Distributed Pointing for Multimodal Collaboration over Sketched Diagrams

Paulo Barthelmess, Ed Kaiser and
Xiao Huang
Oregon Health & Science University
OGI School of Science & Engineering
{paulo,kaiser,huangx}@cse.ogi.edu

David Demirdjian
Massachusetts Institute of Technology
Artificial Intelligence Laboratory
demirdji@ai.mit.edu

ABSTRACT

A problem faced by groups that are not co-located but need to collaborate on a common task is the reduced access to the rich multimodal communicative context that they would have access to if they were collaborating face-to-face. Collaboration support tools aim to reduce the adverse effects of this restricted access to the fluid intermixing of speech, gesturing, writing and sketching by providing mechanisms to enhance the awareness of distributed participants of each others' actions.

In this work we explore novel ways to leverage the capabilities of multimodal context-aware systems to bridge co-located and distributed collaboration contexts. We describe a system that allows participants at remote sites to collaborate in building a project schedule via sketching on multiple distributed whiteboards, and show how participants can be made aware of naturally occurring pointing gestures that reference diagram constituents as they are performed by remote participants.

The system explores the multimodal fusion of pen, speech and 3D gestures, coupled to the dynamic construction of a semantic representation of the interaction, anchored on the sketched diagram, to provide feedback that overcomes some of the intrinsic ambiguities of pointing gestures.

Categories and Subject Descriptors: H.5.3 [Group and Organization Interfaces]: Collaborative Computing; Synchronous interaction H.5.2 [User Interfaces] Natural language; Input devices and strategies I.2.10 [Vision and Scene Understanding] 3D/stereo scene analysis

General Terms: Collaboration; Multimodal; Context-aware

Keywords: Collaborative Interaction; Multimodal processing; Intelligent interfaces; Gesture.

1. INTRODUCTION

Collaboration has traditionally been associated with participants' ability to orient their actions according to a rich multimodal communicative context created by a group's intertwined and fluid use of speech, gestures, writing and sketching. Communication technologies have allowed remote participants to interact, but at the price of limited access to a remote communicative context.

Strategies normally used to cope with this limited access lead in general to more circuitous language to compensate for the lack of a more tightly nit collaboration context, transforming the communication to accommodate the lack of direct access [6, 3], e.g. by including explicit narrations of what would otherwise be directly visible within the common context. Group dialogues are thus made to fit the available technology rather than follow a natural flow of interaction that would be otherwise possible.

Collaboration technology has strived to attenuate some of the adverse effects of distribution by providing visibility to certain aspects of remote contexts.

While providing valuable support for collaboration, these tools do not take advantage of, or sometimes even accommodate natural work practices and communicative behavior of collaborative groups. In most cases, participants are required to introduce into their routine operations that serve the sole purpose of steering the technology, adding to users' cognitive load and disrupting the flow of work they are used to without clear contributions to work performance.

We are interested in exploring technology affordances to support meetings *as they are normally run*, and still be able to provide a rich shared experience that goes beyond what is currently supported by commercial collaboration tools. We explore a collaboration scenario in which small groups of co-located participants interact with groups located at remote sites, and wish to support both the co-located interaction as well as the remote. To this end, we examine the affordances of an emerging class of applications capable of capturing and interpreting a wide range of group multimodal communicative acts by incorporating recognition of speech, sketches, gestures, and capabilities for modeling dialogues, detecting topic and phase shifts, and identifying user-action patterns.

In the present work, we take steps in this direction by exploring novel ways to leverage the capabilities of the Charter Suite [9] - a multimodal intelligent system that supports co-located sketching of project schedules. We extend the Charter Suite to include support for distributed collaborative design of project schedules, that can now be simultaneously sketched on multiple distributed interactive boards. The distributed Charter presented here deals with issues introduced by the simultaneous updates originated by potentially

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ICMI'05, October 4–6, 2005, Trento, Italy.
Copyright 2005 ACM 1-59593-028-0/05/0010 ...$5.00.

dissimilar devices (e.g. an instrumented board and a tablet PC), including the diffusion of strokes and maintenance of the uniformity of the user interfaces across sites. The distributed version promotes as well the cross-site awareness of naturally occurring pointing gestures referencing elements of the shared sketched diagram made for instance by a participant that is sitting at a meeting table.

We show how the system tracks, interprets and makes gestures visible across sites, so that remote participants may be able to integrate them seamlessly into the ongoing multimodal discourse. We highlight how the multimodal fusion of pen strokes, gesture and speech is used to handle the ambiguity intrinsic to large amplitude, short duration untethered gestures made from a distance, as is the case with naturally occurring gestures made by participants towards a diagram sketched on a board while they are sitting at a meeting table or standing at a distance from the board.

Target resolution is based on the fine-grained semantic model of the diagram that is dynamically built from interpretations of pen strokes captured from an instrumented board. This model provides evidence as to what the likely targets of an imprecise gesture might be, based on the distance of these elements from a perceived pointing gesture's point of intersection with the board's surface. Further disambiguation is provided via fusion of spoken references to named elements of the diagram.

We begin the discussion by contrasting the approach with existing work reported in the literature (Section 2). We then describe how the application is used and introduce its distributed collaboration support capabilities in Section 3. The system is described next, starting with an overview of the architecture (Section 4.1), followed by the the discussion of the processing of each modality (Sections 4.2-4.4). Section 5 shows the results of the pointing accuracy evaluation and discusses the implications in face of collected data. The paper ends with conclusions and description of future work (Section 6).

2. RELATED LITERATURE

Shared sketches using virtual distributed white boards have been in use for some time, and have been incorporated into popular commercial products such as Microsoft NetMeeting. NetMeeting and similar tools do not apply any form of semantic interpretation to the content of the boards, which is simply treated as pixels shared across screens. Telepointers have been explored as a means to provide a sense of embodiment to remote users of such shared artifacts. These pointers allow participants to make reference to regions of shared displays and are therefore able to support some level of access to a remote context via gesturing. This gesturing is in most cases conveyed through mouse movements, thus requiring explicit device manipulations that might be fitting in scenarios in which each participant is at their workstation, but that are problematic while supporting groups of co-located participants interacting from remote sites, as we target here.

Video has been used as a means to bridge distributed contexts, but has proved problematic in supporting a stronger sense of co-presence. Actions conveyed via video are framed within local contexts and become hard or impossible to reconstruct from a foreign context. A pointing gesture made towards a video display, for instance, is in general not retrievable at remote sites, as participants are unable to tell what object within their own space was being pointed to [12]. Video manipulation has been used to attenuate this problem and provide a more unified sense of context among remote participants. Clearboard [7] explored video projections on shared surfaces that would give users the impression that they were working in contiguous rooms separated by a glass window onto which they could sketch; systems such as MAJIC [16], HyperMirror [15], and Reflection [1], compose images from multiple participants onto a single composite video space. Others (e.g. VideoWhiteboard [18]) display participants' images as shadows or silhouettes at remote sites. While providing interesting solutions to the problem of bridging contexts, these systems are intrinsically limited to what is achievable by manipulations of video image pixels.

Finally, immersive and virtual reality environments attempt to provide users an experience that mimics face-to-face meetings. The interpretation of user actions is in most cases restricted to a limited vocabulary that give users capabilities to navigate the environment using specific gestures.

Unlike the approaches described in this section, that rely on the manipulation of the direct appearance e.g. of video images, or require users to perform specific actions (e.g. moving pointing devices) that are associated with the constrained semantics the system is able to interpret, here we explore the affordances of an intelligent, contextually aware and perceptually rich system that is able to dynamically build semantic models of the ongoing natural interaction. The existence of the system becomes visible when it plays a role in mediating distributed work contexts.

3. THE APPLICATION

This section describes the context within which the system is used, the artifacts whose construction it supports, and overviews the distributed collaboration functionality that is offered, emphasizing the support for gesturing awareness.

3.1 Application context

The distributed Charter targets collaborative sessions in which multiple participants, usually stakeholders, get together to define a schedule of a project. Schedule design is usually a high-stake, highly collaborative activity. The outcome of these planning sessions is expected to shape future work in important ways; it in many cases represents commitments of a group of people with respect to their own individual actions and expectations with respect to the actions of others. It offers thus a good domain for experimenting with intelligent multimodal collaboration support, particularly because of the richness of the multimodal dialogue that is in general elicited.

Charter supports a common collaboration scenario in which co-located groups of participants interact among themselves, and might interact with additional participants located at one or more remote sites. At each site, participants assemble for a design session in a regular meeting room that is unobtrusively instrumented. For the work presented here, the room is fitted with an interactive board or a touch sensitive plasma display; microphones for capturing individual speech and room audio; and a stereo camera mounted above the board that captures images used for gesture tracking and recognition. The distributed support reported here makes it possible for participants at multiple remote sites, including mobile users (using e.g. a tablet PC) to collaborate.

The collaboration revolves around a few different artifacts at different times, as participants' understanding of the details of what a project might entail evolves. Eventually a work break-down structure emerges, and participants solidify their commitments to the expected level of effort, temporal allocation and milestones. The Charter suite currently provides services to this latter phase, based on free-hand sketching of project schedules, a sample of which is displayed in Figure 1.

Charter adopts Gantt Charts as the formalism used to represent project schedules. A Gantt chart is a diagrammatic, two-dimensional representation of a planned series of tasks and important milestones, and incorporates in a compact and graphical form constraints related to time, tasks, budget and other dependencies (Figure 1). Lines represent tasks; milestones are represented as diamonds. Temporal information is associated with the placement of milestones and task lines within a temporal grid defined as an xy-axis. Task durations are represented by the length of the lines according to this same temporal framework.

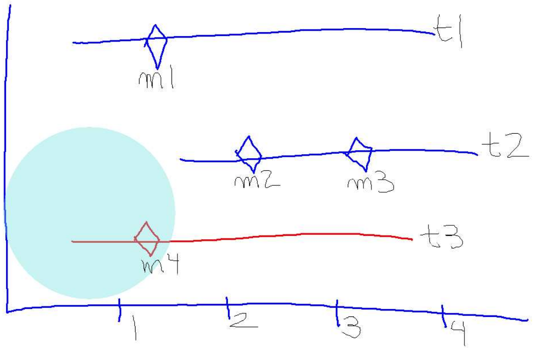


Figure 1: Graphical representation of project schedule used by Charter. The blue circle indicates that a pointing gesture over that regions has been detected. The elements of the diagram are displayed color-coded from blue to red to indicate the increasing likelihood that they are the intended target of the pointing.

3.2 Distributed collaboration support

A first level of support for distributed collaboration is provided by the diffusions and rendering of pen strokes performed over interactive boards at multiple remote sites. At all times, the system keeps consistent the displays that are shared among sites, that are made to show the same final appearance, integrating contributions to the sketched diagram originated at multiple sites.

Besides supporting awareness of remote pen strokes, the distributed Charter supports the diffusion of pointing gestures. Naturally occurring pointing gestures towards the sketched diagram on the board, made by a participant while sitting at a meeting table or standing at a distance are tracked and analyzed by a 3D stereo-vision system.

The challenges associated with interpreting such gestures are related to their intrinsic ambiguity. This ambiguity can be understood under the framework provided by Fitt's Law [13]. Fitt's Law gives the time T to acquire a target, e.g. move from a starting position to the center of the target as a function of the amplitude of the movement (from start to the target's center), and the size of the target, as $T = a + b \log_2 (D/W + 1)$. D is the movement amplitude from start to target's center, and W is the width of the target. The constant a is associated to the time required to perform whatever action is required once the target has been acquired; in our case, a is associated with the period of time the pointing arm needs to remain stationary in order for the system to detect that pointing is being performed. The constant b reflects intrinsic difficulties of use associated with the type of device that is used to achieve the movement. In the present case, this is associated with the difficulty of unsupported full arm pointing. $ID = \log_2 (D/W + 1)$ is called the *index of difficulty.* It isolates the influence that gesture amplitude and target width variables have on the acquisition time. ID reflects the intuition that it takes longer to point to a small target from a large distance than to point to a large target from a close distance. Naturally occurring pointing gestures made from someone sitting across a table towards a single individual small element on a board sketch, say a milestone represented as a diamond shaped figure a few centimeters wide, would require a long time to be made, if it were to be unambiguously precise. Human beings overcome this difficulty by moving closer to the their targets, or by adding complementary information [17] via speech, disambiguating quick gestures towards an approximate region of focus by naming or describing the specific objects within this region [10].

Gesture processing in Charter takes these factors into account, and bases its analysis of pointing gestures on a region of focus around the centroid of the gesture that grows or shrinks as the user moves away or closer to the board respectively. The focus is used internally to restrict the search and assign likelihoods to individual target elements of the schedule sketch according to their proximity to the gesture centroid, within a *region of focus.*

Following MAVEN [10], the region of focus is expressed in terms of the intersection of a cone originating at the tip of the pointing arm and the surface of the interactive board. The angle of the cone is set based on the empirical evaluation (see Section 5) to indicate the pointing precision from a sitting position at a meeting table located at a distance from the interactive board.

Charter is able to distinguish individual elements of a sketch, including their names given by labels, due to the online parsing and analysis that associates pen strokes to items of a semantic representation. Based on this semantic representation, Charter is able to determine the individual diagram constituents that are most likely to be the targets given a region of focus. This is made visible though coloring of the sketch constituents according to the likelihood that individual elements are the targets of the gesture (Figure 1).

Participants' speech accompanying a gesture is used to provide additional cues to disambiguate the gesture, based on references to element names, as given e.g. by their attached handwritten labels. Successful disambiguation via speech fusion results in a narrowing of the gesture display to include only those items within the original area of focus whose names match the spoken utterances in the vicinity of a gesture.

In the next sections we describe in further detail how the signal processing analysis and fusion takes place within the system.

4. SYSTEM DESCRIPTION

We now turn our attention to the system aspects of the solution. We begin with an overview of the distributed system's architecture (Section 4.1), and then describe in further detail the processing of each of: pen (Section 4.2), gesture (Section 4.3), and speech modalities (Section 4.4). We emphasize throughout the issues related to collaboration support.

4.1 Architecture

The system is organized around a multiagent architecture, and is a direct descendant of the QuickSet [4] multimodal system and MAVEN [10], from which gesture recognition and multimodal integration components were taken. The system is an augmentation of the Charter Suite [9], to which collaboration support functionality is added as described in the present paper.

Figure 2 presents a (simplified) view of the system's architecture. The input generated by the pen strokes on an instrumented board, and the speech and video captured from microphones and cameras are processed by components serving each of the potentially multiple sites involved in a collaboration.

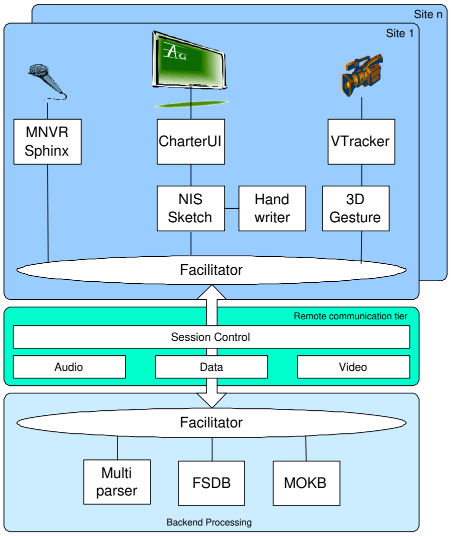


Figure 2: System architecture.

Communication among components is organized as a two-tiered structure. Agents communicate using the OAA multiagent architecture [14] within each a site, routing messages via a facilitator. A second communication tier handles remote communication among sites. This second tier employs a data transport that connects remote OAA facilitators, and audio and video multipoint communication transport among sites. For testing purposes within a single local area network, we forgo the secondary communication tier and employ a single OAA facilitator.

The message streams generated at each site are fused and integrated by a set of back-end processing agents. These agents are responsible for building a unified semantic model that reflects the most likely interpretation of the inputs originated at one or more sites at any moment.

We next examine how each of the modalities (pen strokes, gestures and speech) are processed, highlighting the role played to support multi-site remote collaboration.

4.2 Pen stroke processing

Three agents are responsible for handling processing of electronic ink captured from an instrumented interactive board: CharterUI, NISSketch and Handwriter.

CharterUI handles the sketching user interface front-end functions. CharterUI captures pen movements over an instrumented board and performs an initial segmentation of the electronic ink traced while an instrumented marker is pressed against the board. These individual strokes are combined into glyphs forming e.g. a handwritten word, or a milestone, or a task line. This first segmentation is based on temporal and spatial thresholds and has to be further refined by other mechanisms that we do not elaborate in this paper, that make use of context to reinterpret glyphs by breaking them apart or joining them together.

NISSketch[1] is a commercial electronic ink recognizer that handles the classification of glyphs into tokens corresponding to what will be interpreted within a Gantt chart vocabulary of symbols, including e.g. xy-axis, tick marks, task lines, milestones, and labels. NISSketch outputs a list of hypotheses ranked by likelihood. The decision of how to interpret each of these symbols is made by Multiparser - the multimodal fusion engine - as we describe later.

Handwriter is the agent that provides handwritten recognition services. It is based on commercial recognition engines (currently MS-Handwriter and Calligrapher). Its output is also a ranked list of hypotheses.

A basic level of awareness diffusion for collaboration is directly supported by the CharterUI agent, that displays ink originated at remote sites that is propagated through the communication infrastructure via messages. Each instance of the CharterUI agent running at each site participating in a collaboration displays the foreign ink alongside the local ink, allowing therefore for a first shared, integrated view of the inking actions performed by multiple remote participants. CharterUI is also responsible for displaying the remote pointing gestures, as described in the following section.

4.3 Pointing gesture recognition

Two components handle processing of pointing gestures towards the board diagram: VTracker and 3DGesture recognizer.

VTracker estimates the body pose of a user using an untethered vision-based system, using images from a stereo camera placed on top of the board, mounted at an angle of about 45 degrees. A tridimensional reconstruction of the scene is performed in real-time via the use of a disparity map estimation algorithm.

In order to model human bodies, we use a 3D cylindrical model of articulated appearance : limbs (head, torso, arms, forearms) are modeled as rigid bodies connected by spherical joints. We have designed an algorithm for estimating articulated motion based on rigid motion estimates of the articulated models constituent parts.

In brief, the tracking algorithm maps a 3D cylindrical model to the tridimensional reconstruction of the scene provided by the stereo camera. More exactly, the well-known ICP algorithm [2] is used to coarsely align two clouds of 3D

[1] http://www.naturalinteraction.com

points and estimate an initial rigid motion between body parts. The ICP algorithm iteratively estimates the rigid transformation between two clouds of 3D points. First, a set of elementary displacements is estimated by finding, for each point in the first cloud (model) the closest point in the second cloud (scene). Then the global rigid motion is computed by integrating all the elementary displacements and applied to the first cloud. The process is repeated until convergence is achieved (that usually happens in few iterations when the two clouds are initially close to each other).

An additional joint constraint reinforcement step corrects the previously estimated body part motions so that they exactly satisfy the joint constraints of the body model. This algorithm is fully described in [5]. The advantage of the approach is that the computation of the articulated motion is performed on reduced size equation systems even though the articulated model has many degrees of freedom.

3DGestureRecognizer is the component that receives and analyzes tracked data streams generated by VTracker. Full arm pointing gestures are currently recognized. The location pointed to by a user is estimated as the intersection of the line formed by the (right) shoulder and hand, and the board. 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. The detailed gesture recognition mechanism was developed in the context of the Maven system [10].

To face the intrinsic ambiguities of gestures of large amplitude, the Charter Suite makes use of the semantic model of the sketched diagram that is built online by the system, in order to identify individual potential constituent targets. FSDB (Feature Structure Data Base) is the component that stores semantic level information about the instantaneous state of a sketched diagram interpretation. FSDB can thus answer queries about probable targets around the centroid of a perceived pointing gesture. FSDB responds to spatial proximity queries issued by the 3DGesture agent by producing a list of target hypotheses ranked by likelihood, based on how close these elements are from the gesture centroid. Imprecision in the tracking will therefore not prevent a target from being identified, as constituents in the vicinity of the region of focus will be included for consideration. Precise within target pointing would be hard to achieve, given the physical constraints revealed by Fitt's law as discussed.

Whenever there is enough confidence that a tracked gesture is indeed referring to an element of a sketched diagram, the 3DGesture agent propagates a message informing the probable individual target elements within the diagram. A gesture might be made towards an empty region of a diagram as well, in which case the list of target hypotheses will be empty, or will include elements in the periphery with low likelihoods. Also informed are the gesture centroid and diameter of focus, calculated based on the perceived distance of the pointing participant to the board.

Awareness of this gesture then takes place as CharterUI displays both the circular region of focus and colors individual sketched elements according to their likelihood of being the target, using a color scheme that ranges from blue to red as the likelihoods increase

Notice that no special action is required from users to activate this functionality. The system pro-actively looks for opportunities to interpret naturally occurring conventional actions performed by the participants, making them visible at the participating sites. Pointing at multiple sites are displayed potentially concurrently. Conflict avoidance or resolution must then be handled by the participants through social protocols similar to the ones that might regulate speech turn taking.

The message informing of a gesture, which triggers the awareness display described above is also routed by the communication infrastructure to the multimodal fusion component - Multiparser. Multiparser attempts to combine a recognized gesture with speech produced by participants in the temporal vicinity of the gesture.

Fusion in Multiparser is based on a temporal chart parsing technique and is guided by grammar rules that specify operations and constraints expressed in terms of unification over feature structures [8]. The fusion process attempts to match semantic items. If successful, it produces semantic items that represent a potentially more robust interpretation that is based on the fused information. A threshold is used to remove from consideration those items that were not combined within a certain amount of time, so that the matching process is kept within bounds.

In the next section we describe the details of how the names of diagram constituents are detected from a multimodal streams of handwritten and spoken information and dynamically enrolled so that they can be used, among other things, to disambiguate gestures.

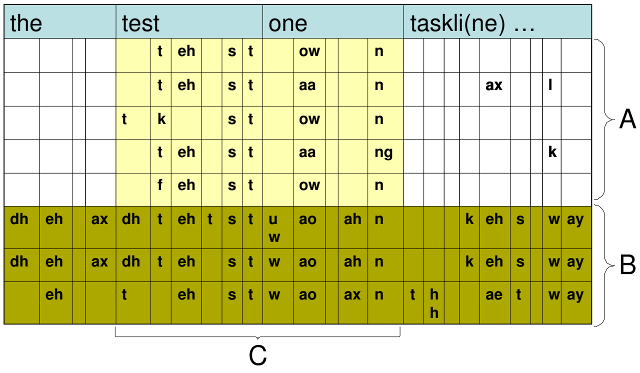
4.4 Speech based disambiguation

A disambiguation mechanism that might be used by participants of a collaborative session while gesturing is to refer to the intended item by its name, as expressed e.g. in labels attached to diagram constituents. The Charter Suite takes advantage of the fine-grained knowledge provided by the semantic model that it builds to explore opportunities for disambiguation via multimodal fusion of gesture, speech and sketched information exploring the capabilities of the Multimodal New Vocabulary Recognition (MNVR) technique [11].

In our baseline approach to MNVR we allowed new words to occur only within grammar-defined "carrier phrases" - a predefined constrained set of standardized phrases that the system is able to process. Recently we have generalized our approach, and now the system no longer requires the use of carrier phrases. In making this transition to a more general approach, we have kept the basic architecture of grammar-based language modeling implemented as a Recursive Transition Network [11] within a continuous speech recognizer (Carnegie Mellon University's Sphinx 2 recognizer).

However, instead of having only two grammars (i.e., the carrier grammar and the syllabic sub-grammar) we now use four grammars: sequence constrained syllable and phone grammars, and unconstrained syllable and phone grammars. These grammars each run within a separate instance of the speech recognizer, so an ensemble of four speech recognizers is used to perform an integrated phone level recognition pass. The results are coupled with letter-to sound hypotheses generated from the handwriting recognition, and with aspects of the context to arrive at the spelling, pronunciation and semantics of the handwritten and spoken input. In our system handwriting and speech occur together in this way when users are labeling constituents of the sketched diagram.

We employ an ensemble approach to phone recognition because each grammar-based recognizer yields different phone sequence interpretations. A single grammar-based recognizer could also yield multiple interpretations from a second-pass lattice search; however, without a stochastic model of phone sequence for English, this lattice search is typically either intractable, or if appropriate pruning is used to ensure tractability, then variations in resulting interpretations tend to be bunched toward the end. Using an ensemble of Viterbi first-pass recognizers allows full variation across each interpretation rather than just at the sequence ends. Our phone recognition rates are low (less than 70%). Our recognizer is not specifically optimized for phone level recognition. This is why there is considerable variation in interpretations across the ensemble of variously constrained phone recognizers; however, since we do know that each interpretation is of the same utterance, and we're using an articulatory-feature-based alignment mechanism (not described here, but which identifies the necessary class similarities between poorly recognized phone hypotheses) they can be reliably aligned. This results in phone hypotheses lattices against which letter-to-sound (LTS) interpretations of the handwriting letter-string hypotheses can be aligned. Figure 3 show a diagram that illustrates the alignment process.



the				test						one					taskli(ne) ...								
					t	eh		s	t		ow			n									
					t	eh		s	t		aa			n					ax			l	
				t	k			s	t		ow			n									
					t	eh		s	t		aa			ng								k	
					f	eh		s	t		ow			n									
dh	eh		ax	dh	t	eh	t	s	t	uw	ao		ah	n				k	eh	s		w	ay
dh	eh		ax	dh	t	eh		s	t	w	ao		ah	n				k	eh	s		w	ay
	eh			t		eh		s	t	w	ao		ax	n	t	hh			ae	t		w	ay

Figure 3: MNVR alignment. A) LTS from handwriter; b) ensemble speech recognition; C) extracted handwriter / speech matrix supporting dynamically built positional bigram for re-recognition pass.

Given a phone alignment across the speech and handwriting we extract the matrix segment covering the handwriting, dynamically build a positional bigram model over phone sequences, and then perform a second recognition pass over that segment using both a Viterbi first pass and a stochastically constrained second pass lattice search, yielding an n-best list of pronunciation interpretations of the handwriting.

All possible orthography/pronunciation combinations are made, scored by their handwriting and speech recognition scores (combined with phone and letter alignment distance measures), and ranked. The best scoring combinations are returned as chart constituent label hypotheses, with label semantics being determined by the spatial location of ink and the current state of the chart.

Once a label has been recovered, as described above, it is enrolled in a fifth grammar, optimized for word and phrase-spotting. This grammar is capable of recognizing the enrolled words or phrases when they are subsequently spoken, e.g. while participants are pointing to a diagram element.

Semantic elements representing pointing gestures are combined with the spoken labels recovered by the word spotting recognizer. Those items that are within the temporal window kept by the system are considered. The fusion rule iterates over the candidate constituents within the region of focus to the gesture, attempting to unify these constituents' labels with the recovered spoken references. The likelihoods of the target hypotheses are updated based on the likelihoods of the recovered spoken labels. As a result, targets that might have been considered less likely but were referenced by name are promoted and may become the most likely ones. The process is made tractable by the relatively constrained number of constituents that are expected to be within the narrow region of focus around a gesture centroid.

5. EVALUATION AND DISCUSSION

The precision of the pointing gestures was evaluated, to determine how appropriate it can in fact be in natural pointing situations such as the ones we aim to support. In this section we describe the preliminary results and discuss the implications in terms of actual usage.

5.1 Pointing accuracy

In this experiment, data was collected during five sessions, involving three users. Subjects were sitting at a desk at a distance that ranged from 3 meters to 3.5 meters from the interactive board. Subjects pointed five to ten times towards each of the milestones of the sketched diagram shown in Figure 1. This diagram contained an x-y axis, four milestones (labeled $m1$ to $m4$), and three task lines (labeled $T1$ to $T3$). Two of the datasets were discarded due to callibration problems.

We analyzed the accuracy of the pointing gestures towards milestones, the smallest constituents of a diagram, and therefore the hardest to acquire targets. Pointing was performed against a diagram sketched in actual ink, avoiding bias that might have been introduced by a display feedback mechanism.

Measurements were made in terms of the virtual drawing region of homogeneous size that is used internally by the system. This region is defined by a [(0,0),(1200,1000)] coordinate system (corresponding to top-left and bottom-right coordinates respectively), onto which actual interactive board coordinates are mapped. Under this coordinate system, milestones $M1$, $M2$, $M3$, $M4$ are found at coordinates (484, 232), (637,411), (770, 421), and (506,569) respectively.

The distances from the actual location of the gesture centroid as reported by the gesture recognition system was then compared to the known location of the targets. We define the normalized distance $D = \sqrt{(x - \bar{x})^2 + (y - \bar{y})^2}/dd$ between each milestone and pointing gesture as the ratio of the distance between the target centroid and the gesture centroid divided by the dimension $dd = \sqrt{1200^2 + 1000^2}$ of diagonal of the coordinate space; $(\bar{x}, \bar{y})$ represents the known coordinate of the milestone and (x, y) the coordinate of the centroid of each pointing gesture.

Figure 4 shows the resulting average normalized distances and its standard deviation for each data collection session. The average normalized distance for all gestures was 0.045 units, and is below 0.08 units in the worst case; the standard deviation is below 0.02.

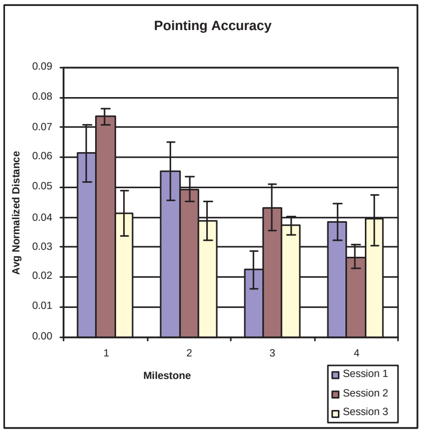


Figure 4: The average value and standard deviation of each milestone's normalized distance.

5.2 Effectiveness of actual target acquisition

The system's ability to discern among individual constituents of a sketched diagram made it possible in all cases to retrieve the intended target as the first element in a n-best list of target hypothesis. Recall that the system bases its pointing gesture hypotheses on the proximity of the gesture centroid to individual elements in its vicinity. These results indicate that the implemented mechanism is capable of distinguishing in practice among target elements that might be as close as 0.16 units (two times 0.08) of the drawing surface. The actual measurements on a board depend on the size of the board onto which the virtual coordinate space is mapped. On a 77" diagonal board such as the one used in the experiments, pointed to constituents within a 11" radius can be distinguished unambiguously; this figure drops to 7" for a 50" diagonal board. Gestured towards constituents that are drawn closer than that might cause the intended target to be listed as a secondary choice in the list of hypotheses. Speech-based disambiguation will then play a role in selecting the intended constituent from within this list, in case the user does indeed voice the name of a constituent.

The net effect of the systems' ability to handle ambiguous gestures whose centroid fall well off the target's center can be understood in terms of the Fitt's law index of difficulty $ID = \log_2(D/W + 1)$. Consider that a milestone has in average a width of 1.5", corresponding to approximately 0.2 units of the internal coordinate space when a 77" diagonal board is used; being able to acquire targets within the 0.18 region of focus represents a nine-fold increase in the practical target size. This in turns reflects on a reduced $ID' = \log_2(D/0.18 + 1)$, as opposed to the $ID = \log_2(D/0.02 + 1)$ corresponding to the intended target size.

5.3 Analysis of independently produced diagrams

To assess how appropriate the technique would in handling actual diagrams produced by users, we analyzed sixteen different diagrams produced by five different users over the course of four months. These diagrams were produced under a co-located collaborative scenario involving a single independent site that is evaluating the systems' support for collaboration as part of a larger project.

Given the limited accuracy of the tracking mechanism, we wanted to examine how much actual target disambiguation would need to rely on multimodal fusion of speech. We counted the number of milestones wf that would fall within a region of focus of 0.08 units around each of the n milestones of a diagram as illustrated in Figure 5. We then computed $d = (\sum_{i=1}^{n} wf_i)/n$, that indicates how dense a diagram is by showing the average number of milestones that fall within regions of focus around every milestone of a diagram. This density is directly related to the ambiguity of a gesture and therefore indicates how necessary multimodal disambiguation might be.

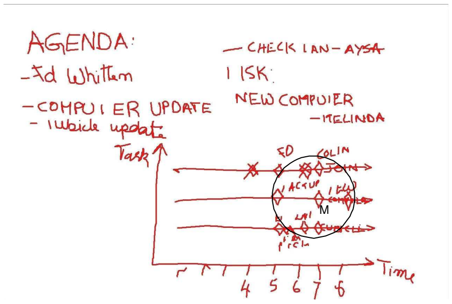


Figure 5: The number of milestones within the region of focus around a milestone M of a dense diagram $(wf_m = 6)$.

Of the sixteen collected diagrams, only three had d equal to 1, indicating no ambiguity. The highest d was 4.38 and the average d for all diagrams was 2.23, with a standard deviation of 1.02. This indicates that multimodal disambiguation is indeed required to correctly select a target in most cases, as the region of focus will potentially include multiple targets.

6. CONCLUSIONS AND FUTURE WORK

We showed how the distributed Charter can provide multimodal awareness services to groups of participants that may be distributed across different remote sites while working on a shared project schedule.

The Charter Suite embraces a policy of minimal intervention in support of unencumbered work practices. The system builds a semantic interpretation of an interaction by observing a multiuser multimodal dialogue. This detailed interpretation built by observation is the basis for the system's services.

The system's presence and services are made visible in cases where there are opportunities for mediation among distributed contexts. The functionality we explored was related to support for awareness diffusion of naturally occurring pointing gestures, made by participants towards shared sketched diagrams as they discuss aspects of the project while sitting at a table or standing from a distance.

We showed how the system's fine-grained knowledge of diagram constituents, coupled to speech based disambiguation can be used to resolve the ambiguity of large amplitude gestures, which pose intrinsic problems due to physical constraints, as expressed by Fitt's law [13].

This paper reports on the first steps of what we intend will result in a family of different related solutions, that combine perceptual and cognitive multimodal capabilities to affect collaboration. As short term next steps, we plan to:

- Explore the design concept along different dimensions, exploring other types of sensors and rendering mechanisms to support awareness of participants of unaltered, naturally occurring communicative behavior at remote sites.
- Conduct user studies to assess the effectiveness of the mechanisms during actual meetings. We are particularly interested in understanding how functionality such as the one we describe will be appropriated by users in practice, and how it might enhance their communication with remote participants.
- Continue to enhance the tracking and speech disambiguation mechanisms, based on current and further analysis of data originated from actual system use.

Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA or the Department of Interior-National Business Center (DOINBC). The author would like to thank Xiaoguang Li and Matt Wesson for designing and implementing the sketch/handwriting recognition module on which our system depended. The Charter Suite results from the cumulative work of many generations of developers, to which the authors are also indebted.

7. REFERENCES

[1] S. Agamanolis, A. Westner, and V. M. B. Jr. Reflection of presence: Toward more natural and responsive telecollaboration. In *Proc. SPIE Multimedia Networks*, volume 3228A, 1997.

[2] P. Besl and N. MacKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.

[3] S. A. Bly. A use of drawing surfaces in different collaborative settings. In *CSCW '88: Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, pages 250–256, 1988.

[4] P. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow. Quickset: Multimodal interaction for distributed applications. In *Proceedings of the Fifth ACM International Multimedia Conference*, 1997.

[5] D. Demirdjian, T. Ko, and T. Darrell. Constraining human body tracking. In *Proc. IEEE International Conference on Computer Vision*, Nice, France, 2003.

[6] J. Heiser, B. Tversky, and M. Silverman. Sketeches for and from collaboration. In J. S. Gero, B. Tversky, and T. Knight, editors, *Visual and spatial reasoning in design III*, pages 69–78. 2004.

[7] H. Ishii and M. Kobayashi. Clearboard: a seamless medium for shared drawing and conversation with eye contact. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 525–532, 1992.

[8] M. Johnston, P. Cohen, D. McGee, S. Oviatt, J. Pittman, and I. Smith. Unification-based multimodal integration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 1997.

[9] E. Kaiser, D. Demirdjian, A. Gruenstein, X. Li, J. Niekrasz, M. Wesson, and S. Kumar. Demo: A multimodal learning interface for sketch, speak and point creation of a schedule chart. In *Proceedings of the Sixth International Conference on Multimodal Interfaces (ICMI 2004)*, pages 329–330, 2004.

[10] E. Kaiser, A. Olwal, D. McGee, H. Benko, A. Corradini, X. Li, P. Cohen, and S. Feiner. Mutual disambiguation of 3d multimodal interaction in augmented and virtual reality. In *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI 2003)*, pages 12–19, 2003.

[11] E. C. Kaiser. Multimodal new vocabulary recognition through speech and handwriting in a whiteboard scheduling application. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 51–58, 2005.

[12] P. Luft, C. Heath, H. Kuzuoka, J. HindMarsh, K. Yamazaki, and S. Oyama. Fractured ecologies: Creating environments for collaboration. *Human-Computer Interaction*, 18(1 & 2):51–84, 2003.

[13] I. S. MacKenzie and W. Buxton. Extending fitts' law to two-dimensional tasks. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 219–226, 1992.

[14] D. Martin, A. Cheyer, and D. Moran. The Open Agent Architecture: A framework for building distributed software systems. *Applied Artificial Intelligence*, 13(1/2), 1999.

[15] O. Morikawa and T. Maesako. Hypermirror: toward pleasant-to-use video mediated communication system. In *CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 149–158, 1998.

[16] K.-I. Okada, F. Maeda, Y. Ichikawaa, and Y. Matsushita. Multiparty videoconferencing at virtual social distance: Majic design. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 385–393, 1994.

[17] S. Oviatt. Multimodal interfaces. In J. Jacko and A. Sears, editors, *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, chapter 14, pages 286–304. Lawrence Erlbaum Assoc., Mahwah, NJ, 2003.

[18] J. C. Tang and S. Minneman. Videowhiteboard: video shadows to support remote collaboration. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 315–322, 1991.