Mutual Disambiguation of 3D Multimodal Interaction in Augmented and Virtual Reality

Ed Kaiser[1] Alex Olwal[2] David McGee[3] Hrvoje Benko[2] Andrea Corradini[1] Xiaoguang Li[1] Phil Cohen[1] Steven Feiner[2]

[1]Oregon Health and Science University
OGI School of Science & Eng.
20000 NW Walker Road
Beaverton, OR 97006, USA
+1 503 748 7803
{kaiser,andrea,xiaoli,pcohen}@cse.ogi.edu

[2]Columbia University
Department of Computer Science
1214 Amsterdam Avenue
New York, NY 10027, USA
+1 212 939 7000
{aolwal,benko,feiner}@cs.columbia.edu

[3]Pacific Northwest National Laboratory
Rich Interaction Environments
906 Battelle Boulevard
Richland, Washington 99352, USA
+1 503 293 8414*
dmcgee@naturalinteraction.com*

ABSTRACT

We describe an approach to 3D multimodal interaction in immersive augmented and virtual reality environments that accounts for the uncertain nature of the information sources. The resulting multimodal system fuses symbolic and statistical information from a set of 3D gesture, spoken language, and referential agents. The referential agents employ visible or invisible volumes that can be attached to 3D trackers in the environment, and which use a time-stamped history of the objects that intersect them to derive statistics for ranking potential referents. We discuss the means by which the system supports mutual disambiguation of these modalities and information sources, and show through a user study how mutual disambiguation accounts for over 45% of the successful 3D multimodal interpretations. An accompanying video demonstrates the system in action.

Categories

H.5.1 (Multimedia Information Systems): Artificial, augmented, and virtual realities; H.5.2 (User Interfaces): Graphical user interfaces, natural language, voice I/O; I.2.7 (Natural Language Processing); I.2.11 (Distributed Artificial Intelligence): Multiagent systems; I.3.7 (Three-Dimensional Graphics and Realism): Virtual reality

General Terms

Measurement, Human Factors

Keywords

Multimodal interaction, augmented/virtual reality, evaluation.

1. INTRODUCTION

Techniques for interacting in 3D worlds are usually derived from the direct manipulation metaphor—in order to perform an operation on something, you have to "touch" it. This style of interaction works well when the objects to be manipulated are known and at hand and the means for selecting objects and other actions are relatively straightforward. Unfortunately, 3D interaction often breaks all of these rules—for example, the objects of interest may be unknown or at a distance. To cope with these problems, some researchers have taken the direct manipulation style of interaction to extremes, creating devices with many buttons and modes [8], arbitrarily stretchable "arms" [25], and 3D menus [18]. However, there may be far more possible actions that can be performed on objects than such GUIs can realistically provide. We argue that most prior approaches have placed too much functionality on too impoverished a communications channel (3D arm/hand motions), and that by incorporating multimodal interaction, the burden of various interactive functions can be off-loaded to appropriate modalities, such as speech and gesture, in a synergistic fashion. In particular, by incorporating speech into the interface, the user could describe unseen/unknown objects and locations or invoke functions, while her hands and eyes may be engaged in some other task.

However, unlike direct manipulation interfaces, multimodal interface architectures must cope first and foremost with uncertainty. Recognizers return a set of classification hypotheses, each of which is assigned a score, such as a posterior probability. Moreover, language is ambiguous, and thus even a single correctly recognized utterance can lead to multiple hypotheses. Likewise, trackers have errors, gestures are uncertain, their meanings are ambiguous, and a correct gesture (e.g., selection) can have multiple interpretations (e.g., what is being selected). Given all this uncertainty, it is perhaps surprising that few, if any, multimodal systems that support speech and 3D gestures are able to deal directly with that uncertainty.

To address these issues, we present an architecture for 3D multimodal interaction in virtual reality (VR) and augmented reality (AR), and show how it can reduce errors by fusing symbolic and statistical information derived from speech, gesture, and the environment. We begin by describing related work in Section 2. Then, in Section 3, we introduce an application scenario that we are exploring. We describe the architecture itself in Section 4, the results of our study in Section 5, with discussion in Section 6. We draw our conclusions and present future work in Section 7.

2. RELATED WORK

Multimodal 3D interaction that includes speech dates back at least to Bolt's pioneering Put-That-There system [2], in which speech was integrated with 3D magnetic tracking of a user's arm in order to manipulate a 2D world. Motivated by Bolt's landmark work, numerous researchers have investigated multimodal 3D interaction for 2D worlds. Koons et al. [13] present a system that tracks 3D hand-based pointing gestures, speech, and gaze, and discuss its extension to other kinds of 3D gestures. The system copes with linguistic and referential ambiguity, but not erroneous

ICMI-PUI '03, November 5-7, 2003, Vancouver, BC

* Work was conducted at Pacific Northwest National Laboratory, operated by Battelle for the U.S. Department of Energy. Dr. McGee is now an employee of Natural Interaction Systems, 503-293-8414, dmcgee@naturalinteraction.com.

recognizer inputs. Lucente et al. [19] describe a system using a speech recognizer and a vision-based hand and body tracker that enables a user to manipulate large objects on a 2D display. Because of the size of the objects, it does not appear that reference resolution or uncertainty was of particular concern, nor was any error correction capability discussed. Similarly, no mention of coping with uncertainty was mentioned by Poddar et al. [24], who discuss a sophisticated system that understands speech and natural 3D gesture in a 2D environment in which the speech and gesture of cable television weather channel narrators were analyzed as the narrators described the movement of weather fronts across a map.

Many initial steps were taken that motivated building immersive multimodal systems [3, 28]. More recently, Duncan et al. (in [23]) present a multimodal 3D virtual aircraft maintenance assistant that includes an avatar driven by the user's tracked limbs, gesture recognition (seven CyberGlove-based gestures), spoken natural language input, and semantic fusion of temporally co-occurring input modes. However, the handling of uncertainty is not discussed. The usability of both LaViola's [17] multimodal 3D system, in which tools are created at the 3D location where a user's virtual hand is located, and Krum et al.'s [14] multimodal VR system, in which 2D finger gestures and speech are used to support 3D navigation, was reportedly undermined by speech recognition errors, because these systems also lacked error handling capabilities.

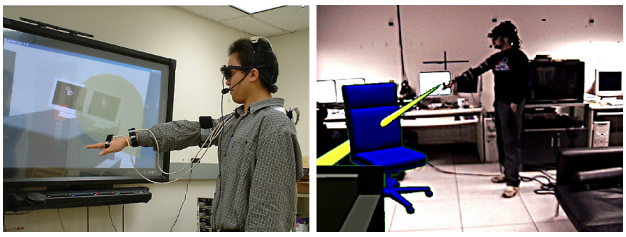
Three works are most comparable to ours. Latoschik [16] developed a 3D multimodal VR system based on augmented transition networks, which merges speech and gesture. However, neither the handling of recognition errors nor the possibility of mutual disambiguation (cf. 5.3 below) is mentioned. The 2D multimodal QuickSet architecture [4] was integrated into the Naval Research Laboratory's Dragon 3D VR system [5] to create a multimodal system that employs a 3D gesture device to create 2D "digital ink" projected onto the earth's surface in a 3D topographical scene. This system inherits the advantages of earlier 2D-only Quickset implementations; namely, it has been shown to offer *mutual disambiguation of modalities* [20, 22], resulting in error rate reductions of 19–40% [22]. Recently, Wilson and Shafer [29] have described a new six-degree-of-freedom (6DOF) tracked device (the X-Wand) that supports 3D gestures (pointing and rotating about any of the axes), which are coupled with simple speech recognition to manipulate objects in a living room. The system incorporates Bayesian networks for fusing interpretations, so in principle, it should be capable of mutual disambiguation. User testing was conducted for pointing accuracy, but not for the entire multimodal system.

Thus, few 3D multimodal projects consider the issues involved in the management of uncertainty across modalities. In this paper, we discuss how an architecture similar to that used in QuickSet for 2D gestures and digital ink in the Dragon environment can be extended to handle 3D gestures directly, and to take uncertainty into account in immersive 3D VR and AR environments.

3. APPLICATION SCENARIO

We illustrate the kinds of interactions that we address with an example of manipulating a virtual object in a simple interior design scenario. The user is standing in the room, with four 6DOF trackers attached to the right hand, right wrist, right upper arm, and head, as illustrated in Figure 1.

The system is configured to use either tethered magnetic sensors (Figure 1a) or hybrid sensors (Figure 1b). The user also wears a head-worn display (opaque for virtual reality, see-through for augmented reality). The user's view is shown in the plasma screen on the left of Figure 1a. He views a virtual surrounding environment in VR or a combination of real and virtual objects in AR.



(a) **(b)**

Figure 1. Interior design scenario: (a) turning a monitor in VR; (b) making a chair blue in AR.

4. SYSTEM ARCHITECTURE

The multimodal recognition architecture that we have developed consists of six components: an agent communication infrastructure, an interactive 3D environment (and its 3D proxy agent), a 3D reference resolution agent, a set of unimodal recognizer agents (one for each separate modality: speech and 3D gesture), a natural language parser, and a multimodal integrator agent (with embedded 3D gesture and gaze parsers). Their interactions are shown in Figure 3. The agent communication infrastructure [15], implemented in Prolog and Java, is the underlying distributed communication system that connects all other components, supporting both facilitated communication (through a blackboard) and direct peer-to-peer communication. The interactive 3D environment is responsible for capturing raw user interactions, handling virtual world state changes, visualizing the interaction as VR or AR, and performing geometric processing needed to determine candidate referents for manipulation; it communicates with the rest of the components through its 3D proxy agent. The 3D reference resolution agent maintains the relationship between sensors and body positions, and, based on the time stamps of raw 3D gestural and speech recognitions, requests (from the 3D proxy agent) the list of objects that were captured by the appropriate tracker's regions of interest. In Section 4.1.1 below, we discuss this in more detail.

4.1 Interactive 3D Environment

4.1.1 Regions of Interest

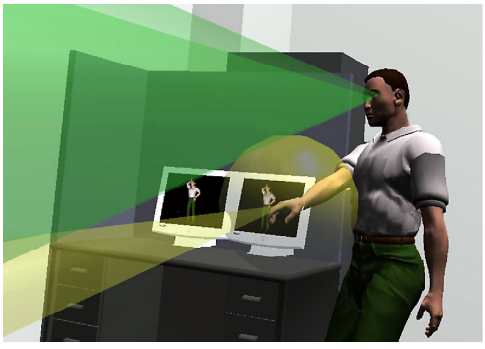


Figure 2. VR avatar controlled by tracked user, showing attached regions of interest.

An important task of the interactive 3D environment component is to find geometric correlates for the semantic meaning of deictic terms, such as "that," "here," and "there," as well as to facilitate selection of objects. We accomplish this through regions of interest that we call *SenseShapes*—volumes controlled by the

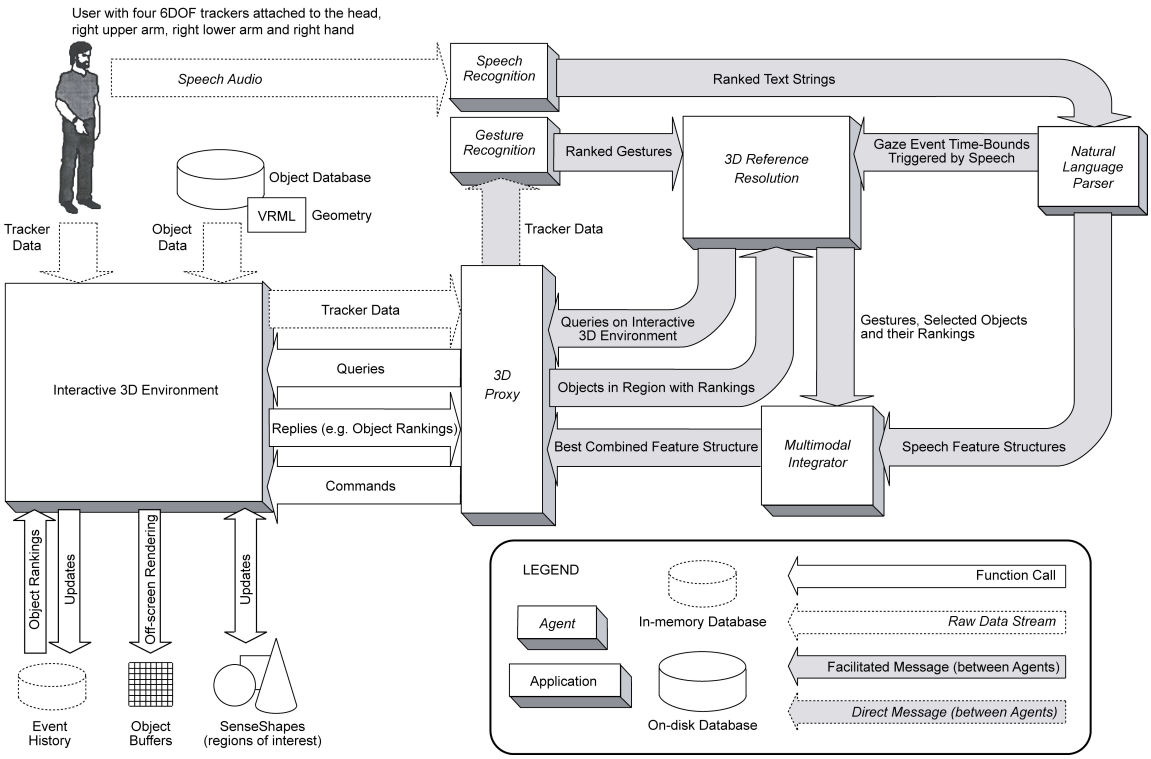


Figure 3. Multimodal interaction architecture.

user as she interacts with the environment, and which support statistical calculations about the objects they intersect. SenseShapes were developed in a test-bed that uses speech, head tracking, and glove-based finger tracking, without mutual disambiguation [21]. Our current implementation includes four primitives: cuboids, cylinders, cones, and spheres.

For example, Figure 2 shows two cones emanating from the user's eyes to approximate the field of view, a sphere around the user's hand to represent a volume that would encompass objects that are nearby and within reach, and another cone emanating from the user's hand, representing a "pointing volume." Multiple SenseShapes can be added to sensors, depending upon the application under development, adjusted in real-time to support various domain specific sensing needs, and made visible or invisible to the user as appropriate.

The SenseShapes are tested at each frame for intersections with objects in the environment, and this information is stored in the environment's event history. The event history uses an in-memory database, and supports complex queries through SQL. The 3D reference resolution agent queries the event history for objects that have been contained within a SenseShape at any point during a specified time period. The queries return the objects along with rankings that are used by the multimodal integrator to facilitate mutual disambiguation.

4.1.2 Object Rankings

We currently provide four different types of SenseShape–based rankings for an object, aggregated over a specified time period: time, stability, visibility, and center-proximity. The *time* ranking of an object is derived from the fraction of time the object spends in a region over a specified time period: the more time the object is in the region, the higher the ranking.

The *stability* ranking of an object expresses the stability of the object's presence in the region relative to other objects. We currently calculate this based on the number of times an object enters and exits the region during the time period. The most stable object possible never leaves the region, and the more entries and exits an object has, the lower its stability ranking.

The *visibility* and *center-proximity* rankings of an object reflect its visibility relative to selected SenseShapes. We compute the visibility of a conical region by rendering into an off-screen object buffer [1] a low-resolution version of the scene from a center of projection at the cone's apex, cropped to the cone's cross-section at a specific distance from the user. Each object is rendered with a unique color, allowing its pixels to be identified in the frame solely by their color. We currently generate two object buffers, one for an eye cone and one for a hand cone. We calculate three rankings for each object relative to a buffer. The visibility ranking orders objects by the fraction of the buffer that the object covers. The two center-proximity rankings order objects by how close the object's closest pixel is to the buffer's center, and by how close the object's pixels are on average from the buffer's center.

Included with the object rankings are the absolute time the object spent in the region and the number of times the object entered and exited the region. These values allow the multimodal integrator to classify the objects through a joint interpretation of the rankings and the application of appropriate thresholds.

4.2 Unimodal Recognition and Parsing

The source of all interaction in our system is the user's speech and the tracker data that represents her motion. These unimodal data streams are processed independently and in parallel, and then fused in the multimodal integrator agent.

4.2.1 Spoken Natural Language

Our speech agent uses an off-the-shelf recognition engine—the Microsoft Speech API 4-compliant *Dragon Naturally Speaking 6.* Our prototype system has a 151 word vocabulary and uses a context-free grammar recognition scheme. Results from the speech recognition engine are passed to the natural language parser as a

list of probability-ranked, time-stamped text strings. The parser interprets raw text strings such as, "Move that couch there," generating a potentially ambiguous set of meaning representations embodied in typed feature structures.

4.2.2 3D Gesture

Our 3D hand-arm gesture recognition agent receives and analyzes tracker data and sends messages to the 3D reference resolution agent whenever supported 3D gestures are encountered in the tracker data stream (see Section 4.4 for detail). We consider the tracker data stream for a particular sensor to be in a *stationary state* whenever the sensor's reports do not vary over time by more than an offset. The recognizer determines explicit start and end points by detecting stationary states without the need for specific user-defined positions or trigger mechanisms for locating start/end gesture points.

Recognition is based on a model of the body for which we track human movements and a set of rules for those movements. These rules were derived from an evaluation of characteristic patterns we identified after analyzing sensor profiles of the movements underlying the various gestures [7].

The system recognizes four 3D gestures, plus head gaze/direction (which we refer to as a *look*). The 3D gestures are:

- *Point*: a stationary wrist state, with un-bent arm (i.e., a relative angle between the directions of the wrist and upper arm sensors below a threshold of divergence), and convergence of looking and pointing directions (i.e., a relative angle between the directions of the wrist and head sensors below a threshold of divergence).
- *Push*: a point with hand up-down waving motion relative to the stationary wrist position.
- *Twist*: a point with hand palm-down/palm-up movement, with wrist and hand sensors maintaining a below-threshold relative angular difference.
- *Rotate*: a point with hand side-to-side motion relative to the stationary wrist position.

A rule-based analysis of pushing, twisting and rotating gestures can be given using the quaternion components provided by the sensor. Over the duration of any wrist stationary period, thresholding of the relative angular and positional differences—both over time and over the sensors relative to each other—moves the 3D gesture recognizer from state to state through the various 3D gestures listed above. Movement into a given 3D-gesture-type's state increments a counter associated with that state. At the end of the wrist stationary state (or after a specified time-out period) a weighted average of each 3D-gesture-type's count over the duration provides an *n*-best list of 3D gesture recognition probabilities.

4.2.3 Looking

The head sensor implicitly defines the gaze direction by estimating where a person is looking based solely on her head direction. This is a plausible simplification we have used to determine the focus of attention of the user without having to perform eye gaze tracking [26]. The details of *look* event generation are described below in Section 4.4.

4.3 Multimodal Integration

The job of the multimodal integrator is to find the highest scoring multimodal interpretation, given a set of *n*-best lists from each of the individual input recognizers. The basic principle is that of typed feature structure unification, which is derived from term unification in logic programming languages. Here, using multimodal grammar rules in a generalized chart parser [10, 11], unification of constituents rules out inconsistent information, while fusing redundant and complementary information through binding of logical variables that are values of "matching" attributes. The matching process also depends on a type hierarchy and a set of spatio-temporal or other constraints. The set of multimodal grammar rules specifies, for a given task, which of these speech and gesture interpretations unify to produce a command. For example, a rule that unifies a pointing gesture interpretation with a spoken language interpretation might specify that a 3D pointing gesture selecting an office object could be unified with speech referring to that same type of office object.

4.4 Integration Architecture

As shown in Figure 3, speech and gesture signals are recognized in parallel, and the unimodal recognizers then output lists of speech and 3D gesture hypotheses. After gesture recognition, the gestures are routed to the 3D reference resolution (3DRR) agent where, based on the gesture time stamp common to all list members, a copy of the *n*-best list of objects referenced at that time (based on hand cone tracking) is embedded within each gestural hypothesis. These hypotheses next are routed to the multimodal integrator.

Simultaneously, when speech is recognized, the *n*-best list of output strings, which all share a common start and end time stamp, flow into the natural language parser (NLP). The NLP maps each parsable speech string into a typed Feature Structure (FS). The *n*-best list of FSs then flows into the multimodal integrator.

Look events are triggered from within the NLP whenever there is an occurrence of a 3D speech FS of a lexical type whose semantics call for an object color change. The system assumes that during the time boundaries of this speech FS the user was gazing at the object(s) to be manipulated, and triggers a message to the 3DRR agent to create a "gestural" hypothesis of type *look_3D*, with an embedded *n*-best list of the objects referenced at that time (based on eye cone tracking). The duration of the look event corresponds to either the duration of the speech or the duration of a configurable minimum time window centered at the midpoint of the speech (in case the speech utterance is very short). That gestural hypothesis is, in turn, routed to the multimodal integrator (MI).

Thus, although the MI receives parsed FSs from the NLP, it receives only gestural hypotheses with embedded object lists from the 3DRR agent (eye cone objects with *look_3d* hypotheses; hand cone objects with other 3D types). Once received, these gestural hypotheses are immediately converted into typed FSs, which—as with speech FSs—are then processed by the MI's internal chart parser. In order for unification of a spoken FS and a 3D gestural FS to occur, certain constraints must be met. Generally these constraints are time-based, but in the case of 3D gestural FSs, the first constraint requires that the embedded object list be filtered by the object type (e.g., "table") provided by the speech FS. (This technique is common in computational linguistics). This process generates single object FSs of the type specified for the speech/gesture combination, which are subsequently enrolled in the chart. Currently, the probabilities of speech, gesture, and object identification are multiplied to arrive at a probability for their multimodal combination. Whenever the chart's agenda empties, the complete edges present on the chart become the multimodal *n*-best list, and the top member of this list is then executed by the system (e.g., the specified object's color, position, or orientation is changed).

5. USER TEST

In order to assess the strengths and weaknesses of this architecture, we conducted a small pilot test.

5.1 Subjects

Six unpaid subjects were recruited from colleagues and friends. Two were non-native speakers of English; one was female.

5.2 Methods

5.2.1 Equipment

Users wore four tethered Ascension Flock of Birds 6DOF magnetic tracker sensors in the following positions: (1) on top of the head, (2) on the upper arm near the shoulder, (3) on the wrist, and (4) on the back of the hand. They also wore a wireless microphone for speech recognition (in open-microphone mode), and wore an Olympus Eye-Trek FMD-150W head-worn display (in wide-screen mode), through which they viewed the virtual room in which they were interacting. They could turn and move about a target position on the floor one or two steps in each direction, enabling them to view the entire virtual room around them and to change their virtual position relative to objects in the room. The tests were run on a Pentium 4 Windows 2000 desktop computer (1.9 GHz, 1 GB RAM, 64MB DDR NVIDIA GeForce 3 graphics card). The 3D environment, gesture recognizer, speech recognizer, multi-agent infrastructure, and parsing agents were all run simultaneously on this machine. Auxiliary control and logging agents were run on two separate laptop computers connected by 100 Mbit Ethernet.

The head sensor's local position and orientation were used both to position the avatar representing the user in the virtual room, and to approximate gaze tracking. The SenseShape attached to the head sensor was an invisible cone, while that attached to the hand sensor was a visible cone, whose cross-sectional diameter was three-fifths that of the eye cone.

5.2.2 Task

Seven multimodal command templates were employed in the study. Two were used for color changes:

- *Look+speech*: Color changes accomplished only by looking at an object, with speech specifying what change to make. For example, "Make the table red <looking at part of a scene that contains a table>"
- *Point+speech*: Color changes accomplished by pointing at an object, with speech specifying what change to make. For example, "Make that <point to a chair> blue"

Two commands were used for moving objects:

- *Speech+2 points*: A single speech utterance (e.g., "Put that there") combined with two separate point gestures, the first designating an object to be moved and the second a position on the floor of the room to which it should be moved.
- *Speech+point* followed by *speech+point*: An object designation speech utterance (e.g., "Move that") and point gesture, followed by a pause in the speech recognition and then a position-designating spoken utterance (e.g., "over there") and a second point at a position on the floor.

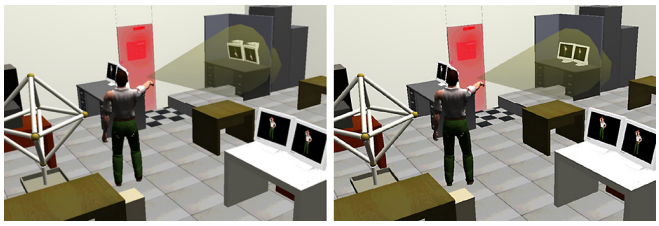


Figure 4. Result of, "Flip the monitor," with twist gesture.

The final three were rotational commands. In these, the 3D gesture designated both the object(s) under reference and the axis of rotation, while speech specified the direction and degree of the specified rotation, and optionally further specified the object(s) (e.g., "that table," versus the deictic, "that"):

- *Twist+speech*: Rotation of an object on the horizontal axis parallel to the twist gesture's pointing direction (see Figure 4).
- *Rotate+speech*: Rotation of an object on its vertical axis (e.g., "Turn the table <rotating gesture> clockwise").
- *Push+speech*: Rotation of an object on the horizontal axis perpendicular to the push gesture's pointing direction (e.g., "Stand the desk up <push gesture>").

Each user was trained to perform these actions using 16 example multimodal commands. Training typically took 15 minutes. After training, users engaged in a test session of 23 multimodal command tasks, typically taking 25 minutes. They could retry a failed command up to two additional times before going on to a subsequent one. During testing they were not coached, although direct informational questions (e.g., questions about the names of objects) were answered. With one exception, all objects acted on during testing had not been acted on during training. The tasks resembled those one would expect to perform in an interior decorating scenario: moving objects from one location to another, rotating objects to different orientations, arranging out of order objects, and changing object colors. The user was told to look at a certain part of the room, observe an object's changing state, and then replicate that change. The room contained 50 objects, categorized into 13 types. After the study, users filled out a questionnaire that assessed their opinion about the system's usability and learnability.

5.3 Measures

The system provided *n*-best recognition lists for speech, gesture, gaze, object reference and multimodal interpretation. From these lists, we computed *individual modality* and *multimodal recognition rates* as a function of (1) all command attempts, and (2) those attempts for which the system produced an output for that modality. Attempts that did not result in integrations could occur for a variety of reasons, including lack of speech recognition results (e.g., because the user was speaking too slowly, rapidly, loudly, softly, or ungrammatically), incorrectly performed pointing actions (e.g., a steady state was not reached), and incorrectly performed 3D gestures (e.g., the speed was too slow or the degree of pushing, twisting or rotating was too small).

Given that these were multimodal commands only, if no output was produced for an individual modality, then no output was generated for multimodal integration. Therefore, we present the system's performance results in a manner analogous to "recall/precision" in the information retrieval and natural language processing literature. We also measured the *overall system response time*, calculated from the time the user stopped speaking or gesturing to the time the system provided a graphical response. Data logging and auxiliary control and display agents did not significantly affect the response time.

Finally, the mutual disambiguation (MD) rate [22] was computed over all *N* scorable commands—those multimodal commands for which the correct multimodal integration occurred at the top of the multimodal *n*-best list.[1]

$$MD = \frac{1}{N}\sum_{i=1}^{N} Sign\left(\frac{\sum_{c=1}^{C} R_i^c}{C} - R_i^{MM}\right)$$

[1] Our definition of "scorable" is a modification of that in [22], in that we are only interested in those multimodal interpretations with MD that occur on the *top* of the multimodal *n*-best list, rather than anywhere within that list.

The MD rate computes the average over N commands of those for which the average rank (Ri) of the constituent recognitions (C) that contribute to the multimodal interpretation is higher than the rank of the correct multimodal integration on the n-best list (Ri^{MM}), minus those in which that average is less than Ri^{MM}.[2]

The parallel coordinate plot [9] in Figure 5 presents MD in terms of the n-best lists of hypotheses, including rankings (in square brackets) and probabilities, for each recognizer. A plot shows MD when the solid line that indicates the system's chosen interpretation is not straight. Occasionally, there are instances of *double pull-ups*, in which more than one modality was compensated for (i.e., "pulled-up") by the remaining modalities. For example, in Figure 5 the correctly ranked speech at the top of the speech n-best list "pulls up" *both* gesture and object hypotheses from lower down on their n-best lists.

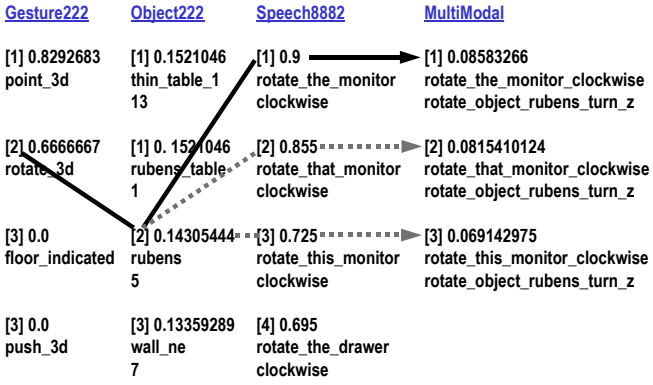


Figure 5. Parallel coordinate plot showing multiple hypotheses for each modality.

5.4 Results

After completing our user test we corrected system errors and re-scored the data (using a regression scripting mechanism similar to that described in [12]). These system corrections—which dealt with an error in the object list iterator, incorrectly normalized score combinations, and an inadequate end-point mechanism—allowed us to score 7 additional correct integrations. Results given in Table 1 and below are for the corrected system.

Modality	Percent of attempts producing an *n*-best list		Functional Accuracy	
User Test				
Speech	225/267	84.3%	194/225	86.2%
Gesture	222/228	97.4%	168/222	75.7%
Gaze	32/38	84.2%	32/32	100.0%
Gaze Objects	32/38	84.2%	23/32	71.9%
Gesture Objects	221/228	96.9%	115/221	52.0%
Multimodal	172/237	72.6%	140/172	81.4%
Overall Success			140/237	**59.1%**

Table 1. Recognition results.

Of the 237 multimodal command attempts across the 6 subjects 172 were in fact integrated. Subjects often made additional command attempts (up to three total attempts for each command) when previous ones failed. The 65 unintegrated commands are analyzed below in Section 5.4.1. Among the 172 integrated commands, 140 were correct, resulting in an overall 81.4% multimodal success rate. The 32 integrated, but incorrect commands are also analyzed below in Section 5.4.1. Of these 140 instances, 65 were correct in virtue of mutual disambiguation, representing an MD rate of 46.4% of the successful attempts, and 27.4% of all attempts. Of those 65 instances, 18 (27.7%) were successful in virtue of double pull-ups, yielding a double pull-up rate of 12.9% for successful commands, and 7.6% over all attempts. Over all attempts, the system succeeded 140 times (59.1%).

In terms of the individual modalities, among the 267 spoken command attempts, 225 (84.3%) produced results from the recognizer, of which 194 (86.2%) were correct at the sentence level, and 157 (69.7%) were verbatim correct. Likewise, of the 228 gestural attempts, 222 (97.4%) produced results, of which 168 (75.7%) were correct. However, of the 221 gesture object attempts that produced a result, 115 (52%) were correct. Finally, the subjects attempted to use speech plus gaze (but not gesture) 38 times, with the gaze identification agent producing a result 32 times. For those 32 object gaze instances, the top object on the gaze's n-best list of identified objects was correct 23 times (71.9%).

Given these baseline recognition rates, functionally correct spoken interpretations were pulled up by MD 13 times (20% of the MD cases), while gesture and objects were pulled up 22 times (33.8%) and 42 times (64.6%) respectively. By subtracting these instances in which MD corrected incorrect interpretations, one can compute that MD accounted for a relative error rate reduction of 41.9% for functionally correct speech, 40.7% for gesture, and 39.6% for object identification.

5.4.1 Error Analyses

A. Failure to Integrate Errors			65
1. *With Some Recall for all Recognizers*		*30*	
Poor speech or gesture recognition	14/30	46.7%	21.5%
Poor object identification	4/30	13.3%	6.2%
Procedural errors	10/30	33.3%	15.4%
Unexplained System Errors	2/30	6.7%	3.1%
2. *With Some Failure of Recognition*		*35*	
Non-native, accented speech	22/35	62.9%	33.7%
Out of grammar speech	2/35	5.7%	3.1%
Speech disfluency	2/35	5.7%	3.1%
Unexplained Speech Recog. Errors	2/35	5.6%	3.1%
Poor gesture recognition	7/35	20.0%	10.8%
B. Incorrect Integration Errors		32	
Incorrect Object Selection	14/34	43.8%	
Procedural errors	9/34	28.1%	
System Errors	8/34	25.0%	
Poor gesture recognition	1/34	3.1%	

Table 2. Error Analyses.

Of the total system failures, 65 were unintegrated attempts (Table 2.A). In 30 of these cases (Table 2.A.1) there was some recognition for all input streams. Failure to find an integrated combination was caused by: poor speech or gesture recognition (14 cases), poor object identification (4), procedural errors (10), and unexplained system errors (2). For the other 35 unintegrated attempts one or more recognizers had no output (Table 2.A.2). The contrib-

[2] Since for this paper, we are only interested in the top-scoring multimodal interpretation, the last case ("negative MD") cannot arise. However, for the sake of consistency with the literature, the formula is left unchanged.

uting factors were speech (28/35) and gesture errors (7/35). For speech, the factors leading to recognition failure were non-native accented speech (22), out-of-grammar speech (2), speech disfluency (2), and unexplained failure (2). For gesture failures, the factors were: too fast, too slow, or ambiguous user motion (6/35), and a non-intersecting floor location gesture (1/35).

Among the 34 integrated commands that were not correct (Table 2.B), the contributing factors were 14 incorrect object selections (e.g., pointing through one object towards another, occluding the target in the off-screen buffer), 9 procedural errors (e.g., out-of-grammar speech, or wrong command given), 8 system errors, and 1 ambiguous gesture.

The 14 *incorrect object selection* errors in Table 2.B were instances in which the user was inadvertently gesturing in front of an object that filled up the hand cone's off-screen buffer. She was, however, still looking over that point-occluding object at the second object about which she was speaking. Here, the use of speech and gaze was found to override the semantically incoherent information provided by pointing. We can systematically address this error in the future by using a weighted combination of *hand* and *eye* object lists for all such arm-gestural object selection events.

5.4.2 Response Times

Based on a sample of 20% of the subjects' successful attempts (randomly selected), the mean system response time was 1.1 seconds (standard deviation = 0.68), and was not significantly affected by logging. The average length of user 3D hand/arm gestures was 3.62 seconds (standard deviation = 1.76), with the maximum length being 7.28 and minimum length being 0.58 seconds.

5.4.3 Subjective Evaluation

From the questionnaires, it was found that the system scored an average of 3.5 on the usability scale of 1–5 (not usable–extremely usable), and an average of 3.3 on learnability (not learnable–extremely learnable). Users were concerned that some of the gestures were difficult to perform, and that the gesture(s) needed to perform the changes depicted were not obvious.

6. DISCUSSION

The results are presented in terms of functionally correct results, rather than based on verbatim speech recognition, because the recognizer is built to map utterances that are slightly disfluent or out of grammar into the closest grammatically correct utterance, a phenomenon that occurred in 14.7% of the successful commands. A second reason to concentrate on functional accuracy is that this is what the user experiences.

Whereas Oviatt [22] reports that 12.5% of multimodal 2D pen/voice interactions were successful because of MD, 46.4% of the interactions in this 3D user test succeeded because of MD. Oviatt also reported very few cases of double pull-ups, while such cases were relatively common here (12.9% of successful commands). Given that object identification was correct only 52% of the time, these findings confirm that the system is functioning as desired. Moreover, we hypothesize that there will always be substantial ambiguity about 3D object identification given only gestures, requiring an architecture that can employ other sources of information to filter the list of possible objects.

7. CONCLUSIONS AND FUTURE WORK

We have described an architecture in which mutual disambiguation can support multimodal interaction in immersive 3D AR and VR environments. The system is designed to uncover the best joint interpretation of speech, gesture, and object identification given semantic and statistical properties. To validate our hypothesis, we designed and implemented a test bed based on this architecture and conducted a small user test. Initial results demonstrate that over 45% of the system's successful performance was due to its mutual disambiguation capabilities.

These results demonstrate how mutual disambiguation of multimodal inputs can function to produce a more robust system than would be possible based on the success of the individual modalities. The architecture described here improves upon the current state-of-the-art in 3D multimodal research by reducing uncertainty and ambiguity through the fusion of information from a variety of sources. Our basic architecture is an extension to what has been previously reported for 2D multimodal interaction, taking full advantage of additional 3D sources of information (e.g., object identification, head tracking, and visibility).

This initial prototype system interpreted its users correctly only 59% of the time. Aside from many immediately fixable errors, the system is still somewhat limited; in particular, its current probability combination scheme (multiplication) performs at the low end of its possible range [30]. We expect that other probability combination schemes (average, sum, linear combination with trained coefficients) will be more valuable as the system scales up.

We are currently planning a number of improvements to this initial implementation:

- *More natural gesture recognition.* Based on data collected during a multimodal "Wizard of Oz" VR experiment [7], natural gestures are being identified, classified by hand, and then provided as a corpus for training hidden Markov model-based gesture recognizers.
- *Lowering the procedural error rate.* Various techniques will be developed to minimize the number of attempts that do not produce results (e.g., the use of audio tones to indicate that a stationary state has been attained).
- *Better use of gaze direction.* Gaze direction is currently underutilized and in need of more empirical support. The system's object identification rate could likely be improved by learning how to weight the objects in either the hand or eye cones.
- *Determining whether the hand cone must be visible.* We will conduct a formal user study to determine whether MD may compensate for uncertainty about where the user is pointing, especially in augmented reality.
- *Learning the utility of recognition features.* Object identification provides a number of features whose importance is currently unknown. We intend to collect a corpus of user interactions, compute the set of recognition features, and model statistical weights over the speech, gesture, and object features.
- *Use of vision-based tracking technologies.* Existing tracking technologies are cumbersome, and difficult to integrate. Future versions of this architecture should be particularly well-suited to using lower-precision vision-based tracking [27].
- *Support for finger gestures.* We have already, in another test-bed, begun to experiment with an instrumented glove to track the fingers [6], in addition to the arm, wrist, hand, and head.
- *A more comprehensive vocabulary and grammar.* While vocabulary and grammar extensions can increase expressive power, they can also result in more speech recognition errors and linguistic ambiguities. Future work will explore wider coverage and other domains.

8. ACKNOWLEDGEMENTS

Work conducted at Columbia University was supported by ONR Contracts N00014-99-1-0394, N00014-99-1-0249, and N00014-99-

1-0683, NSF Grants IIS-00-82961 and IIS-01-21239, and a gift from Microsoft. Dr. McGee's work on this project was conducted at Pacific Northwest National Laboratory (PNNL). PNNL is operated by Battelle for the U.S. Dept. Of Energy, contract DEAC06-76RL0 1830. Work conducted at OHSU was supported by ONR Contracts N00014-99-1-0377, N00014-99-1-0380, and N00014-02-1-0038.

9. REFERENCES

1. Atherton, P. R. A method of interactive visualization of CAD surface models on a color video display. *Proc. ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '81)*, ACM Press, 1981, 279-287.
2. Bolt, R. A. Put-That-There: Voice and gesture at the graphics interface, Computer Graphics, 14(3), 1980, 262-270.
3. Bolt, R. A. and Herranz, E. Two-handed gesture in multi-modal dialog. *Proc. ACM Symposium on User Interface Software and Technology (UIST '92)*, ACM Press, 1992, 7-14.
4. Cohen, P. R., Johnston, M., McGee, D. R., Oviatt, S. L., Pittman, J. A., Smith, I., Chen, L. and Clow, J. QuickSet: multimodal interaction for distributed applications. *Proc. International Multimedia Conference (Multimedia '97)*, ACM Press, 1997, 31-40.
5. Cohen, P. R., McGee, D. R., Oviatt, S. L., Wu, L., Clow, J., King, R., Julier, S. and Rosenblum, L. Multimodal interactions for 2D and 3D environments, IEEE Computer Graphics and Applications, (July/Aug), 1999, 10-13.
6. Corradini, A. and Cohen, P. R. Multimodal Speech-Gesture Interface for Handfree Painting on a Virtual Paper using Partial Recurrent Neural Networks as Gesture Recognizer. *Proc. Int. Joint Conf. on Artificial Neural Networks (IJCNN '02)*, 2293-2298.
7. Corradini, A. and Cohen, P. R. On the Relationships among Speech, Gestures, and Object Manipulation in Virtual Environments: Initial Evidence. *Proc. Int. CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems, (2002)*, 52-61.
8. Fröhlich, B., Plate, J., Wind, J., Wesche, G. and Göbel, M. Cubic-moused-based interaction in virtual environments, IEEE Computer Graphics and Applications, 20(4), 2000, 12-15.
9. Inselberg, A. and Dimsdale, B. Parallel coordinates: A tool for visualizing multi-dimensional geometry, Proceedings of IEEE Visualization, 901990, 361-378.
10. Johnston, M. Unification-based multimodal parsing. *Proc. Int. Joint Conf. of the Assoc. for Computational Linguistics and the Int. Committee on Computational Linguistics*, Association for Computational Linguistics Press, 1998, 624-630.
11. Johnston, M., Cohen, P. R., McGee, D. R., Oviatt, S. L., Pittman, J. A. and Smith, I. Unification-based multimodal integration. *Proc. Meeting of the Assoc. for Computational Linguistics*, ACL Press, 1997, 281-288.
12. Kaiser, E. C. and Cohen, P. R. Implementation testing of a hybrid symbolic/statistical multimodal architecture. *Proc. Int. Conf. on Spoken Language Processing (ICSLP '02)*, 173-176.
13. Koons, D. B., Sparrell, C. J. and Thorisson, K. R. Integrating simultaneous input from speech, gaze, and hand gestures, in *Intelligent Multimedia Interfaces*, M. T. Maybury. AAAI Press/MIT Press: Cambridge, MA, 1993, 257-276.
14. Krum, D. M., Omoteso, O., Ribarsky, W., Starner, T. and Hodges, L. F. Speech and gesture control of a whole earth 3D visualization environment. *Proc. Joint Eurographics-IEEE TCVG Symposium on Visualization (VisSym 02)*, IEEE Press, 2002, 195-200.
15. Kumar, S., Cohen, P. R. and Levesque, H. J. The Adaptive Agent Architecture: Achieving Fault-Tolerance Using Persistent Broker Teams. *Proc. Int. Conf. on Multi-Agent Systems*, 2000.
16. Latoschik, M. E. Designing transition networks for multimodal VR-interactions using a markup language. *Proc. IEEE fourth International Conference on Multimodal Interfaces (ICMI '02)*, IEEE Press, 2002.
17. Laviola, J. MSVT: A virtual reality-based multimodal scientific visualization tool. *Proc. IASTED Int. Conf. on Computer Graphics and Imaging*, 2000, 1-7.
18. Liang, J. and Green, M. JDCAD: A highly interactive 3D modeling system, Computers and Graphics, 18(4), 1994, 499-506.
19. Lucente, M., Zwart, G.-J. and George, A. Visualization Space: A testbed for deviceless multimodal user interfaces. *Proc. AAAI Spring Symp.*, AAAI Press, 1998, 87-92.
20. McGee, D. R., Cohen, P. R. and Oviatt, S. L. Confirmation in multimodal systems. *Proc. Int. Joint Conf. of the Assoc. for Computational Linguistics and the Int. Committee on Computational Linguistics*, Université de Montréal, 1998, 823-829.
21. Olwal, A., Benko, H. and Feiner, S. SenseShapes: Using Statistical Geometry for Object Selection in a Multimodal Augmented Reality System, in *The Second International Symposium on Mixed and Augmented Reality*.
22. Oviatt, S. L. Mutual disambiguation of recognition errors in a multimodal architecture. *Proc. ACM Conf. on Human Factors in Computing Systems*, ACM Press, 1999, 576-583.
23. Oviatt, S. L., Cohen, P. R., Wu, L., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J. and Ferro, D. Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions for 2000 and beyond, in *Human-Computer Interaction in the New Millennium*, J. Carroll. Addison-Wesley: Boston, 2002,
24. Poddar, I., Sethi, Y., Ozyildiz, E. and Sharma, R. Toward natural gesture/speech HCI: A case study of weather narration. *Proc. ACM Workshop on Perceptual User Interfaces (PUI 98)*, ACM Press, 1998.
25. Poupyrev, I., Billinghurst, M., Weghorst, S. and Ichikawa, T. Go-Go interaction technique: Non-linear mapping for direct manipulation in VR. *Proc. ACM Symposium on User Interface Software and Technology (UIST '96)*, ACM Press, 1996, 79-80.
26. Stiefelhagen, R. Tracking Focus of Attention in Meetings. *Proc. 4th International Conference on Multimodal Interfaces (ICMI 02)*, IEEE Press, 2002, 273-380.
27. Tollmar, K., Demirdjian, D. and Darrell, T. Gesture + Play: Full-body interaction for virtual environments. *Proc. ACM Conference on Human Factors in Computing Systems (CHI 2003)*, ACM Press, 2003, 620-621.
28. Weimer, D. and Ganapathy, S. K. A synthetic visual environment with hand gesturing and voice input. *Proc. ACM Conference on Human Factors in Computing Systems (CHI '89)*, ACM Press, 1989, 235-240.
29. Wilson, A. and Shafer, S. XWand: UI for intelligent spaces. *Proc. ACM Conference on Human Factors in Computing Systems (CHI '03)*, ACM Press, 2003.
30. Wu, L., Oviatt, S. L. and Cohen, P. R. Multimodal integration-A statistical view, IEEE Transactions on Multimedia, 1(4), 1999, 334-341.