representational systems. Map objects are represented both as nodes in a semantic network within the categorical system and as models in the spatial system. The interpretation module's task is to gather information from the input system and match the message to elements within the knowledge base.

The user input is processed in two major steps. First, the three input streams are parsed to produce a frame-like description of the structure of the incoming data. Second, the frames are interconnected and evaluated. Some frames can be encoded and evaluated in the categorical system where others find values in the spatial system. Together the expression guides the evaluation of the user's utterance. Once the expression is completely evaluated and all references have been resolved, the computer can then respond to the user's request. An example of this process is given in Section 3.6.

3.4. Parsing

Each mode has its own parser that takes advantage of the structure or syntax inherent in the corresponding data stream. The output of the three parsers is an expression in a common intermediate frame-based representation.

A parse tree is produced from the incoming words in the speech channel (Figure 4). As syntactic tokens are created and added to the parse tree, frames associated with those tokens are created and arranged into nested expressions. The timing information of individual words is carried up through the syntactic tokens and into the frames.

For this prototype, gestures and eye movements are treated in a simplified way and have only deictic interpretations. Posture and movement data from the hand-sensing hardware are processed to recognize postures and movements directed at the workstation display.

When such a movement is detected, a frame is created with the glove and time data. Data from the eye tracker is analyzed to detect characteristic features in the motion of the eyes: fixations, saccades and blinks. A frame is created for each fixation containing the associated tracking data and time stamp.

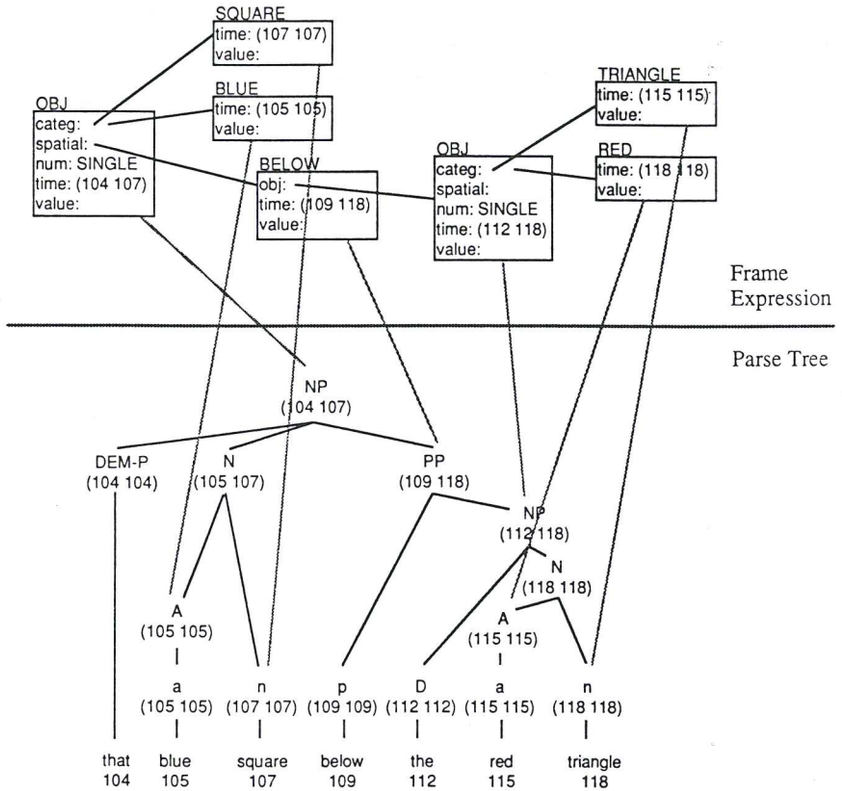

Figure 4. A parse tree is built from the incoming speech and then connected to the frame-based system. Here the parsed utterance is "... that blue square below the red triangle." Figure 5 shows frames produced from the other two modes.

3.5. Evaluation

Each frame produced during parsing has a corresponding "evaluation method" that controls the search for that frame's value within the knowledge base. Depending on the type of frame, values can range from *nodes* in the propositional system (representing attributes or individual objects) to *points* or *regions* in the spatial system. When an evaluation method is successful, the resulting value for the corresponding frame is now available to other evaluation methods. (Frames can serve as slot values in other frames, creating nested expressions similar to LISP functions.) If a problem arises in the evaluation of a frame, a "problem method" is started. A new subgoal is then created that attempts to find the missing information in other modes or by asking the user for additional information.

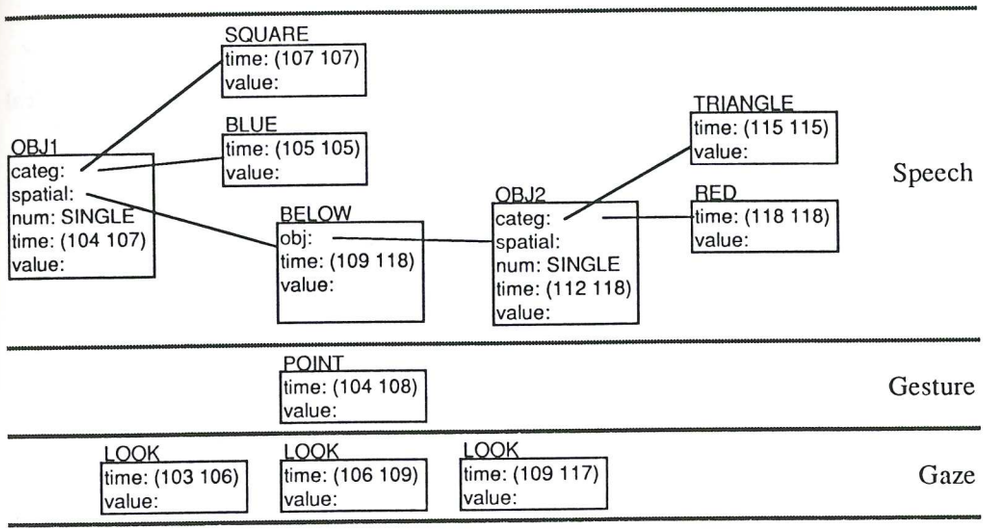

Figure 5. An idealized example of the frames produced from all three input modes during utterance "...that blue square below the red triangle"

3.6. Example Evaluation

Suppose the user is sitting in front of the display. He now says, "That blue square below the red triangle" while looking at the upper right quadrant of the screen and pointing in a similar direction. After this speech input is parsed, a nested expression of frames will be produced (Figure 5). Parsing the hand data and eye tracking data will produce additional, but at this time, disjoint frames. The evaluation methods attached to each of the frames begin to search for values for their frames in the representational systems. Because all gestures in this prototype are treated as deictic, an evaluation method attached to a gesture frame will produce a *point* in the spatial system; this *point* is the frame's value. The fixation frames are treated in a similar manner.

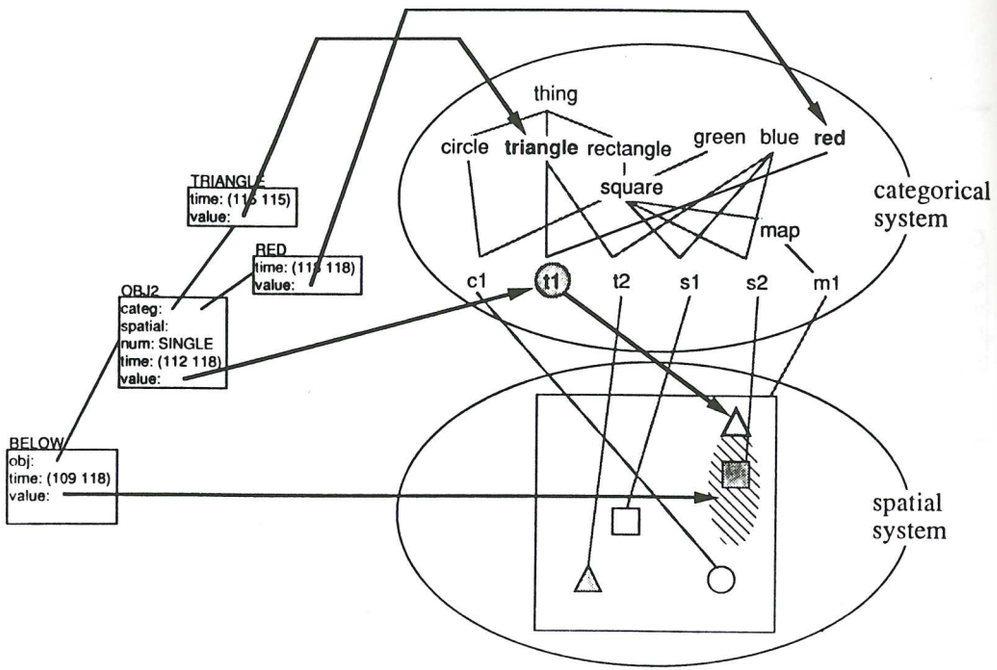

Figure 6. The expression associated with "below the red triangle" is evaluated by finding values for each frame. Frame values include object models and regions in the spatial system as well as nodes in the categorical system.

The evaluation method attached to the speech frame labeled OBJ2 (and associated with the speech input "the red triangle") in Figure 6 first attempts to use the propositional system to find an object that is both *red* and a *triangle.* This object, represented as a node in the propositional system (with links into its representation in the spatial system), is now available to the evaluation method attached to the BELOW frame. This method accepts the *red* triangle *t1* as an argument and shifts to the spatial system to produce a region that is in the proper spatial relation to the red triangle (below it). Meanwhile, the evaluation method attached to the OBJ1 frame (associated with the speech input "that blue square") attempts to find a single object in the propositional system but finds that there are multiple blue squares within the current map. A problem method now searches the other information sources and finds the frames in the gesture and eye modes that have an acceptable temporal relation to its frame (determined by temporal proximity). With the additional information from these deictic frames, and the BELOW frame from speech, the evaluation method for the OBJ1 frame can now use the spatial system to find the only blue square that is in the correct location on the map (Figure 7). Once the utterance has been successfully evaluated (all references have been resolved), the computer will react to the user's input. For this example, the result of evaluation is the square *s2* (for example, as the user's answer to an incomplete command). In the case of a query or a command, the interpretation module will send the appropriate commands to the map database and generate a simple statement that is sent to a speech synthesizer.

3.7. Discussion

This prototype system highlights many important points in attempting to interpret simultaneous multi-modal inputs. First and most obviously, this interface is a departure from current computer interface design. Unlike the interaction on a modern workstation, the user is able to use any combination of modes to communicate the request. For example, the user can choose to use speech alone with a request like "Delete all the blue squares." Or, speech can be reduced to "Move that to here" with gestures or glances filling in the necessary information. This opens up a highly flexible communication style for the user (and the interface designer).

A related point demonstrated in the prototype is that the interpretation of multi-modal input must be handled in a way that takes advantage of the interdependencies between the information supplied by each mode. While one mode may carry a significant portion of the information (usually speech), most messages cannot be interpreted without using the information from the other modes. These interdependencies require an interpretation

process that is able to build up a single meaning by using all the modes simultaneously in a process similar to constraint satisfaction.

A shortcoming of this first prototype is its oversimplification of gestures and eye movements. Contrary to the idea that interpretation should not be carried out in any one mode, gestures and fixations in the prototype were automatically assigned a deictic interpretation. This treatment ignores the rich and subtle communicative abilities of both gestures and eye movements in natural discourse.

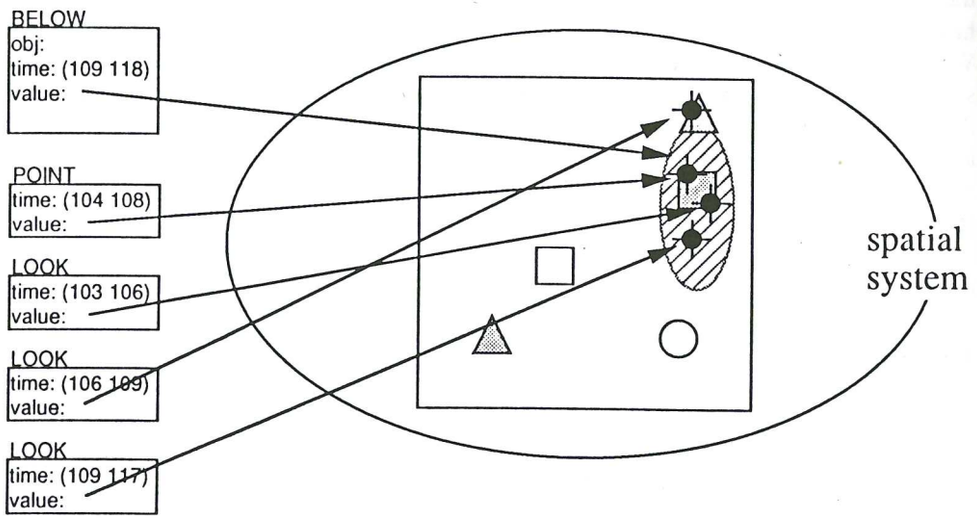

Figure 7. Spatial values for frames originating in the speech, gesture and eye tracking channels are compared in the spatial representation system.

4. Beyond Pointing

Gestures are most often integrated with our speech and other channels of communication. We fluidly switch context in the process of communicating a message, such that

two identical hand movements might represent different actions, objects or ideas even when performed in the space of one verbal phrase. What are the different ways that a speech-related movement might be interpreted? Several taxonomies have been proposed for categorizing gestures that occur with speech. The taxonomy proposed by Rimé and Schiaratura [1991], which is a revision of the Efron classification system, proposes the following gesture types:

Symbolic gestures can be translated directly to some verbal meaning (such as the "OK" *posture* made by touching the forefinger to the thumb and extending the other fingers). Gestures such as these are normally part of a culture and have come to represent a single unambiguous meaning within that culture.

Deictic gestures include pointing or motioning to direct the listener's attention to objects or events in the surrounding environment.

Iconic gestures are used by a speaker to display information about the shape of objects, spatial relations, and actions.

Pantomimic gestures usually involve the manipulation of some invisible object or tool in contact with the speaker's hand.

Of these possible interpretations of speech-related hand movements, only the symbolic gestures can be interpreted immediately (within a given cultural context). Deictic, iconic and pantomimic gestures usually cannot stand alone and must be interpreted with additional information from the other channels and/or the surrounding context.

4.1. Representational Level

In order to extract meaning from the streams of gestural data, an appropriate level of abstraction must be chosen. At the lowest level, we have the constant flow of raw data from the full-hand input device hardware. At the highest level of abstraction would be a pure symbolic language (such as American Sign Language) in which complete gestures are specifically categorized to have an exact meaning.

It has been previously stated that using the highest level of abstraction, as in a symbolic language, limits the flexibility of the hands. This method changes the hands into "tools" and restricts their communicative power, creating problems such as the "Midas Touch." We have tried to find an intermediate level of abstraction that refines and reduces the information from the raw data and faciltates interpretation in the broader context of information available from other sources.

4.2. Gesture Features

Two layers of abstraction are built before the final interpretation of the gestures (Figures 8 and 9). First, the hand data is classified into features of *posture, orientation,* and *motion* at each discrete sample point. Currently, the posture features for each finger are *straight, relaxed,* or *closed.* For orientation of the hand, we look at the direction of two vectors coming out of the hand. The first is a normal vector out of the palm (vector A in Figure 10). The second is a longitudinal vector indicating where the hand is pointing (vector B in Figure 10). The general direction of both vectors is quantized into the values *up, down, left, right, forward,* or *back* (relative to the person's trunk). Hand motion is currently only specified as *moving* or *stopped.* While relatively crude, these descriptive tags are useful in detecting important changes in the hand data.

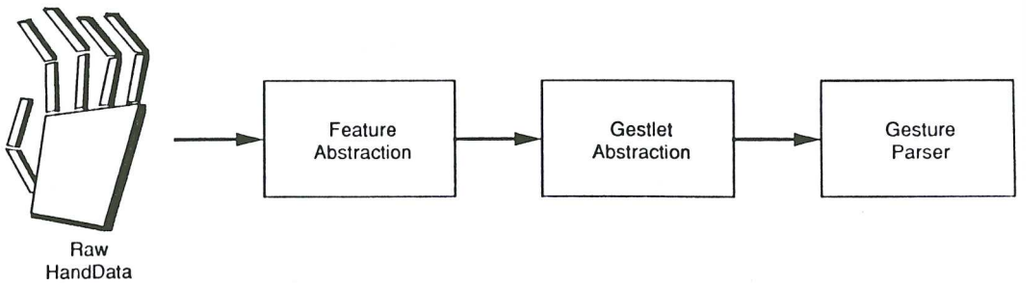

Figure 8. Raw hand data are processed successively on three separate representational levels.

The features are first used for data reduction: when a record is received with one or more of the values differing from the preceding record, it is extracted for further processing. These extracted records are passed to the next abstraction layer. Subsequent identical records are saved but not processed. (The unprocessed records are preserved for cases where the exact path is important, such as in the case of a person drawing out a detailed shape with their hand.)

4.3. Gestlets

A second layer of abstraction is created by collapsing the stream of features into structures similar to speech phrases (Figure 9). We refer to these structures as *gestlets.* The gestlets are pieces of gestures that have been formed by grouping portions of the feature stream together using certain rules. The most useful rule for this purpose has been to group all contiguous data sets where the hand is moving together with the preceding and following records when the hand is stopped.

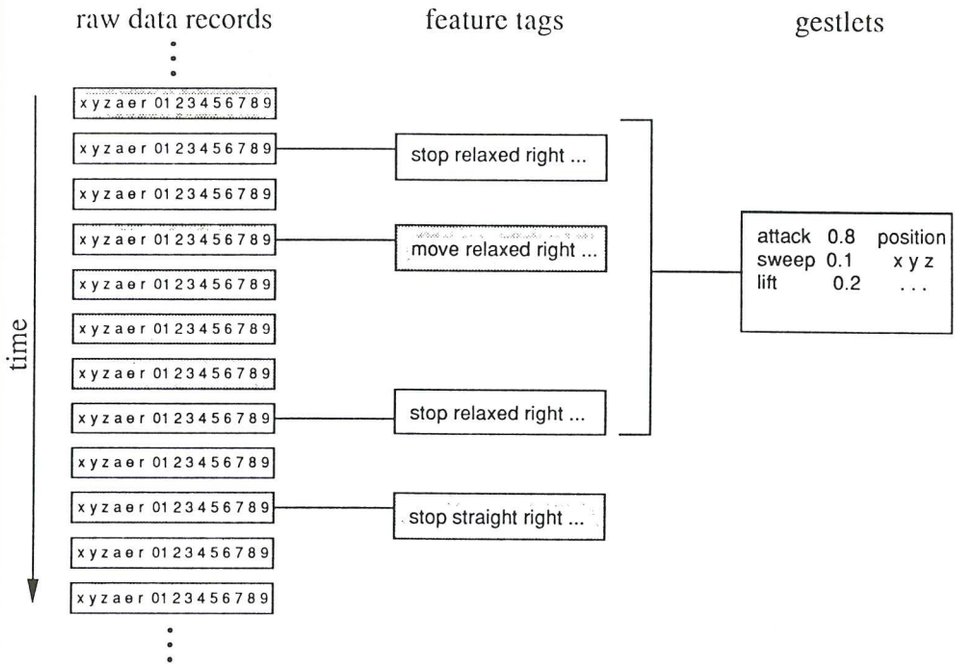

Figure 9. Parsing the raw hand data involves extracting features (feature tags) and combining these into meaningful units to produce gestlets.

The resulting stream of gestlets is buffered. If evidence in the speech channel suggests that important informaion may be found in gestures, the interpretation module searches the gestlet buffer for specific categories of gestures. The gesture parsing routines produce a broad description of the hand motion that occurred, using various weighted parameters. A pointing gesture, for example, would include *attack* (motion towards the gesture space), *sweep* (motion from side to side), and *end reference space* (position of hand at the end of the motion). By adding up the parameter weights, and looking for various logical combinations of gestlets, the parameters provide a way to estimate the likelihood that a certain categry of gesture happened. For this specific example, if a deictic gesture is gound, the hand orientation would be used to find a vector in three-space that intersects the screen.

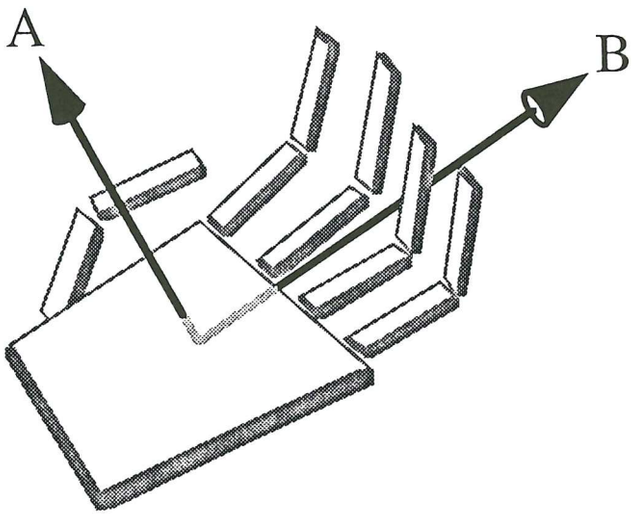

Figure 10. Normal and Longitudinal Vectors from Palm

4.4. Example Evaluation

A second prototype system, based on these modifications, enables a user to manipulate objects in a simple "blocks world" by using not only deictic but iconic and pantomimic

gestures as well. A typical interaction is shown in Figure 11. The user in this example wishes to move and rotate a cylinder so that it ends up in a particular orientation next to a cube. It is important to note that gesture types are fluidly mixed in situations such as this. The raw data must be processed in a way that these motions can be separated and interpreted in the context of the accompanying speech.

While the hand data is being processed into gestlets, the speech recognizer is converting the voice input into words. The eye tracking module is also working in parallel determining fixations, saccades and blinks. Each channel of information is brought to a similar level of abstraction allowing for close examination of the interdependencies of the modalities.

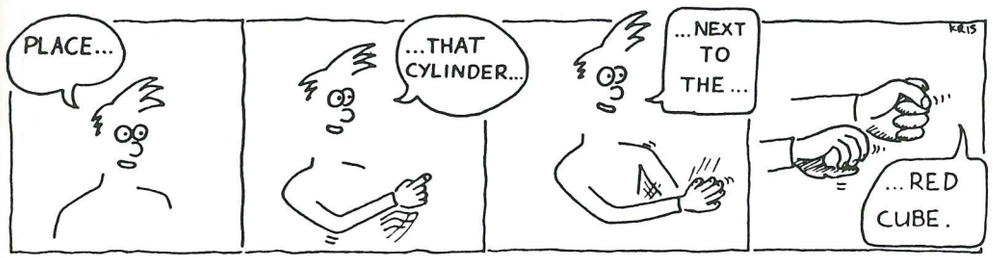

Figure 11. Typical Interaction

Processing the speech produces a structure as shown in Figure 12. The first part of the command to be resolved is the reference with the phrase "that cylinder." The use of "that" in the utterance *"Place that cylinder"* strongly suggests a deictic gesture. The speech and gesture phrases are tied together and interpreted in the spatial representational system to determine a referent in the environment. The remainder of the directive, "next to the red cube," indicates the possibility of either a deictic or an iconic gesture.

The phrase "next to the red cube" allows for several interpretations. The first might be that the user doesn't care about the exact location and orientation of the cylinder, only that it is in close proximity to the cube. In that case, no gesture information would have been given, and the exact placement of the cylinder could be arbitrary. Alternatively, the person might use a deictic gesture and point to a location near the cube where the cylinder should be placed.

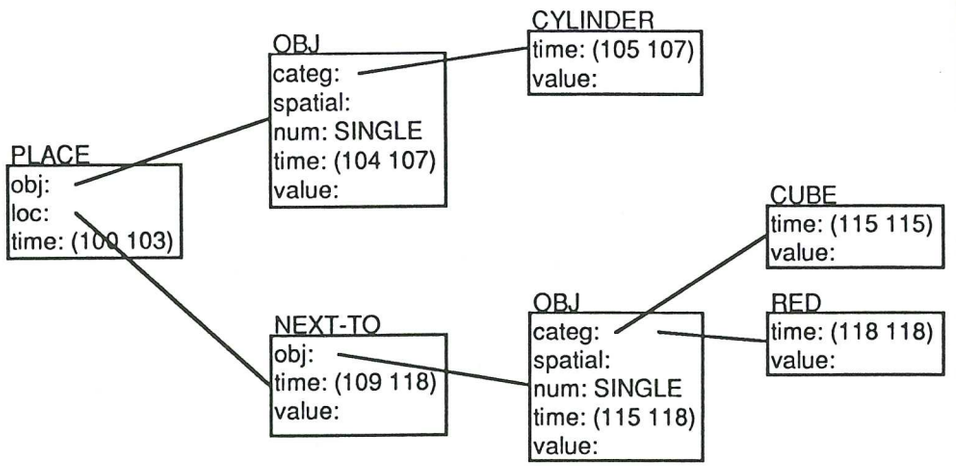

Figure 12. Structure After Speech Processing

A third possibility is the one shown in Figure 11. The person has used a two-handed iconic gesture to indicate not only the relative position of the cylinder with respect to the cube but also the orientation. The movement of both hands is important. First, the left hand is brought up to represent the cube. Then the right hand is moved in next to it to show the relative positioning of the two objects. Orientation information is provided by the right hand: it is placed in a curled posture suggesting that an invisible cylinder is being held.[1] With one command the user has accomplished a selection, a three-dimensional translation, and a rotation around all three axes.

5. Beyond Looking as Deixis

Most uses of gaze at the computer interface have been interpreted as deictic gestures indicating the user's interest [Thorisson et al. 1992; Starker and Bolt 1990; Jacob 1990]. However, like gestures, eye movements can be interpreted in many different ways, depending on the context in which they are performed. For example, gaze is a good indicator of a person's attention over time [Kahneman 1973] and could be useful in predicting the user's behavior in the context of a broader task. The eyes also play a very special role in social interaction; they are important in the regulation of turn-taking between participants in a dialog (who has control of the "floor") [Argyle and Cook 1976]. Turn taking is crucially important in both clarification and negotiation [Whittaker and Walker 1991]. Additionally, eye movements have been found to play a significant role in conveying personality, emotional states, and interpersonal attitudes [Argyle et al. 1974; Kleinke 1986].

Future work should include the ability to incorporate this eye behavior information in an integrated interpretation process. Someday, significant looks, tired stares, winks and rapid searches can all be useful in our interaction with machines.

6. Summary

We have shown a frame-based method of interpreting multi-modal input. The system takes into account various types of gestures, fairly complex speech and deictic gaze. By

[1] The point might also be argued that the user is representing the cylinder by the shape of his hand. According to Rimé and Schiaratura's modified Efron classification, this would be an iconic gesture. In many cases this distinction will not make a difference for interpretation.

bringing gestures to an intermediate representational level, interpretation of hand data is made more flexible. The gesture parsing method described can currently handle deictic references, pantomimic and iconic gestures, but should be easily extended to accommodate other categories as well.

Future work should include extensions such as full-arm and full-body descriptions. We are currently working on methods to accommodate more complex interpretations of looking and are exploring other gesture categories in a variety of contexts.

Acknowledgments

We would like to thank our director, Dr. Richard Bolt, and acknolwedge the contributions of graduate students Brent Britton and Edward Herranz, and assistants David Berger, Brian Brown, Michael Johnson, Mathew Kaminski, Brian Lawrence, Christopher Wren, and research affiliate Masaru Sugai, NEC, Japan.

This research was supported by the Defense Advanced Research Projects Agency (DARPA) under Rome Laboratories, contract F30602-89-C-0022.

Section 3

Architectural and Theoretical Issues

The papers in this final section address architectural and theoretical issues that underlie intelligent multimedia interfaces. The first chapter by Yigal Arens, Eduard Hovy, and Mira Vossers addresses the media allocation problem, that is how and on what basis should information be apportioned to different kinds of media (e.g., text, charts, maps, tables, menus). They first ague that information (e.g., ships locations) should not be directly allocated to particular media (e.g., text or maps) rather characteristics of information (e.g., data with spatial denotations) should be assigned to characteristics of media (e.g., graphs, tables, and maps are planar media). They identify four classes of knowledge that are required to allocate information to media – the characteristics of the information, the characteristics of the media, the goals and nature of the speaker, and the nature of the perceiver and the communicative situation. For example, information characteristics include dimensionality, transience, urgency, quantity; media characteristics include dimensionality, temporal endurance, visual/aural nature. They formalize a range of these features in a systemic network and indicate interdependencies among features as rules or constraints between producer goals, information content, and surface features of presentations. They conclude by illustrating how these knowledge sources could be used to produce and interpret multimedia displays, in particular to indicate the location of Paris and to analyze an illustrated instruction explaining how to adjust a Honda car seat.

Andrea Bonarini also discusses the kinds of models required to support multimedia communication, however, in the context of the communication between a driver and an artificial co-pilot. Bonarini discusses the need to model the features of interaction tools, including general features such as the sensory channel used (e.g., auditory, visual) as well as particular characteristics (e.g., the type of action and attention required to manipulate a tool, the fidelity of the tool). A model of the driver is derived from his behavior (e.g., using a history of the velocity and acceleration of the vehicle, braking and