Chapter 14:

GESTURE BASED INTERACTION*

Introduction

Gestures and gesture recognition are terms increasingly encountered in discussions of human-computer interaction. For many (if not most) people the term includes character recognition, the recognition of proof readers symbols, shorthand, and all of the types of interaction described in the previous chapter, Marking Interfaces. In fact every physical action involves a gesture of some sort in order to be articulated. Furthermore, the nature of that gesture is generally an important component in establishing the quality of feel to the action. Nevertheless, what we want to isolate for discussion in this chapter are interactions where the gesture is what is articulated and recognized, rather than a consequence of expressing something through a transducer. Thus we use the definition of gesture articulated by Kurtenbach and Hulteen (1990):

"A gesture is a motion of the body that contains information. Waving goodbye is a gesture. Pressing a key on a keyboard is not a gesture because the motion of a finger on its way to hitting a key is neither observed nor significant. All that matters is which key was pressed".

And, of course, this is true regardless of the gesture that was used to push the key. It could have been pushed lovingly or in anger. Either could be easily sensed by an observer. But both are irrelevant to the computer, which only cares about what key was pushed when.

The type of communication that we are discussing here is far richer in many ways than what we have been dealing with. Consequently, it is not hard to understand why this use of gesture requires a different class of input devices then we have seen thus far. For the most part, gestures, as we discuss them, involve a far higher number of degrees of freedom than we have been looking at. Trying to do gesture recognition by using a mouse or some other "single point" device for gestural interaction restricts the user to the gestural

* The primary author of this chapter is Mark Billinghurst.

vocabulary of a fruit fly ! You may still be able to communicate, but your gestural repertoire will be seriously constrained.

The first step in considering gesture based interaction with computers is to understand the role of gesture in human to human communication. In the next section we review the psychology and anthropology literature to categorize the types of gestures that are commonly made and their attributes. In the remainder of the chapter we use these categories to discuss gesture based interfaces, from symbolic gesture systems to multimodal conversational interfaces. We end with a discussion of future research directions, in particular reactive environments where the user's entire surroundings is able to understand their voice and gestural commands and respond accordingly.

Gestures in the Everyday World

If we remove ourselves from the world of computers and consider human-human interaction for a moment we quickly realize that we utilize a broad range of gesture in communication. The gestures that are used vary greatly among contexts and cultures (Morris, Collet, Marsh & O'Shaughnessy 1980) yet are intimately related to communication. This is shown by the fact that people gesticulate just as much when talking on the phone and can't see each other as in face to face conversation (Rime 1982).

Gestures can exist in isolation or involve external objects. Free of any object, we wave, beckon, fend off, and to a greater or lesser degree (depending on training) make use of more formal sign languages. With respect to objects, we have a broad range of gestures that are almost universal, including pointing at objects, touching or moving objects, changing object shape, activating objects such as controls, or handing objects to others. This suggests that gestures can be classified according to their function. Cadoz (1994) uses function to group gestures into three types:

- *semiotic:* those used to communicate meaningful information.
- *ergotic:* those used to manipulate the physical world and create artifacts
- *epistemic:* those used to learn from the environment through tactile or haptic exploration

Within these categories there may be further classifications applied to gestures. Mulder (1996) provides a summary of several different classifications, especially with respect to semiotic gestures.

In this chapter we are primarily interested in how gestures can be used to communicate with a computer so we will be mostly concerned with empty handed semiotic gestures. These can further be categorized according to their functionality. Rime and Schiaratura (1991) propose the following gesture taxonomy:

- *Symbolic gestures:* These are gestures that, within each culture, have come to have a single meaning. An Emblem such as the "OK" gesture is one such example, however American Sign Language gestures also fall into this category.

- *Deictic gestures:* These are the types of gestures most generally seen in HCI and are the gestures of pointing, or otherwise directing the listeners attention to specific events or objects in the environment. They are the gestures made when someone says "Put that there".

- *Iconic gestures:* As the name suggests, these gestures are used to convey information about the size, shape or orientation of the object of discourse. They are the gestures made when someone says "The plane flew like this", while moving their hand through the air like the flight path of the aircraft.

- *Pantomimic gestures:* These are the gestures typically used in showing the use of movement of some invisible tool or object in the speaker's hand. When a speaker says "I turned the steering wheel hard to the left", while mimicking the action of turning a wheel with both hands, they are making a pantomimic gesture.

To this taxonomy McNeill (1992) adds types of gestures which relate to the process of communication; *beat gestures* and *cohesives*. Beat or baton gestures are so named because the hand moves up and down with the rhythm of speech and looks like it is beating time. Cohesives, on the other hand, are variations of iconic, pantomimic or deictic gestures that are used to tie together temporally separated but thematically related portions of discourse.

Gesture is also intimately related to speech, both in it's reliance on the speech channel for interpretation, and for its own speech like-qualities. Only the first class of gestures, symbolic, can be interpreted alone without further contextual information. Either this context has to be provided sequentially by another gesture or action, or by speech input in concert with the gesture. So these gesture types can also be categorized according to their relationship with speech:

- Gestures that evoke the speech referent: ***Symbolic, Deictic***
- Gestures that depict the speech referent*:* ***Iconic, Pantomimic***
- Gestures that relate to conversational process*:* ***Beat, Cohesive***

The need for a speech channel for understanding varies according to the type of gesture. Thus gesture types can be ordered according to their speech/gesture dependency. This is described in Kendon's Continuum (Kendon 1988):

Gesticulation -> Language-Like -> Pantomimes -> Emblems -> Sign Language
(Beat, Cohesive) (Iconic) (Pantomimic) (Deictic) (Symbolic)

Progressing from left to right the necessity of accompanying speech to understand the gesture declines, the gestures become more language-like, and idiosyncratic gestures are replaced by socially regulated signs. For example sign languages share enough of the syntactic and semantic features of speech that they don't require an additional speech channel for interpretation. However iconic gestures cannot be understood without accompanying speech.

In contrast to this rich gestural taxonomy, current interaction with computers is almost entirely free of gestures. The dominant paradigm is direct manipulation, however we may wonder how direct are direct manipulation systems when they are so restricted in the ways that they engage our everyday skills. This deficiency is made obvious when we consider how proficient humans are at using gestures in the everyday world and then consider how few of these gestures can be used in human-computer interaction and how long it takes to learn the input gestures that computers can understand. Even the most advanced gestural interfaces typically only implement symbolic or deictic gesture recognition. However this need not be the case. In the remainder of the chapter we move along Kendon's Continuum from right to left reviewing computer interfaces from each of three categories; gesture only interfaces, gesture and speech interfaces, conversational interfaces.

As we shall see from this review, one of the compelling reasons for using gesture at the interface is because of it's relationship to the concepts of chunking and phrasing. In chapter seven we described how the most intuitive interfaces match the phrase structure of the human-computer dialogue with the cognitive chunks the human should be learning. Unintuitive interfaces require simple conceptual actions to be broken up into compound tasks; for example a *Move* action that requires separate *Cut* and *Paste* commands. In contrast, gesture based interfaces allow the use of natural gestural phrases that chunk the input dialog into units meaningful to the application. This is especially the case when voice input is combined with gesture, allowing the user to exactly match their input modalities to the cognitive chunks of the task. For example, saying the command "move the ball like this"

while showing the path of the ball with an iconic gesture specifies both a command and its relevant parameters in a single cognitive chunk.

Gesture Only Interfaces

The gestural equivalent of direct manipulation interfaces are those which use gesture alone. These can range from interfaces that recognize a few symbolic gestures to those that implement fully fledged sign language interpretation. Similarly interfaces may recognize static hand poses, or dynamic hand motion, or a combination of both. In all cases each gesture has an unambiguous semantic meaning associated with it that can be used in the interface. In this section we will first briefly review the technology used to capture gesture input, then describe examples from symbolic and sign language recognition. Finally we summarize the lessons learned from these interfaces and provide some recommendations for designing gesture only applications.

Tracking Technologies

Gesture-only interfaces with a syntax of many gestures typically require precise hand pose tracking. A common technique is to instrument the hand with a glove which is equipped with a number of sensors which provide information about hand position, orientation, and flex of the fingers. The first commercially available hand tracker, the Dataglove, is described in Zimmerman, Lanier, Blanchard, Bryson and Harvill (1987), and illustrated in the video by Zacharey, G. (1987). This uses thin fiber optic cables running down the back of each hand, each with a small crack in it. Light is shone down the cable so when the fingers are bent light leaks out through the cracks. Measuring light loss gives an accurate reading of hand pose. The Dataglove could measure each joint bend to an accuracy of 5 to 10 degrees (Wise et. al. 1990), but not the sideways movement of the fingers (finger abduction). However, the CyberGlove developed by Kramer (Kramer 89) uses strain gauges placed between the fingers to measure abduction as well as more accurate bend sensing (Figure 1). Since the development of the Dataglove and Cyberglove many other glove based input devices have appeared as described by Sturman and Zeltzer (1994).

Figure 1: The CyberGlove
The CyberGlove captures the position and movement of the fingers and wrist. It has up to 22 sensors, including three bend sensors (including the distal joints) on each finger, four abduction sensors, plus sensors measuring thumb crossover, palm arch, wrist flexion and wrist abduction. (Photo: Virtual Technologies, Inc.)

Once hand pose data has been captured by the gloves, gestures can be recognized using a number of different techniques. Neural network approaches or statistical template matching is commonly used to identify static hand poses, often achieving accuracy rates of better than 95% (Väänänen and Böhm 1993). Time dependent neural networks may also be used for dynamic gesture recognition **[REF]**, although a more common approach is to use Hidden Markov Models. With this technique Kobayashi is able to achieve an accuracy of **XX**% (Kobayashi et. al. 1997), similar results have been reported by **XXXX** and **XXXX**. Hidden Markov Models may also be used to interactively segment out glove input into individual gestures for recognition and perform online learning of new gestures (Lee 1996). In these cases gestures are typically recognized using pre-trained templates, however gloves can also be used to identify natural or untrained gestures. Wexelblat uses a top down and bottom up approach to recognize natural gestural features such as finger curvature and hand orientation, and temporal integration to produce frames describing complete gestures (Wexelblat 1995). These frames can then be passed to higher level functions for further interpretation.

Although instrumented gloves provide very accurate results they are expensive and encumbering. Computer vision techniques can also be used for gesture recognition overcoming some of these limitations. A good review of vision based gesture recognition is provided by Palovic et. al. (1995). In general, vision based systems are more natural to use that glove interfaces, and are capable of excellent hand and body tracking, but do not provide the same accuracy in pose determination. However for many applications this may not be important. Sturman and Zeltzer point out the following limitations for image based visual tracking of the hands (Sturman and Zeltzer 1994):

- The resolution of video cameras is too low to both resolve the fingers easily and cover the field of view encompassed by broad hand motions.
- The 30- or 60- frame-per-second conventional video technology is insufficient to capture rapid hand motion.

- Fingers are difficult to track as they occlude each other and are occluded by the hand.

There are two different approaches to vision based gesture recognition; model based techniques which try to create a three-dimensional model of the users hand and use this for recognition, and image based techniques which calculate recognition features directly from the hand image. Rehg and Kanade (1994) describe a vision-based approach that uses stereo camera to create a cylindrical model of the hand. They use finger tips and joint links as features to align the cylindrical components of the model. Etoh, Tomono and Kishino (1991) report similar work, while Lee and Kunii use kinematic constraints to improve the model matching and recognize 16 gestures with **XX**% accuracy (1993). Image based methods typically segment flesh tones from the background images to find hands and then try and extract features such as fingertips, hand edges, or gross hand geometry for use in gesture recognition. Using only a coarse description of hand shape and a hidden markov model, Starner and Pentland are able to recognize 42 American Sign Language gestures with 99% accuracy (1995). In contrast, Martin and Crowley calculate the principle components of gestural images and use these to search the gesture space to match the target gestures (1997).

Natural Gesture Only Interfaces

At the simplest level, effective gesture interfaces can be developed which respond to natural gestures, especially dynamic hand motion. An early example is the Theremin, an electronic musical instrument from the 1920's. This responds to hand position using two proximity sensors, one vertical, the other horizontal. Proximity to the vertical sensor controls the music pitch, to the horizontal one, loudness. What is amazing is that music can be made with orthogonal control of the two prime dimensions, using a control system that provides no fixed reference points, such as frets or mechanical feedback. The hands work in extremely subtle ways to articulate steps in what is actually a continuous control space **[REF].** The Theremin is successful because there is a direct mapping of hand motion to continuous feedback, enabling the user to quickly build a mental model of how to use the device.

Figure 2: The Theremin.
The figure shows Dr. Robert Moog playing the Theremin. This electronic musical instrument generates a violin-like tone whose pitch is determined by the proximity of the performer's right hand to the vertical antenna, and the loudness is controlled by the proximity of the left hand to the horizontal antenna. Hence, a musical performance requires control over great subtlety of nuance over gesture on the part of the artist, with no mechanical aids (such as frets) as a guide. It is an extreme example of the human's potential to articulate controlled gestures. (Photo: Big Briar, Inc.)

Myron Krueger's Videoplace is another system with responds to natural user gesture (Krueger 1991). Developed in the late 1970's and early 80's, Videoplace uses real time image processing of live video of the user. Background subtraction and edge detection are used to create a silhouette of the user and relevant features identified. The feature recognition is sufficiently fine to distinguish between hands and fingers, whether fingers are extended or closed, and even which fingers. With this capability, the system has been programmed to perform a number of interactions, many of which closely echo our use of gesture in the everyday world.

Videoplace is a stunning piece of work, displaying an extremely high degree of virtuosity and creativity. The key to its success is the recognition of dynamic natural gestures, meaning users require no training. Figure 3 shows a kind of "finger painting" while Figure 4 shows how one can select from a menu (in this case the alphabet, thereby enabling text entry) by pointing at items with the index finger. Finally, Figure 5 shows an object being manipulated by simultaneously using the index finger and thumb from both hands.

Figure 3: Finger Painting
Here the index finger is recognized and when extended, becomes a drawing tool. Shaping the hand in a fist, so that the finger is no longer extended lets the hand be moved without inking.

Figure 4: Selecting from a Menu
Here, text is entered by pointing at the character desired. Note that the alphabet is simply a menu, and the example demonstrates how extending the index finger can be used as a generalized selection tool.

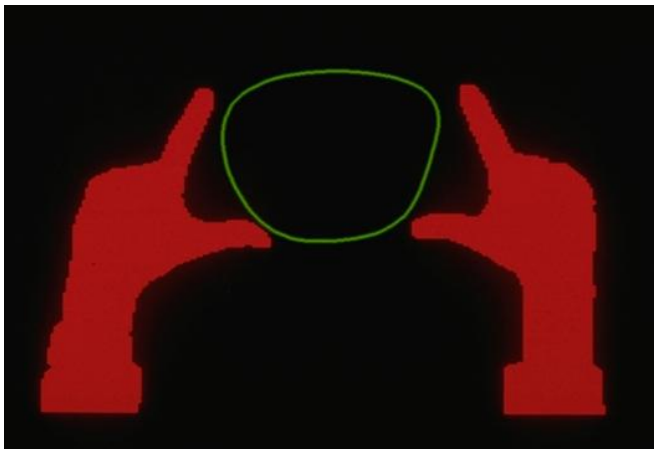

Figure 5: Controlling a Bezier Curve.
Here the index fingers and thumbs of the two hands are recognized and are used to control the shape of the object being defined.

An interesting issue that arises with the system is that of segmentation. Since the system is driven by the gesture of the hand, how in the previous figure, for example, does the user specify that the current form of the curve is to be fixed? If this is not done, the curve will "stick" to the fingers and distort when the hand goes to perform some other task. The solution generally used is to utilize a temporal cue: if the hands hold the shape in place for a certain time, a beep is heard indicating that it has been fixed. The hands can now go and do something else. The interaction is much like having to hold something in place until the glue dries (but where the glue is relatively fast drying).

Symbolic Gesture Recognition

VideoPlace and the Theremin both respond to natural free form gestures, however interfaces with a wider range of commands may require a symbolic gesture interface. In this case certain commands are associated with pretrained gesture shapes. Symbolic gesture interfaces are often used in immersive virtual environment where the user cannot see the real world to traditional input devices. In this setting there are typically a set of pre-trained gestures used for navigation through the virtual environment and interaction with virtual objects. For example in the *Rubber Rocks* virtual environment, users could pick up virtual rocks by making a fist gesture, and throw and release rocks with a flat hand gesture (Codella, 1992). The GIVEN virtual environment (Gesture-driven Interactions in Virtual Environments), uses a neural network to recognize up to twenty static and dynamic gestures (Vaananen and Bohm, 1993). These include pointing gestures for flying, fist gestures for grabbing and other whole hand gesture for releasing objects or returning back to the starting point in the virtual environment.

As Baudel and Beaudouin-Lafon (1993) point out there are a number of advantages in using symbolic gestures for interaction, including:

- *Natural Interaction:* Gestures are a natural form of interaction and easy to use.
- *Terse and Powerful:* A single gesture can be used to specify both a command and its parameters.
- *Direct Interaction:* The hand as input device eliminates the need for intermediate transducers.

However the are problems with using symbolic gesture only interfaces. Users may become tired making free-space gestures and gesture interfaces are not self-revealing, forcing the user to know beforehand the set of gestures that the system understands. Naturally, it becomes more difficult to remember the gestural command set as the number of gesture

increase. There is also a segmentation problem, in that tracking systems typically capture all of the user's hand motions so any gestural commands must be segmented from this continuous stream before being recognized. This causes a related problem in that the gestures chosen may also duplicate those that are very natural and used in everyday life. For example, the pointing gesture using the index finger is often used to fly through the virtual space. However since this gesture is also performed during other natural behaviors (such as scratching one's chin), resulting in unintended flights.

CASE STUDY Charade:

Charade (Baudel & Beaudouin-Lafon, 1993), is an excellent example of an effective gesture only interface. Charade uses dynamic hand gestures to control a computer-aided presentation. As shown in Figure 6, the user stands in front of a projection screen wearing a DataGlove connected to a Macintosh computer. When they point their hand towards the projection screen they can issue one-handed gestural commands controlling the Hypercard presentation shown on the screen, such as advancing to the next slide. There are a total of 16 commands that can be recognized. Charade recognizes 70-80% of first time user's gestures correctly and has an accuracy of 90-98% for trained users.

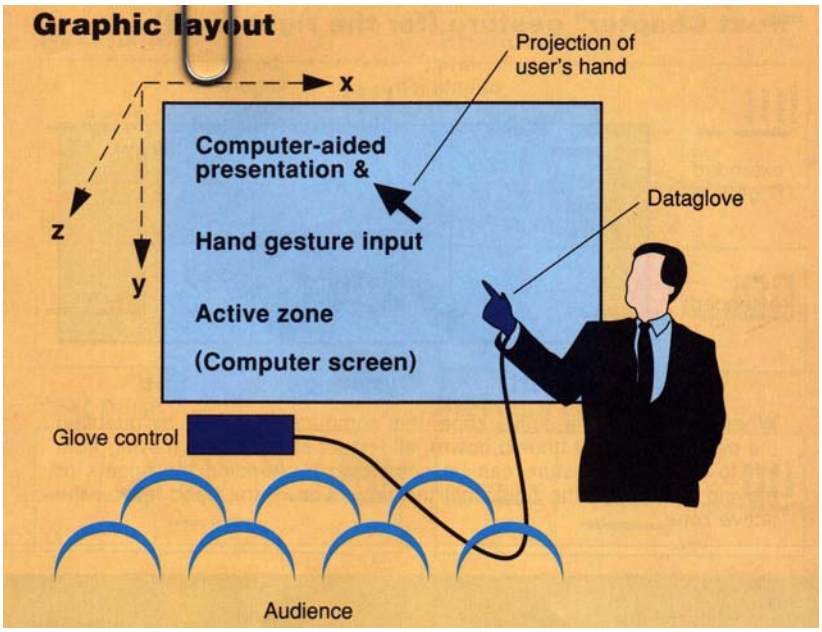

Figure 6: The Charade Interface

To ensure accurate gesture recognition and an intuitive interface a number of constraints are applied. A region in front of the projection screen is defined as the active zone and gestures are ignored if they were performed out of this area. This enables users to mix gesture commands with normal conversational gestures. Gestures are also defined by a set start posture, end posture and dynamic motion between the start and end postures. These constraints enable gesture recognition to be performed using the following steps:

- *Detection of intention* – gestures were only interpreted when they were made within the active zone
- *Gesture Segmentation* – static start and end postures aided in the separation of command gestures from normal user gestures.
- *Gesture Classification* – gestures were classified according to their start position and dynamic motion.

In addition, the start positions differ from all the end positions and the gesture commands do not differ solely by their end position. This enables users to string together multiple commands in a single hand motion and for users to complete a command by either using an end posture or moving their hand out of the active zone.

In developing Charade, Baudel and Beaudouin-Lafon arrived at four guidelines that are generally applicable to symbolic gestural interfaces:

- *Use Hand Tension:* In Charade the start positions all involved tensing the hand into static postures. This makes explicit the user's intention to issue a command. Tension also emphasizes the structure of the human-computer dialogue. Conversely end positions should not be tensed.

- Provide *Fast, Incremental and Reversible Actions:* This is one of the basic principles of direction manipulation interfaces adapted for gestural input. Speed is essential so that user does not get tired forming gestures, reversibility is important to enable the user to undo any action, and incremental actions are vital for the system to provide continuous feedback to improve the users confidence in the interface.

- *Favor Ease of Learning:* In symbolic gestural interfaces a compromise must be made between natural gestures that are immediately learned by the user and complex gestures that give more control. In Charade, the most common interface actions are mapped to the most natural gestures, ensuring ease of learning.

- *Use Hand Gesture for Appropriate Tasks:* It is important to chose carefully the tasks that gesture input is going to be used for. While gesture input is natural for some navigation and direct manipulation tasks, it is inappropriate for tasks that require precise interaction or manipulation. These tasks typically require some form of physical contact.

Sign Language Recognition

A obvious application for gesture interfaces is in the interpretation of formal sign language. In contrast with other gestures, sign language does not rely on other input modalities for interpretation and can be used to completely express syntactic and semantic information. Perhaps the most interesting use of sign language interfaces is in sign to speech translation. Starner and Pentland's work **[SUMMARISE THEIR WORK]**

While this work is useful for discrete letter or word recognition, sign language interfaces can also be used for lower level continuous speech production. GloveTalk (Fels and Hinton 1993) and GloveTalk II (Fels and Hinton 1995) are systems that translate hand gestures into word, vowel and consonant sounds. In the original GloveTalk, features from a glove and magnetic hand tracker are used as input into five neural networks that control speech production. Hand shape determined the root word with its ending specified with the direction of hand movement. The speed of movement and amount of displacement also fixed the rate of speech and syllable stress. Glove-Talk II extended this by allowing production of individual speech formants. Three neural networks are used to allow gestures to continuously control 10 parameters of a parallel formant speech synthesizer. In this way the users hand gestures control an artificial vocal tract the produces speech in real time, and they have an unlimited vocabulary as well as control over formant pitch and volume. However, significant training is needed to learn the gestures required by the system; one user was able to speak intelligibly after 100 hours of practice. This is much less than previous attempts at gesture to speech interfaces.

Designing Gesture Only interfaces

In this section we have described three types of gesture only interfaces; those which interpret natural gesture, symbolic input, or those which understand formal sign languages. These can be distinguished by the amount of training required, and the expressiveness and usability of the interface. Although natural interfaces such as those developed by Kruger are easy to use, and don't need training, they only provide a limited command vocabulary. In contrast Fels' interface can produce any word in the English language, but requires many hours of training before it is usable. This suggests that in developing gesture based interfaces designers must carefully consider the command vocabulary required and use this to select the type of gestural input most appropriate.

Sturman and Zeltzer provide an iterative design method for whole hand gestural interfaces (Sturman, Zeltzer 1993). This enables a developer to evaluate the appropriateness of using whole hand input for an interface and then design an efficient interface by matching task characteristics to hand action capabilities and device attributes. The design method is broken into several stages, (Figure 7**)**. The first of these, *Appropriateness for Application,* involves determining the appropriateness of an application for whole hand input by considering the following series of questions about the application in the areas of naturalness, adaptability and coordination:

Naturalness:

Are the following characteristics useful for controlling the application tasks?

- Pre-acquired sensorimotor skills
- Existing hand signs
- Absence of an intermediary device
- Task control maps well to hand actions (position and motion of hand).

Adaptability:

Are diverse modes of control used in the tasks?
Is it important to be able to switch between modes of control rapidly, and smoothly?

Coordination:

Do the tasks require the coordination of many degrees of freedom?

The more positive answers to these questions the more suitable gesture input is for the application. Once gesture input is determined to be appropriate it is necessary to develop a *Taxonomy* which describes how gestures will be used in the interface. The taxonomy categorizes possible hand actions and the interpretation of hand actions by the application, and helps discriminate between the different input styles. The enables the designer to select the gesture input most suitable for the application. For example, if there are many desired input modes than perhaps symbolic recognition is most appropriate, which if only a few natural gestures may suffice.

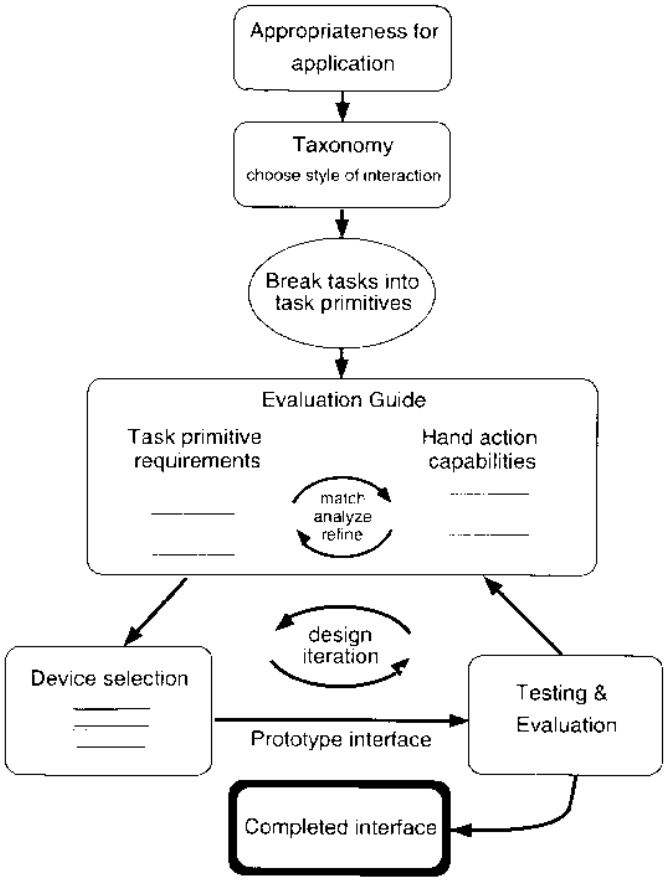

Figure 7: Design Method for Whole Hand Input – from Sturman and Zeltzer (1993)

Next, the application is broken down into task primitives that can be quantified in ways that relate to specific hand actions. Table X.X shows a list of task characteristics and related hand action capabilities. The task characteristics describe the task primitive to be performed while the hand action capabilities describe the control available from the hand. Sturman (1992) gives a complete description of these measures. As the application is broken down into task primitives, the developer can use these measures to select potential hand actions for each primitive. In many cases the developer may need to carry out experimental trials to quantify the hand action capabilities.

The final step in the design method is choosing a particular gesture input device that matches the task characteristics and hand actions. Potential devices should evaluated according to a set of measure such as those given in Table 1. These measures enable an easy comparison to the task characteristics and hand action capabilities. Once the input device has been chosen and a prototype developed an iterative design process can begin which leads to the best gestural input method for the target application.

Using this design method Sturman and Zeltzer evaluate the use of whole hand input for three different interface activities, controlling the walking rate of a virtual robot, orienting a virtual object and moving a virtual robot along a path. In each case hand input with a DataGlove was compared to input using a dial box. The design method predicted that walking control using the DataGlove would be better than the dial box, there would be no difference between input modalities for the orientation task, and that the glove input would prove worse than the dial box for path following. User studies verified these predictions, showing the validity of the design method for gestural interface design.

Task Characteristics	Hand Action Capabilities	Device Capabilities
Degrees of Freedom	Degrees of Freedom	Degrees of Freedom
Task Constraints - Degrees of freedom - Physical constraints - Temporal constraints - External forces	Hand Constraints - range of motion - coupling - spatial interference - strength	Cross-coupling
Coordination	Coordination	Device Constraints
Resolution - spatial - temporal	Resolution - spatial - temporal	Fidelity
Speed	Speed	Resolution
Repeatability	Repeatability	Steadiness
Steadiness	Steadiness	Reliability
Endurance	Endurance	Mass
Expressiveness	Expressiveness	Comfort
Modality	Adaptability	Convenience
Task Analogy - comparison to existing methods - similarity to other tasks	Familiarity - similarity to existing skills - similarity to everyday motions	Sampling Rate
		Computation Required

Table 1: the Relationship between Task Characteristics, Hand Action and Device Capabilities.

Speech with Gesture

Perhaps use of gesture is most powerful when combined with other input modalities, especially voice. Allowing combined voice and gestural input has several tangible advantages. The first is purely practical; ease of expression. As Martin (1989) points out, typical computer interaction modalities are characterized by an ease versus expressiveness trade-off. Ease corresponds to the efficiency with which commands can be remembered, and expressiveness the size of the command vocabulary. Common interaction devices range from the mouse that maximizes ease, to the keyboard that maximizes expressiveness. Multimodal input overcomes this trade-off; combined speech and gestural commands are easy to execute whilst retaining a large command vocabulary.

Voice and gesture complement each other and when used together, creating an interface more powerful that either modality alone. Cohen (Cohen et. al. 1989, Cohen 1992) shows how natural language interaction is suited for descriptive techniques, while gestural interaction is ideal for direct manipulation of objects. For example, unlike gestural or mouse input, voice is not tied to a spatial metaphor [11]. This means that voice can interact with objects regardless of degree of visual exposure, particularly valuable in a virtual environment where objects may be hidden inside each other or occluded by other objects. Some tasks are inherently graphical, others are verbal and yet others require both vocal and gestural input to be completed [10]. So allowing both types of input maximizes the usefulness of an interface by broadening the range of tasks that can be done in an intuitive manner. Cohen points out many complimentary attributes of direct manipulation and natural language interfaces as summarized in Table 2[ref].

	Direct Manipulation	Natural Language
Strengths	1. Intuitive 2. Consistent Look and Feel 3. Options Apparent 4. Fail Safe 5. "Direct Engagement" with object a. Point, act b. Feedback	1. Intuitive 2. Description, e.g., a. Quantification b. Negation c. Temporal d. Motivation/Cause 3. Context 4. Anaphora 5. Asynchronous
Weaknesses	1. Description 2. Anaphora 3. Operation on sets 4. Delayed actions difficult	1. Coverage is opaque 2. Overkill for short or frequent queries 3. Difficulty of establishing and navigating context; Spatial specification cumbersome 4. Anaphora problematic 5. Error prone (ambiguity, vagueness, incomplete)

Table 2: Complimentary Characteristics of Direct Manipulation and Natural Language Interfaces

For many types of applications, users prefer using combined voice and gestural communication to either modality alone. For example, Oviatt and Olsen (199X) evaluated speech and pen-based multimodal input in a verbal task and a numerical task. In the verbal task 56% of the users preferred combined pen/voice input, while 89% preferred it in the numerical task to using either modality alone. These results agree with those of Hauptman and MacAvinney [9] who used a simulated speech and free hand gesture recognizer in a typical graphics task. Three different modes were tested - gesture only, voice only, and gesture and voice recognition. Users overwhelmingly preferred combined voice and gestural recognition due to the greater expressiveness possible. When combined input was possible, subjects used speech and gesture together 71% of the time as opposed to voice only (13%) or gesture only (16%).

Combining speech, and gesture improves recognition accuracy and produces faster task completion time compared to speech only interfaces. Using a multimodal speech and pen-based interface, Oviatt (1996) evaluated user performance on map tasks performed using only speech, only pen, or combined speech and pen input. She found that multimodal input produced a 36% reduction in task errors and 23% fewer spoken words resulting in a 10% faster completion times compared to the speech only interface. Similarly, Martin (1989) finds that people using speech input for CAD programs were able to remain visually focused on the screen while using speech commands, causing a 108% improvement in productivity over purely keyboard entry. This was due to the additional response channel provided by the speech input as well as speech being a more efficient response channel than typed input.

There are also psychological reasons for integrating speech and gesture recognition. Experiments in cognitive psychology have shown that a person’s ability to perform multiple tasks is affected by whether these tasks use the same or different sensory modes, for example visuo/spatial or verbal modes. According to the multiple resource theory of attention [6,7] the brain modularizes the processing of different types of information - when different tasks tap different resources much of the processing can go on in parallel. Such is the case with speech and visuo/spatial modalities. So by adding speech input to the interface users should be able to perform gestural tasks at the same time as giving verbal commands with little cognitive interference. Experimental evidence supports this theory.

Treisman and Davis [8] found that the ability to concentrate on more than one task at a time was expanded when the tasks were presented in separate perceptual channels and people responded to them across different response channels. This is in part due to the spatial and verbal information being stored separately within human memory [9].

Simple Multimodal Interfaces

One of the first interfaces to support combined speech and gesture recognition was the Media Room constructed by Negroponte's Architecture Machine Group (Bolt 1984). Designed by Richard Bolt, the Media Room allowed the user to sit inside the computer interface and interact with the computer through voice, gesture and gaze. It consisted of a large room, one wall of which was a back projection panel. The user sat in the center of the room in a chair wearing a magnetic position sensing devices on their wrist to measure pointing gestures (Figure 8).

Figure 8: The Media Room

Within the Media Room the user could use speech, gesture, eye movements or a combination of all three to add, delete and move graphical objects shown on the wall projection panel. The computer interpreted the user's intentions by speech and gesture recognition and by taking the current graphical situation into account. For example, one application allowed users to manage color-coded ships against a map of the Caribbean. By pointing at a spot above Haiti and saying "Create a large blue tanker there (pointing)", a blue tanker would appear above Haiti. An existing object could be moved by saying "Put that (pointing at object) there (pointing at destination)", colored ("Make that red"), or named ("call that (pointing to object) the Flying Cloud".)

By integrating speech and gesture recognition with contextual understanding, Bolt (Bolt 1984) discovered that neither had to be perfect provided they converged on the users intended meaning. This is because the computer responds to users commands by using speech and gesture recognition and taking the current context into account. For example, if a user says "Create a blue square there (pointing at a screen location)", and the speech recognizer fails to recognize "Create", the sentence can still be correctly understood by considering the graphical context. If there are no blue squares present then the missed word could only be "Create" and the system responds correctly. Coupling voice and gesture recognition in this way means that pronouns and adverbs can be used in speech instead of proper names and complicated object descriptions, considerably reducing the

complexity of the recognition task. The user could say "Put that there", instead of "Move the blue freighter to the northeast of Haiti".

A similar approach is used in the Boeing "Talk and Draw" project [20], an AWACS workstation that allows users to direct military air operations. However in this case mouse commands are processed first to create a graphical context and then the speech input is parsed from within that context and combined with the graphical input. In this way, if the speech recognizer produces incomplete input, knowledge of the graphical context can be used to prompt the user for the relevant command. “Talk and Draw” also kept a history of referents so that if a use selected an aircraft with a mouse click, they could then ask the question “What is it’s heading?”.

Neal and Shapiro’s CUBRICON multimodal command and control application allowed keyboard, mouse or voice interaction with a map display [Neal et. al. 89]. In contrast to “Talk and Draw”, CUBRICON used gestural information to disambiguate speech input, as well as vice versa. This was possible because the user’s input was interpreted using an Augmented Transition Network grammar of natural language and gestural components. ATN’s are commonly used for natural language parsing. In this case they enhanced a traditional ATN with gestural primitives to integrate the multimodal input into a single representation. Figure 9 shows a sample ATN.

[INSERT SAMPLE ATN HERE]

Figure 9: Multimodal ATN

In the ATNs, each noun phrase can consist of zero or more words along with zero or more pointing references, thus the final representation contains information as to where gestures occurred relative to the speech input. Knowledge sources such as the discourse and user models or domain specific knowledge bases are then used to interpret the ATN and find the referent of the combined multimodal input. CUBRICON also used the ATN representation for multi-media output generation.

The most intuitive multimodal interfaces take advantage of the natural strengths and weaknesses of each input modality. For example, “Talk and Draw” exploits the parallel nature of speech and gesture and enables operators to give spoken commands at the same time as specifying the context with gestures. This was further shown by Weiner and Ganapathy [18] who integrated an isolated word recognizer into a three dimensional CAD package. Speech was used for invoking menu commands, whose magnitude was determined by gesture. This was based on three basic assumptions about gesture and speech:

- Gesture recognition is not as efficient as speech
- A spoken vocabulary has a more standard interpretation than gesture
- Hand gesturing and speech complement each other

They found that adding speech recognition markedly improved the interface, especially because of the sensible partitioning of the voice and gesture semantics. Voice was used for navigating the command menus and gestures for determining the graphics information, meaning the user never had to remove their hands from the modeling surface to issue a command.

Marsh, et. al. add to these recommendations by summarizing several years experience of developing multimodal interfaces (Marsh 1994). They were the primary developers of the NAUTILUS and Eucalyptus command and control map interfaces. Comparing graphical interactions to natural language they found that graphical interactions have the following limitations;

- little relationship between operations,

- commands must be entered sequentially,
- previous operations cannot be referred to,
- the user must follow an arbitrary order of operations.

As a result there are a set of discourse and dialogues properties that should be supported in a multimodal interfaces to significantly enhance their functionality. These are reference, deixis and ellipsis, focus of attention and context maintenance, presupposition, and conversational implicature **[SHOULD WE EXPLAIN THESE FURTHER ?]**. In their work they outline several multimodal interfaces with these features.

Multimodal Integration

A key aspect of multimodal interfaces is input integration. In order to respond to the user's voice and gestural commands the interface needs to integrate the speech and gesture input into a single semantic representation that can be used to generate the appropriate system commands and responses. The general approach is to time stamp the raw input data from each modality, parse it into an intermediate level common semantic form and then use temporal or contextual information to merge related inputs into a single representation. For example, if a user said "Move the boat there", while pointing at a particular location, the speech and gesture input could be represented by the frames shown in Figure 10. Frames are simple slot structures that contain related knowledge in attribute value pairs.

Modality	Speech
Content	MOVE
From	*<boat_location>*
To	
Time	T1

"Move the boat there"

Modality	Gesture
Content	POINT
From	
To	<map_location>
Time	T2

Pointing at Destination

Figure 10: Intermediate Representation of Speech and Gestural Command

If the time at which the pointing gesture is executed, T2, close to the time of the speech command, T1, then it is trivial to merge the two frames into a final multimodal frame shown in Figure 11. Integration is achieved by considering the empty slots in each of the input frames, and which of them must be filled to resolve the spoken command.

Modality	Mixed
Content	MOVE
From	*<boat_location>*
To	<map_location>
Time	T1

Figure 11: Final Multimodal Frame

A slightly more sophisticated approach is that described by Nigay and Coutaz [Nigay 95]. Their system, MATIS (Multimodal Airline Travel Information System), allows users to retrieve travel information using speech, keyboard and mouse, or a combination of these input modalities. They address the problem of fusion of information from different input modalities and present a generic fusion engine that can be embedded in a multi-agent multimodal architecture. Their fusion engine attempts three types of data fusion in order:

- *Microtemporal.* Input information produced at the same time is merged together.
- *Macrotemporal.* Sequential input information is merged together.
- *Contextual.* Input information related by contextual features is merged without regard to temporal constraints.

As Johnston et. al. (1997) point out many multimodal interfaces are speech-driven, so that gestural interpretation is delayed until required to interpret speech expressions. They propose a method for multimodal integration that overcomes this limitation. The basic approach is a unification operation using strongly typed feature structures. The feature structures are semantic representations of the user's speech and gestural input. For a given input several structures will be generated with their associated probabilities. Each of these structures has several associated types which limit the ways in which they can be combined with other structures. Unification refers to the operation of combining two or more consistent structures into a single result. In this case consistency is determined by the types of the structures.

Cohen et. al. (1994) describe an implementation of this technique in their QuickSet interface. This uses pen and voice input for military command and control. For example, if the user says "Barbed wire", the feature structure shown in Figure 12 will be generated. The type of each structure is shown at the bottom left of the structures. While they are speaking, if they make a line gesture with their pen the structures shown in figure X.X might be generated. All the possible interpretations are generated so in this case *point* and *line* type structures are produced.

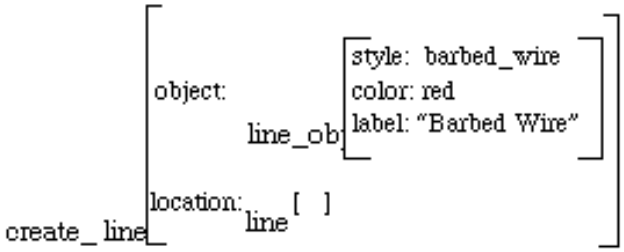

Figure 12: Speech Input Structure

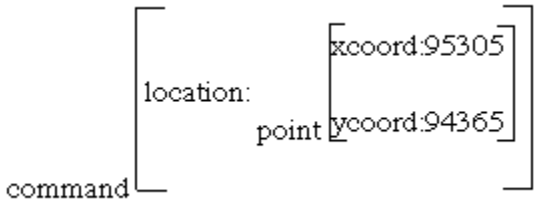

Figure 13a: Point Type Structure

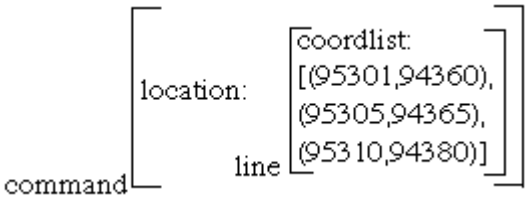

Figure 13b: Line Type Structure

The integration is now guided by tagging of the speech and gesture input as either complete or partial, and examination of the time stamps associated with the multimodal input. In this case the spoken phrase is incomplete and requires a line type structure to fill in the location slot. Thus it become easy to find the correct structure from the gesture input and produce the final integrated result shown in Figure 14. For a given time frame all of the possible gesture and speech input structures are integrated using type consistency and the probability for each result found my multiplying the probability of the component structures together. The integration result with the highest probability is the one selected.

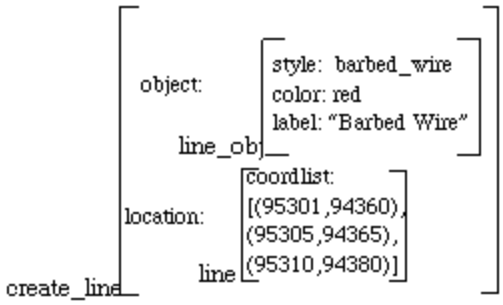

Figure 14: Final integrated frame

The use of typed feature structures has a number of advantages. In contrast to other approaches, this method will work with any gestural input. It also supports partiality, integrating incomplete input structures as shown above. It is impossible to generate results with incompatible types so the input modes can compensate for errors in speech or gesture recognition. For example, users may produce variable gesture input producing a wide range of possible recognition results. However these will almost always be integrated correctly because only those results with the types required by the speech structures will be considered.

Until now the integration methods we have reviewed use time stamping of the user's input at a key guide for integration. However this need not be the case. Vo and Waibel (1997) describe a method for semantic rather than temporal integration. Although temporal integration is easy to implement it forces the user to interact with the interface according to the time constraints assumed by the integration module. For example, gestures may always need to be made before, or within a certain time of the corresponding speech. Similarly, input objects may be combined together purely because they occurred at the same time, rather than having any semantic relationship. Vo and Waibel propose a semantic approach to multimodal understanding that has some similarities to the development of speech grammars for speech recognition applications.

They use a connectionist network to find the most probable multimodal integration result given multiple input streams. Each of these input streams consists of a set of tokens produced by the corresponding speech or gesture recognizer. The input nodes in the network correspond to occurrences of tokens or sets of tokens in the input stream, while the output is the most probable multimodal phrase. The connections between the nodes have weights that correspond to the probabilities of their token given the rest of the tokens in the set. Although this has some similarities to traditional neural networks the weights in the network are automatically generated from a multimodal grammar based input model, eliminating the need to train the network. To support this Vo and Waibel have also developed a visual tool for specifying multimodal input in a grammatical form.

CASE STUDY: Iconic Integration

Although the dominant gestures used in the multimodal systems already discussed are deictic, they need not be so. Bolt and Herranz [1992] have demonstrated a system that combined speech with two handed gestural input, allowing the user to manipulate graphics with semi-iconic gestures. By considering hand rotations their relative positioning the user could intuitively position and orient objects in the virtual world. If the user says "rotate the block like this", while rotating their hands, the block will rotate the same amount as their hands do. Gaze detection was also used to select the intended object.

An even more compelling example is provided by Koons et. al. (1994). **[PUT KOONS WORK HERE – ONE TO TWO PARAGRAPHS]**

Designing Multimodal Interfaces

Grasso et. al. (1998) describe a theory of perceptual structure that can be used to predict performance in multimodal interfaces. This is based on work of Garner (1974) who showed that human performance could be affected by the underlying structure in an input stimulus. Human perception involves a number of processes each with distinct costs. If a person recognized structure in their perceptual stimulus Garner found that some of these costs could be reduced, resulting in an improvement in performance. He defined structural dimensions as either integral or separable; dimensions of a structure are separable if they can be understood in isolation, otherwise they are integral. For example, the multimodal phrase “Put that there” while pointing is integral, while the drawing with a mouse and then speaking a command to change modes is separable.

Grasso et. al. found that the speed, accuracy, and acceptance of multimodal interfaces increase when the modalities match the perceptual structure of the input. This was explored in the context of a multimodal biomedical data collection interface in which histopathologic observations could be entered using a combination of speech and gesture input. Observations consisted of an organ, site, qualifier and morphology, such as *lung alveolus marked inflammation.* Some of these combinations of terms, such as site/qualifier, were assumed to be separable, while others, such as qualifier/morphology, were integral. Subjects used either speech only input, mouse only input, or a combination of modalities to interact with the interface. When using multimodal input they used different combinations of mouse and gesture to input each of the observation terms. Grasso et. al. found that when the input modality matched the perceptual modality users improved task time by 22.5%, speech errors were reduced by 36% and user acceptance increased 6.7%. That is, the speed of multimodal interfaces will increase when the attributes of the task are perceived as separable, while the speed of unimodal interfaces will increase when the attributes of the task are perceived as separable.

These results suggest that matching the input modalities to the perceptual structure of the interface task can produce considerable performance benefits. **[PUT MORE HERE]**

Another factor that should be taken into account is that with multimodal interfaces users may also switch modalities during a task. Oviatt and Olsen identify three factors that influence the user to switch from voice to pen-based gestural input (Oviatt and Olsen 1996):

- *Task Content:* Users are more likely to use pen input for digits and speech for text input.
- *Presentation Format:* Some interfaces such as those using form based entry have a higher percentage of written input, compared to unconstrained interfaces.
- *Contrastive Functionality:* When speakers change conversational functionality they often switch modalities, such as changing between original input and correction, or data and command input.

Similar results are expected for multimodal interfaces that use speech and hand gestures.

[COMPLETE TO HERE – THE FOLLOWING IS NOTES

Lessons Learnt:

importance of context
need for multiple representations
need for common data representation
need to separate data analysis from interpretation

As we have seen from these examples, powerful effects can be achieved by combining speech and gesture recognition with simple context recognition. These results show that:

- Speech and gesture complement each other
- Combined voice and gestural input are preferred over either modality alone.
- Speech should be used for non-graphical command and control tasks
- Gestures should be used for visuo/spatial input, including deictic and iconic.

- Contextual knowledge resolves ambiguous input and increases recognition accuracy
- A large vocabulary is not required and recognition will be improved if it is kept small.

Conversational Systems

Until this point we have described interfaces in which gesture is used to convey content. However in human conversation gesture is also very much involved in moderating the process of conversation, these are typically *beat gestures* and *cohesives* (McNeill REF). For example, a speaker will gaze at listener to indicate they can take the turn, a person will pause in mid gesture as they collect their thoughts, or a speaker will make unconscious beat gestures in time with the rhythm of their speech. In this section we discuss gesture in the context of conversational interfaces. These are the gestures which occur at the lower end of the Kendon Continuum.

The notion of the conversational interface was introduced nearly thirty years ago by Nicholas Negroponte [1]. He imagined a machine that humans could interact with the same way they do with each other on a daily basis; using voice, gaze, gesture and body language However conversational interfaces go beyond integrating speech and gesture at the desktop. As Richard Bolt points out “conversation is speaking back and forth, pointing out this or that - a lively dialogue that involves glancing about and following the other person’s glances as well as using gestures to describe, indicate and emphasize. Whether the topic is trivial or weighty, conversation means a strong sense of another’s presence in a setting we both share.” [Bolt 1992] Conversational interfaces are by definition interfaces that allow this type of relationship to exist between human and computer. Speech enabled interfaces are not necessarily conversational interfaces, because speech recognition is not conversational understanding. Similarly gestural input alone is not sufficient. Rather conversation interfaces require multimodal input and output controlled by an intelligent system that is a conversational expert.

An important distinction between conversational and multimodal interfaces is the way in which the speech and gestural input is understood. In human conversation conversants actively collaborate together to achieve mutual understanding in a process known as “grounding” [Clark and Brennan 90]. In order to achieve this the audio/visual contributions to conversation contain both *propositional* and *interactional* information. The propositional information consists of the conversational content, while the interactional information consists of cues that affect the conversational process. The gestural and multimodal interfaces described earlier try to achieve propositional understanding of the user’s input. The Media Room couldn’t detect sarcasm in the user’s voice or care if they were shouting or whispering at the computer. What distinguishes conversational interfaces from other multimodal interfaces is their attempt to understand interactional cues. These include speech cues such as the presence or absence of speech, explicit words (“O.K.”, “What did you say?”), pauses, paraverbals (“huh?”, “Uh-huh!”), and prosodics or intonation (pitch contour, syllable duration, relative energy levels). They also include visual cues such as presence or absence of a conversant, facial or body gestures and eye contact and gaze direction.

The Motivation for Conversational Interfaces

As can be imagined developing a computer that can understand the nuances of human conversation is a hugely challenging task. However, there are a number of strong motivations for undertaking this task:

Intuitiveness. Conversation is an intrinsically human skill that is learned over years of development and is practiced daily. Conversational interfaces provide an intuitive paradigm for interaction, since the user is not required to learn new skills.

Redundancy and Modality Switching: Embodied conversational interfaces support redundancy and complementarity between input modes. This allows the user and system to increase reliability by conveying information in more than one modality, and to increase expressiveness by using each modality for the type of expression it is most suited to.

The Social Nature of the Interaction. Whether or not computers look human, people attribute to them human-like properties such as friendliness, or cooperativeness [21]. An embodied conversational interface can take advantage of this and prompt the user to naturally engage the computer in human-like conversation. If the interface is well-designed to reply to such conversation, the interaction may be improved

Increase in Bandwidth: Combining conversational cues at an interface allows an increase in input and output bandwidth. The bandwidth limitations of current direct manipulation interfaces are made apparent by considering the interface model that the computer has of the user; a one eyed, one handed, deaf person!

These motivations are from an interface perspective, however there are also strong reasons for developing conversational agents from a communications perspective. Anthropologists, psychologists and linguists have developed many theories of communication, from a psycho-linguistic rather than computational perspective. Developing computation models of reasoning work in the Artificial Intelligence community contributed significantly to psychological theories of cognition. In much the same way, developing a computational model of conversation will benefit the linguistic community by giving them a computation approach for testing their theories.

Interface Requirements

In attempting to develop a conversational interface there are a number of interface requirements. These can be divided into two groups; requirements on the interface of the conversational system, and requirements on the underlying architecture.

Considering human face to face conversation it is obvious that a conversational system must be able to recognize and respond to verbal and non-verbal input and be able to generate verbal and non-verbal output. This necessitates some form of graphical representation, both to display non-verbal cues as well as to act as a point of reference for the many spatial cues that occur in conversation. Without an embodied interface it is impossible to look your computer in the eye to tell it to take the turn in the conversation !

In terms of underlying architectural requirements, Thorisson reviews the behavioral and psychological literature to arrive at the following (Thorison 1996):

- *Incremental Interpretation:* Conversational understanding is not done “batch style”, but occurs continuously as new input is received.

- *Multiple Data Types:* Conversational interfaces must understand and generate multiple input data types such as gesture, gaze, speech and prosody, and also internal representations such as spatial and relational.

- *Seamlessness:* Different data types are inserted seamlessly into the conversation, as are the cues that cause conversational mechanisms such as turn taking.

- *Temporal Variability:* Conversational input, out and understanding occurs over a variety of time scales ranging from less than a second in the case of recognizing intonation to many minutes for discourse understanding.

- *Multi-Layered Input Analysis and Response Generation:* In order to understand input on a variety of time scales and generate matching output, conversation require multiple layers the work at different time scales.

- *High Level Understanding:* Conversation involves more than understanding at the word or sentence level. Participants also take into account the conversational context and high level cues from the on-going discourse.

Perhaps the most important requirement is that ability to map input and output behaviors to conversational function. Unlike the previous multimodal interfaces, in a conversational interface it is not enough just to map behaviors directly to commands. Instead, the emphasis must be on mapping behaviors to the higher level discourse functions they are attempting to convey. Typical conversational discourse functions include *conversation invitation*, *turn taking*, *providing feedback, contrast and emphasis*, and *breaking away* [10][14]. Just as in face to face conversation each of these functions can be realized in different ways. For example, a person could indicate they wanted to take turn in a conversation by interrupting the current speaker and starting to speak, or a more polite person could indicate the same with a glance and eye contact. This mapping between behavior and discourse function occurs both on the input side in interpreting a person's behavior and on the output side in deciding which behaviors to use to represent a particular discourse function. Input events in different modalities may be mapped onto the same discourse function, while in different conversational states the same function may lead to different conversational behaviors, based on conversational state and the availability of input and output modalities.

Example Interfaces

The first attempts at conversational systems focused on one or more aspects of human face to face conversation, rather than attempting to built entire conversational systems. Researchers such as Ball et. al. (Ball 1997) and Beskow et. al. (Beskow 1997) develop embodied conversational which reacted to speech input only. These agents typically integrated spoken language input, a conversational dialogue manager, reactive 3D animation, and recorded speech output. Although they ignored the human's gestures the graphical characters displayed many gestures to convey conversational cues. For example, Ball's character was the parrot shown inFigure 15. It is able to display conversational understanding using gross "wing gestures" (such as cupping a wing to one ear when the parrot has not understood a user's request) and facial displays (scrunched brows as the parrot finds an answer to a question). Similarly other researchers have used human-like avatars to convey more natural conversational gestures. Noma & Badler have created a virtual human weatherman, based on the *Jack* human figure animation system [19]. While discussing the climate the weatherman displays presentation pre-recorded gestures culled from books on public speaking. However in these systems there was little attempt to interpret the users input in terms of conversational function or generate conversational output behavior based on the desired discourse function.

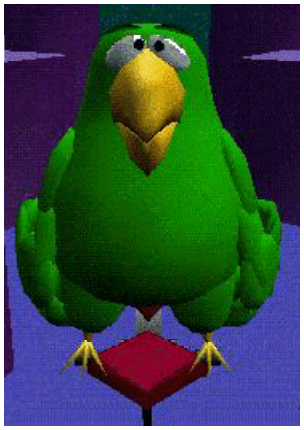

Figure 15: Peedy the Parrot.

The value of building embodied agents that reacted to user gestures was shown by the ALIVE interface developed by Pattie Maes (Maes et. al. 1995). ALIVE is a gesture and full-

body recognition interface that allows users to interact with autonomous agents in a virtual environment. The users stand in front of a large screen projection panel on which is displayed a virtual environment. Video cameras capture their image, so they also see themselves within the virtual world in a so-called "Magic Mirror" approach to virtual reality. Computer vision techniques were also used to recognize the users body position and simple gestures such as pointing, and petting.

By varying gesture and body position the user can interact with the intelligent agents in the virtual world in a believable manner. For example, one of the environments contains a Hamster who responds to the user petting it or offering it virtual food. The agents in the world respond according to their own internal states and past history - if hungry the Hamster will respond readily to food, but after a while it will become "full" and move away disinterested. The contextual states of the agents are also used to guide the image processing system doing the gesture recognition. If the Hamster is hungry then the image processor will focus on the users hands to see if they are reaching for the virtual food. Using contextual cues means that the gesture employed by the user can have rich meaning which varies on the previous history, the agents internal state and the current situation.

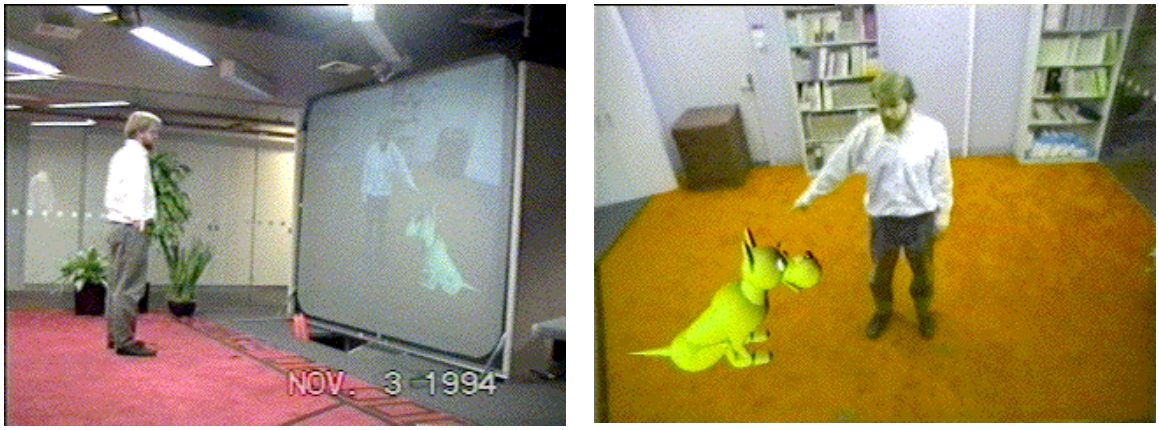

Figure 16: The ALIVE System

Although the agents in ALIVE did not respond to voice commands users found the interface very intuitive because they could use simple gestures which were natural for the given domain. Faced with embodied agents, users immediately adopted the implicit protocol of inter-human gestural communication and had no difficulty in interacting with them. They were also tolerant of imperfections in the gesture recognition, attributing mistakes or lag time to the agent trying to make up its mind or some other human-like quality. For many users, the presence of intelligent reactive agents made the virtual environment more compelling and the interactions more interesting than traditional virtual reality.

Thorisson provides a good example of an early fully integrated conversational multimodal interface with both voice and gestural understanding (Thorisson 1996). In his work the main emphasis was the development of an architecture that could support fluid face-to-face dialog between a human and graphical agent. He found that a multi-layer multimodal architecture could exhibit conversational behavior at a number of different levels and time scales, from low-level non verbal reactive behaviors to high level content based reflective behaviors. The agent, Gandalf, was used to command a graphical model of the solar system in an educational application. The agent and solar system model were shown on different displays, enabling the agent to gaze and gesture toward the user or the task space. This required the agent to have

understanding of real-world spatial locations. Unlike previous systems Gandalf recognized both verbal and non-verbal cues and could interact with people using voice, gesture and facial expression. Thus Gandalf could react to and generate both proposition information and interactional information.

Figure 17: A User Interacting with Gandalf

Thorisson identified a number of issues in developing conversational interfaces, including the need for effective low-level behaviors, and accurate functional analysis. In user studies he found that low-level non-verbal behaviors such as gaze, gesture and eye blinking made the agent more believable than higher level emotional expressions. With good low-level behaviors users would talk to the agent directly rather than focus on the task space. Functional analysis is the problem of identifying the function of a gesture, or other non-verbal input. If the user makes a pointing gesture are they pointing, showing an iconic representation of an object or neither? Thorisson does not provide a general solution to this problem, but emphasizes the need for a combined top-down and bottom-up approach. In general these interfaces show that conversational interfaces should have a multi-layered architecture that uses both a top-down and bottom up approach for multimodal understanding.

Although the work of Thorisson provides a good first example of how discourse and non-verbal function might be paired in a conversational interface, there were a number of limitations. The interface required the user to wear instrumented gloves, an eye tracker and a body tracker, encumbering them and preventing natural gestural input. More importantly Gandalf had limited ability to recognize and generate propositional information, such as providing correct intonation for speech emphasis on speech output, or a co-occurring gesture with speech. [MORE HERE ABOUT GANDALF USE STUDIES]

In contrast, “Animated Conversation” (Cassell et. al. 1994) was a system that automatically generated context-appropriate gestures, facial movements and intonational patterns. In this case the domain was conversation between two artificial agents and the emphasis was on the production of non-verbal propositional behaviors that emphasized and reinforced the content of speech. [MORE ABOUT ANIMATED CONVERSATIONS]

Figure 18: Animated Conversations

Since there was no interaction with a real user, the interactional information was very limited. [MORE HERE]

CASE STUDY: REA

Cassell's work on Rea is an attempt to develop an agent with both propositional and interactional understanding and generation, which can interact with the user in real time (Cassell et al. 1999). As such it combines elements of the Gandalf and the Animated Conversations projects into a single interface and moves towards overcoming the limitations of each. Rea is a computer generated humanoid that has a fully articulated graphical body, can sense the user passively through cameras and audio input, and is capable of speech with intonation, facial display, and gestural output. The system currently consists of a large projection screen on which Rea is displayed and which the user stands in front of. Two cameras mounted on top of the projection screen track the user's head and hand positions in space using the STIVE vision software (Pentland Ref). Users wear a microphone for capturing speech input. A single SGI Octane computer runs the graphics and conversation engine of Rea, while several other computers manage the speech recognition and generation and image processing (Figure 19).

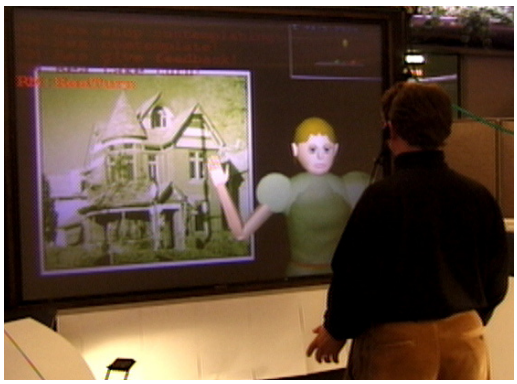

Figure 19: User Interacting with Rea

Rea's domain of expertise is real estate and she acts as a real estate agent showing users the features of various models of houses that appear on-screen behind it. The following is a excerpt from a sample interaction:

Lee approaches the projection screen. Rea is currently turned side on and is idly gazing about. As the user moves within range of the cameras, Rea turns to face him and says "Hello, my name is Rea, what's your name?"
"Lee"
"Hello Lee would you like to see a house?" Rea says with rising intonation at the end of the question.
"That would be great"

A picture of a house appears on-screen behind Rea.
"This is a nice Victorian on a large lot" Rea says gesturing towards the house. "It has two bedrooms and a large kitchen with.."
"Wait, tell me about the bedrooms" Lee says interrupting Rea by looking at Rea and gesturing with his hands while speaking.
"The master bedroom is furnished with a four poster bed, while the smaller room could be used for a children's bedroom or guest room. Do you want to see the master bedroom?".
"Sure, show me the master bedroom". Lee says, overlapping with Rea.
"I'm sorry, I didn't quite catch that, can you please repeat what you said", Rea says.

And the house tour continues…

As can be seen from this example, Rea is able to conduct a mixed initiative conversation, describing the features of the house while also responding to the users' verbal and non-verbal input. When the user makes cues typically associated with turn taking behavior such as gesturing, Rea allows herself to be interrupted, and then takes the turn again when she is able. She is able to initiate conversational repair when she misunderstands what the user says, and can generate combined voice and gestural output. In order to carry on natural conversation of this sort, Rea uses a conversational model that supports multimodal input and output as constituents of conversational functions. That is, input and output is interpreted and generated based on the discourse functions it serves.

A key aspect of the REA is the mapping of users gestural and verbal input to conversational functions. This mapping depends on both the current state of the conversation and the user's input. For turn taking, for example, the specifics are summarized in Table 3. If Rea has the turn and is speaking and the user begins to gesture, this is interpreted as the user *wanting turn* function, or if the user is speaking and s/he pauses for less than half a second this is interpreted as the *wanting feedback* function.

State	User Input	Input Function
Rea speaking	Gesture	Wanting turn
	Speech	Taking turn
User speaking	Pause of <500 msec.	Wanting feedback
	Imperative phrase	Giving turn
	Interrogative phrase	Giving turn
	Declarative phrase & pause >500 msec. & no gesture	Giving turn
	Declarative phrase & long gesture or pause	Holding turn

Table 3. ***Functional interpretation of turn taking input***

Thus, user gesture or speech may convey different interactional information; it may be interpreted as taking turn, giving turn, or holding turn depending on the conversational state and what is conveyed by the other modalities.

A similar approach is taken for the realization of Rea's desired conversational functions as output behaviors. Rea generates speech, gesture and facial expressions based on the current conversational state and the conversational function she is trying to convey, as shown in Table 4. For example, when the user first approaches Rea ("User Present" state), she signals her openness to engage in conversation by looking at the user, smiling, and/or tossing her head, and when the user is speaking and Rea wants the turn she looks at the user and utters a paraverbal ("umm").

State	Conversational Function	Output Behaviors
User Present	Open interaction	Look at user. Smile. Headtoss.
	Attend	Face user.

	End of interaction	Turn away.
	Greet	Wave, “hello”
Rea Speaking	Give turn	Relax hands. Look at user. Raise eyebrows
	Signoff	Wave. “bye”
User Speaking	Give feedback	Nod head Paraverbal
	Want turn.	Look at user. Raise hands. Paraverbal(“umm”).
	Take turn.	Look at user. Raise hands to begin gesturing. Speak.

Table 4*. Output Functions*

By modeling behavioral categories as discourse functions we have developed a natural and principled way of combining multiple modalities, in both input and output. Thus when REA decides to take the turn, for example, she can choose any of several behaviors based on what is appropriate at the moment.

To facilitate this conversational mapping the underlying architecture of Rea is extremely modular, enabling each of the conversational functions to be carried out by a different component. Figure 20 shows the internal architecture of Rea. The three points that differentiate Rea from other embodied conversational agents are mirrored in the organization of the system architecture:

- Input is accepted from as many modalities as there are input devices. However all the different modalities are integrated into a single semantic representation that is passed from module to module.
- The semantic representation has slots for interactional and propositional information so that the regulatory and content-oriented contribution of every conversational act can be maintained throughout the system.
- The categorization of behaviors in terms of their conversational functions is mirrored by the organization of the architecture which centralizes decisions made in terms of functions (the understanding, response planner, and generation modules), and moves to the periphery decisions made in terms of behaviors (the input manager and action scheduler).

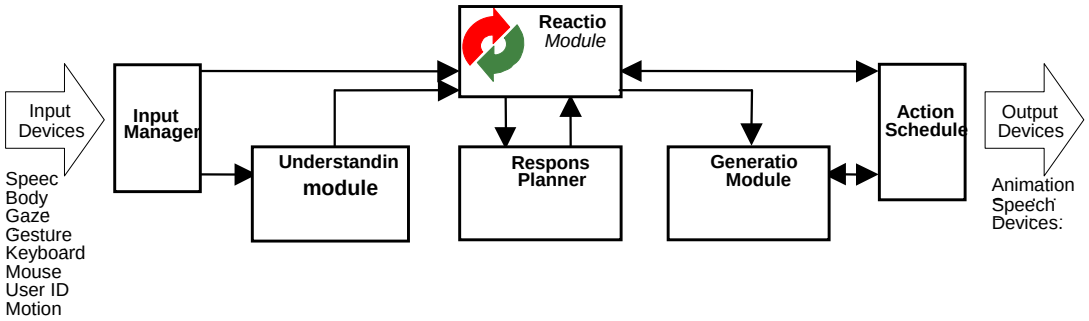

Figure 20: The Rea Software Architecture

LESSONS LEARNED FROM CONVERSATIONAL SYSTEMS
[PUT MORE HERE]

Future Research Directions

[INTRO STUFF]
In the previous section we discussed conversational interfaces based around embodied agents. However many of the same underlying technologies can be used to enable users to interact with the environment as a whole. Naturally science fiction writers have foreseen this decades ago and in movies such as “Star Trek” it is common for characters to shout commands into space and have a computer automatically interpret them. On a more sinister level Arthur C. Clark’s computer HAL in “2001” is an omnipresent entity forever watching and listening to the crew of the space ship **[BLAH]** through it’s many camera and microphones. Embedding vision and audio sensors into physical space in this way enables the creation of reactive environments that automatically respond to user’s speech and gesture.

Bolt incorporated gaze detection to further increase the intimacy of the interface.[12] Now the computer was able to respond to implicit cues as well as the explicit cues of speech and gesture. Experiments by Argyle[13] and others[14] have shown the gaze patterns used differ markedly according the goal of the observer. So even rudimentary gaze recognition adds power to the interface. As an example of this Bolt used the Media Room to design an interface called "Gaze-Orchestrated Dynamic Windows". [15] This involved showing up to thirty different moving images at once on the wall display, complete with a cacophony of their soundtracks all combined into one. Gaze detection was then used to change the relative size of each of the moving images. If the users gaze was fixed on a particular channel all the soundtracks to the other channels were turned off. If they continued watching, the image would zoom to fill the wall display. The net effect is that the computer filters out all but the information of immediate interest to the user, just as humans do when engaged in conversation. Gaze tracking was done by shining infrared light into the users eye and using an infrared video camera to look for reflections.

Conclusions

NOTES – TO BE INSERTED

Bremmer, J. & Roodenburg, H. (Eds.)(1991). *A Cultural History of Gesture.* Ithica: Cornell University Press.

[Etoh, Tomono, Kishino 1994]

Fels, S. S. and Hinton, G. E. (1993). *Glove-Talk: A Neural Network Interface Between a Data-gloveand a Speech Synthesizer.* IEEE Transactions on Neural Networks, 4, 2-8.

Fels, S. & Hinton, G. (1995). Glove-Talk II: An adaptive gesture-to-formant interface. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI’95)*, 456-463.

Garner, W.(1974). *The Processing of Information and Structure*. Potomac, Maryland: Lawrence Erlbaum.

Goldin-Meadow, S. (2003). *Hearing Gesture: How our Hands Help us Think.* Cambridge MA: Belknap Press.

Grasso, M., Ebert, D. & Finn, T. (to appear) The Integrality of Speech in Multimodal Interfaces. Submitted for publication in ACM Transactions on Computer-Human Interaction.

Hauptmann, A.G. & McAvinney, P. (1993). Gestures with Speech for Graphics Manipulation*. Intl. J. Man-Machine Studies*, 38, 231-249.

Johnston, M., Cohen, P., McGee, D., Oviatt, S., Pittman, J., Smith, I. (1997) Unification-based Multimodal Integration. **XXXXXXX**

[Kendon 1988]

Kendon, A. (1990) The negotiation of context in face-to-face interaction. In A. Duranti and C. Goodwin (eds.), *Rethinking context: language as interactive phenomenon*. Cambridge University Press. NY.

Kendon, A. (2004). *Gesture: Visible Action as Utterance*. Cambridge: Cambridge University Press.

Kinsbourne, M. & Hicks, R.E. (1978). Functional Cerebral Space; a Model for Overflow, Transfer and Interference Effects in Human Performance: a Tutorial Review. In J. Requin (Ed). *Attenion and Performance VII*.,. Hillsdale, NJ: Erlbaum, **XX-XX.**

Kobayashi, T. & Haruyama, S. (1997). Partly Hidden Markov Model and its Application to Gesture Recognition, *IEEE Proc. ICASSP97, Vol.VI*, 3081-3084.

Kramer, J. & Larry Leifer, L. (1989). The Talking Glove: A speaking aid for non-vocal deaf and deaf-blind individuals. *Proceedings of RESNA 12th Annual Conference,* 471--472.

Kramer, J. & Larry Leifer, L. (1990). A ``Talking Glove'' for nonverbal deaf individuals*. Technical Report CDR TR 1990 0312*, Centre For Design Research, Stanford University.

Kramer. J. (1991). Communication system for deaf, deaf-blind and non-vocal individuals using instrumented gloves. US Patent 5,047,952, Virtual Technologies.

[Krueger 1991]

Kurtenbach, G. & Hulteen, E. (1990). Gestures in Human-Computer Communications. In B. Laurel (Ed*.) The Art of Human Computer Interface Design*. Addison-Wesley, 309-317.

Lee, C. & Xu, Y. (1996). Online, Interactive Learning of Gestures for Human/Robot Interfaces. *1996 IEEE International Conference on Robotics and Automation,* vol. 4, 2982-2987.

Lee, J., Kunni, T. (1993) Constraint-based hand modelling and tracking. In **X. XX** (Ed*.). Models and Techniques in Computer Animation*, Tokyo: Springer-Verlag, 110-127.

MacKenzie, C.L. & Iberall, T. (1994). *The Grasping Hand.* Amsterdam: North-Holland.

Maes, P., Darrell, T., Blumberg, B., and Pentland, S. (1995*) The ALIVE System: Full-Body Interaction with Autonomous Agents.* In proceedings of the Computer Animation '95 Conference, Geneva, Switzerland, pp. 11-18, IEEE Press, April 1995.

McNeill, D. (1985) So you think gestures are nonverbal? *Psychological Review.* 92, 350-371.

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

[Marsh 1994]

Martin, G.L. (1989). The Utility of Speech Input in User-Computing Interfaces. *Intl. J. Man-Machine Studies*, 30, 355-375

Martin, J.& Crowley, J. (1997). An Appearance-Based Approach to Gesture-Recognition. **XXX**

McNeill, D. (1985) So you think gestures are nonverbal? Psychological Review. 92, 350-371.

McNeill, D. (1992). Hand and Mind: What Gestures Reveal About Tought. Chicago: University of Chicago Press.

McNeill, D. (2005). Gesture & Thought. Chicago: University of Chicago Press.

Morris, D. (1994). *Bodytalk: The Meanding of Human Gestures. New York: Crown Publishers.*

Morris, D., Collett, P., Marsh, P. & O'Shaughnessy, M. (1979). *Gestures*. New York: Stein and Day

[Morris, Collet, Marsh 1980]

Mulder, A (1996). Hand Gestures for HCI. *Hand Centered Studies of Human Movement Project Technical Report 96-1* School of Kinesiology, Simon Fraser University.

[Mur 1991]

[Neal 1989]

Negroponte, N. The Architecture Machine. Cambridge: MIT, 1970.

Oviatt. S. (1996). Multimodal Interfaces for Dynamic Interactive Maps, *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'96)*, 95-102.

Pavlovic, V.I., Sharma, R. & Huang, T.S. (1995). Visual interpretation of hand gestures for human-computer interaction: A review, *Technical Report UIUC-BI-AI-RCV-95-10*, University of Illinois at Urbana-Champaign.

Rehg, J & Kanade. T. (1994). Digiteyes: Vision-based hand tracking for human computer interaction. In J. Aggarwal and T. Huang (Eds.). *Proc. of Workshop on Motion of Non-Rigid and Articulated Objects*, IEEE Computer Society Press, 16-22

Rime, B. (1982) The elimination of visible behaviour from social interactions: Effects on verbal, nonverbal and interpersonal variables. *European Journal of Social Psychology* 12: 113-29.

[Rime and Schiaratura 1991]

Saffer, Dan (2009). *Designing Gestural Interfaces.* Sebastapool, CA: O'Reilly.

Salisbury, M.W., Hendrickson, J.H., Lammers, T.L., Fu, C. & Moody, S.A. (1990). Talk and Draw: Bundling Speech and Graphics. *IEEE Computer*, 23(8), 59-65.

Siegel, I. (1998). *All About Bone: An Owner's Manual.* New York: Demos Medical Publishing.

Starner, T., Pentland, A. (1995) *Visual Recognition of American Sign Language Using Hidden Markov Models.* International Workshop on Automatic Face and Gesture Recognition (IWAFGR) 1995 Zurich, Switzerland.

Sturman, D. J. (1992). *Whole Hand Input.* PhD Thesis. Cambridge, MA: Massachusetts Institute of Technology.

Sturman, D. J. and Zeltzer, D. (1993). *A Design Method For "Whole-Hand" Human-Computer Interaction.* ACM Transactions on Information Systems, 11(3), 219-238.

Sturman, D. & Zeltzer, D. (1994) A Survey of Glove-Based Input. *IEEE Computer Graphics and Applications*, 14, 30-39.

Thórisson, K. R. (1996). *Communicative Humanoids: A Computational Model of Psychosocial Dialogue Skills.* PhD Thesis, MIT Media Laboratory.

Treisman, A. & Davies, A. (1973). Divided Attention to Ear and Eye. In S. Kornblum (Ed *). Attention and Performance IV.* Hillsdale, NJ: Erlbaum , 101-117.

Trumble, A. (2010). *The Finger: A Handbook.* New York: Farrar, Straus and Giroux.

Väänänen, K.. & Böhm, K. (1993). Gesture Driven Interaction as a Human Factor in Virtual Environments - An Approach with Neural Networks. In R. Earnshaw, M. Gigante & H. Jones (Eds.). *Virtual Reality Systems*. New York: Academic Press, **XX-XX**.

Vo, M. & Waibel, A. (1997). Modeling and Interpreting Multimodal Inputs: A Semantic Integration Approach*. Technical Report CMU-CS-97-192*, School of Computer Science, Carnegie Mellon University.

Wachs, Y.P., Kölsch, M., Stern, H. & Edan, Y. (2011). Vision-based hand-gesture applications. *Communications of the Association of Computing Machinery (CACM*), 54(2), 60-71.

Weiner, D. & Ganapathy, S.K. (1989). A Synthetic Visual Environment with Hand Gesturing and Voice Input. . *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'89)*, 235-240.

Wexelbalt, A. (1995). An Approach to Natural Gesture in Virtual Environments. *ACM Transactions on Computer-Human Interaction (TOCHI),* 2(3), 179-200

Wickens, C.D. (1980). The Structure of Attentional Resources. In R. Nickerson & R. Pew (Eds.). *Attention and Performance VIII*. Hillsdale, NJ: Erlbaum, **XX-XX**.

Wilson, F.W. (1998). *The Hand: How its use Shapes the Brain, Language, and Human Culture*. New York: Pantheon Books

Wise, S.& **XX** (1990). Evaluation of a Fiber Optic Glove for Semi-automated Goniometric Measurements*. J. Rehabilitation Research and Development,* 27(4), 411-424.

[Zacharey]

Zimmerman, T., Lanier, J., Blanchard, C., Bryson, S. and Harvil, Y. (1987). A Hand Gesture Interface Device. In Proceedings of CHI 87 and GI, (pp. 189-192): ACM.

Draft References

[3] Leggett, J. & Williams, G. (1984). An Empirical Investigation of Voice as an Input Modality for Computer Programming. *Intl. J. Man-Machine Studies*, 21, 493-520.

[4] Poock, G.K. (1982). Voice Recognition Boosts Command Terminal Throughput. *Speech Technology*, 1, 36-39.

[5] Cochran, D.J., Riley, M.W. & Stewart, L.A. (1980). An Evaluation of the Strengths, Weaknesses and Uses of Voice Input Devices. *Proceedings of the Human Factors Society - 24th Annual Meeting*. Los Angeles. **XX-XX**.

[10] Rohr, G. (1986). Using Visual Concepts. In S. Chang, T. Ichikawa, & P. Ligomenides (Eds.). *Visual Languages*, New York: Plenum Press, **XX-XX**.

[11] Schmandt, C., Ackerman, M.S. & Hindus, D. (1990). Augmenting a Window System with Speech Input*. IEEE Computer,* 23(8), 50-56.

[12] Peacocke, R.D. & Graf, D.H. (1990). An Introduction to Speech and Speaker Recognition. *IEEE Computer*, 23(8), 26-33.

[16] Ford, W.R., Weeks, G.D. & Chapanis, A. (1980). The Effect of Self-Imposed Brevity on the Structure of Dyadic Communication. *Journal of Psychology*, 104, 87-103.

[17] Kay, P. (1993) Speech Driven Graphics: a User Interface. *Journal of Microcomputer Applications* , 16, 223-231.

??????[19] VIEW reference **what is the reference?**

[21] Jones, D., Hopeshi, K, & Frankish, C. (1989). Design Guidelines for Speech Recognition Interfaces*. Applied Ergonomics* , 20(1), 47-52.

???[22] Savage-Carmona, J. & Holden, A. (1994). A Hybrid System with Symbolic AI and Statistical Methods for Speech Recognition. Submitted to VRAIS '95. **This must have appeared by now. Reference?**

Ali, S.. & McRoy, S. (1998). Efficient Representations for Multi-Modal Interaction.**XXXXXXXXX**

McRoy, S., Haller, S., Ali, S. Uniform Knowledge Representation for NLP in the B2 system. *Journal of Natural Language Engineering*, 3(2), XX-XX..

To add:

Goldin-Meadow, S. (2003). *Hearing Gesture: How our Hands Help us Think.*Cambridge MA: Belknap Press.

Kendon, A. (2004). *Gesture: Visible Action as Utterance*. Cambridge: Cambridge University Press.

Wolf, W., Ozer, B., & Lv, T. (2002). Smart Cameras as Embedded Systems. *IEEE Computer,* Spetember 2002, 35(9), 48-53.

Kölsch, M. Turk, M. & T. Höllerer. T. (2004). Vision-based interfaces for mobility. In Proc. MobiQuitous '04 (1st IEEE Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services), 86-94.

Also consider:

Kessler, G.D., Hodges, L.F. & Walker, N. (1995). Evaluation of the CyberGlove as a whole-hand input device. *ACM Transactions on Computer-Human Interaction (TOCHI),* 2(4), 263-283.

See following review:

Pavlovic, V., Sharma, R., Huang, T. (1997). Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), 677-695.

CSCW stuff around my notion of “Reference Space” (in contrast to person space or task space), including:

Kirk, David S. (2006). *Turn it <u>This</u> Way: Remote Gesturing in Video-Mediated Communication*. PhD Thesis. University of Nottingham.

Kirk, David S, Crabtree, Andy, & Rodden, Tom (2005). Ways of the Hand. *Proceedings of the 9th European Conference on Computer Supported Cooperative Work (ECSCW05)*, 1-21.

Kirk, David S. & Fraser, Danaë Stanton (2005). The Effects of Remote Gesturing on Distance Instruction. *Proceedings of ECSCW 2005*, 301-310.

Kirk, David S. & Fraser, Danaë Stanton (2006). Comparing Remote Gesture Technologies for Supporting Collaborative Physical Tasks. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI’06),* 1191-1200.

Kirk, David S., Rodden, Tom & Fraser Danaë Stanton (2007). Turn it <u>This</u> Way: Grounding Collaborative Action with Remote Gestures. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI’07),*1039-1048.

Tuddenham, Philip (2007). Distributed Tabletops: Territoriality and Orientation in Distributed Collaboration. Conference on Human Factors in Computing Systems CHI '07 - Extended Abstracts on Human Factors in Computing Systems, 2237-2242.

Tuddenham, Philip & Robinson, Peter (in press). Territorial Coordination and Workspace Awareness in Remote Tabletop Collaboration. To appear in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI’09).*

Gesture Books:

Goldin-Meadow, S. (2003). *Hearing Gesture: How our Hands Help us Think.* Cambridge MA: Belknap Press

Kendon, A. (2004). *Gesture: Visible Action as Utterance*. Cambridge: Cambridge University Press.

MacKenzie, C.L. & Iberall, T. (1994). *The Grasping Hand.* Amsterdam: North-Holland.

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

McNeill, D. (2005). *Gesture & Thought.* Chicago: University of Chicago Press.

Morris, D. (1994). *Body Talk: The Meaning of Human Gestures.* New York: Crown Trade Paperbacks.

Morris, D. Collett, P., Marsh, P. & O'Shaughnessy, M. (1979). *Gestures*. New York: Stein and Day.

Wilson, F.W. (1998). *The Hand: How its use Shapes the Brain, Language, and Human Culture*. New York: Pantheon Books.