Collaborative Integration of Speech and 3D Gesture for Map-Based Applications

Andrea Corradini

Center for Human-Computer Communication
Oregon Health & Science University, Portland OR 97239, USA
andrea@cse.ogi.edu
http://www.cse.ogi.edu/CHCC

Abstract. QuickSet [6] is a multimodal system that gives users the capability to create and control map-based collaborative interactive simulations by supporting the simultaneous input from speech and pen gestures. In this paper, we report on the augmentation of the graphical pen input enabling the drawings to be formed by 3D hand movements. While pen and mouse can still be used for ink generation drawing can also occur with natural human pointing. To that extent, we use the hand to define a line in space, and consider its possible intersection point with a virtual paper that needs to be determined by the operator as a limited plane surface in the three dimensional space at the begin of the interaction session. The entire system can be seen as a collaborative body-centered alternative to the traditional mouse-, pen-, or keyboard-based multimodal graphical programs. Its potential applications include battlefield or crisis management, tele-medicine and other types of collaborative decision-making during which users can also be mobile.

1 Introduction and Related Work

Over the last decade, computers have moved from being simple data storage devices and calculating machines to become an assistant to the everyday' lives of many people. Web surfing, e-mailing and online activities made computers even extend to vehicle for human socialization, communication, and information exchange. However, for a few simple reasons computers are still far away from becoming an essential and indispensable component of our society.

In the first place, it is very difficult for computers to gain a sociological acceptance unless they will be able to seamlessly improve ordinary human activities. To that extent, before becoming essential, computers will have to understand, interact and dialogue with the real world around them the way people do. The interface for such computers will not be a menu, a mouse, a keyboard but instead a combination of speech, gestures, context and emotions. Human beings should not be required to adapt to technology but vice versa [4]. Moreover, "*…the personal computer has become a little too personal…*" [14]. In many organizations, people work in collaboration with each other by cooperating as a group. When much of an individual's work is team-related, software needs also to support interaction between users [11]. Yet, most current software is designed for the solely interaction between the user and computer.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3038, pp. 913–920, 2004.

Most of the work that people do requires some degree of communication and coordination with others. Unfortunately, the development of technology to support teamwork has proven to be a considerable challenge in practice. To achieve successful computer-mediated implementations, successful designs require: 1) social psychological insights into group processes and organizational coordination, 2) computer science insights into mechanisms to coordinate, share, communicate and organize information, and 3) Human-Computer Interaction (HCI) design insights. Current multimodal systems for Computer Supported Collaborative Work (CSCW) primarily focus on the last two factors. They typically consist of an electronic whiteboard and a video/audio teleconferencing system and maintain data manipulation disjoint from communication.

In the Speech, Text, Image and Multimedia Advanced Technology Effort (Stimulate) project [12], researchers used a combination of prototype technologies to study how people interact with computers employing sight, sound, force-feedback and touch as multiple modes of communicating. In a military application, once a user tell the computer to create a camp, the computer places the new camp where the user's eyes are pointed or where the user points with the glove and then respond that it has created the camp. Users can work together from different locations through a standard Internet connection or other type of network. QuickTurn [13] is a DARPA-funded project aiming to build a collaborative environment for multi-source intelligence analysis. It provides support for multimodal speech/gesture/mouse interaction, access to mediated databases, maps, image tools and the capability to share information between system components and system users in several collaboration modalities. In [3], a map-based application for travel planning domain has been proposed. It runs on a handheld PDA, can access several external data sources (e.g. the web) to compensate for power and data storage limitednesses of PDAs, and makes use of a synergistic combination of speech, gesture and handwriting modalities. Users can draw graphical gestures or write natural language sentences on a map display and issue spoken utterances. These inputs or their combination are deployed to give information about objects of interest, such as hotels, restaurants etc. Rasa [20] is a multimodal environment with visual perceptual input, which augments paper-based tools for military command and control. With Rasa users make use of familiar tools and procedures, creating automatic couplings between physical objects (i.e. maps and Post-it notes) and their counterparts in the digital world. This in turn, allows the users to manipulate paper artifacts, and accordingly control digital objects in the command and control systems.

All of these systems perform interactive collaborative tasks while employing speech and pen gesture recognition. However, what distinguishes our system from those is its extension, rather than replacement, to human gesture recognition and free hand movements (i.e. without the need of a pen or keyboard) to input system commands. To this extent, we built up on top of QuickSet [6,19], a prototype system that was developed at our Department after extensive research efforts in the area of multiagent architectures and multimodal collaborative HCI.

2 Augmenting QuickSet for CSCW and Pervasive Computing

The idea behind the augmentation of QuickSet is based on a vision. Imagine one single command center connected with several single remote operators each keeping track of the events on a different area of a common monitored area [24]. The remote operators provide local information for the central decision center to be evaluated. They use voice commands, 3D gestures and pen marks to add multimodal annotations regarding their local environment on a shared map and to collaborate with the intelligence specialists at the operative center as the local situation evolves. In the event of command center failure, any remote operator can take over and become command center while the interaction with the other operators continues. Users are allowed to move about and concentrate on different aspects/displays of the interface at will, without having to concern themselves with proximity to the host system's.

QuickSet is a collaborative distributed system that runs on several devices ranging from desktops, wireless hand-held PCs to wall-screen display and tablets. Users are provided with a shared map of the area or region on which a simulation is to take place. By using a pen to draw and simultaneously issuing speech commands, each user can lay down entities on the map at any desired location and give them objectives. Digital ink, view and tools are shared among the participants to the simulation session and ink can be combined with speech to produce a multimodal command.

Collaboration capabilities are ensured by a component-based software architecture that is able to distribute tasks over the users. It consists of a collection of agents, which communicate through the Adaptive Agent Architecture [18] that extends the Open Agent Architecture [5]. Any agent registers with a blackboard both the requests for actions it is put into charge and the kind of messages it is interested in. Agents communicate by passing Prolog-type ASCII strings via TCP/IP to the facilitator. This ladder then, unifies messages with agents' capabilities and forwards it to other agents according to the registration information. Collaboration takes place anytime at least two agents produce or support common messages.

There is empirical evidence of both user preference for and task performance advantages of multimodal interfaces compared to single mode interfaces in both general [22] and map-based applications [23]. Speech is the foremost modality, yet gestures often support it in interpersonal communication. Especially when spatial content is involved, information can be conveyed more easily and precisely using gesture rather than speech. Electronic pens convey a wide range of meaningful information with a few simple strokes. Gestural drawings are particularly important when abbreviations, symbols or non-grammatical patterns are required to supplement the meaning of spoken language input. However, the user must always know to which interface the pen device is connected, and switch pens when changing interfaces. This does not allow for pervasive and transparent interaction, as the user gets mobile.

We have extended QuickSet to accept 'ink' input from 3D hand movements independent of any pen device (see Fig. 1). To give input to our hand drawing system, the user attaches four Flock of Birds (FOB) [1] magnetic field tracker sensors; one on the top head, one on the upper arm to register the position and orientation of the humerus, one on the lower arm for the wrist position and lower arm orientation, and finally one on the top of hand (see [9] for details). The data from the FOB are delivered at a frequency of approximately 50Hz via serial lines, one for each sensor, and is processed in real time by a single SGI Octane machine. Because the FOB

devices uses a magnetic field that is affected by metallic objects, and the laboratory is constructed of steel-reinforced concrete, the data from the sensors is often distorted. As a result, the data is processed with a median filter to partially eliminate noise.

To facilitate our extended interface we have created a geometrical system to deal with relationships between entities in our setting. Such relationships are between both manually-entered screen positions (for drawing and visualization) and the perceptually-sensed current hand position and orientation (for recognition of certain meaningful 3D hand movements). This geometrical system is responsible for representing appropriately the information conveyed through hand interaction and automatically selecting the display to which the user's gesture pertains in the multi-display setting.

The system needs to know what display regions the users' 3D gestures can pertain to. Therefore, before the system is started for the first time, the system deployers have to manually set the regions the users will be able to paint in. The chosen regions will typically be a wall screen, a tablet or a computer screen on which the shared map or maps have been projected (see Fig. 1). The deployer accomplishes this by pointing at three of the vertices of the chosen rectangle for each painting region. However, since this procedure must be done in the 3D space, the deployer has to gesture at each of the vertices from two different positions. The two different vectors are triangulated to select a point as the vertex. In 3D space, two lines will generally not have an intersection, so we use the point of minimum distance from both lines.

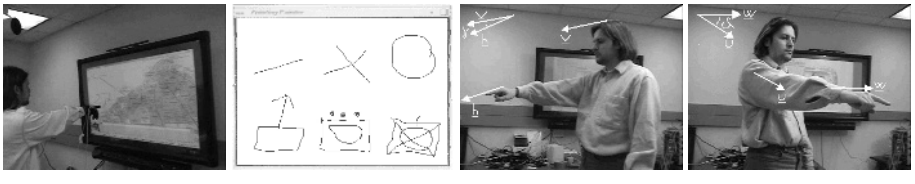


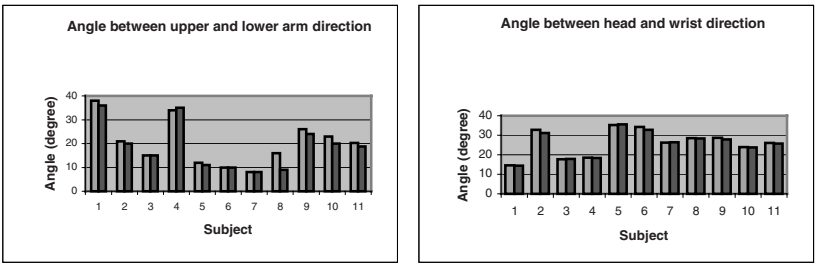
Fig. 1. *(from left to right)* Operating on a wall screen from a distance of approximately 2m; free-hand military symbols with increasing drawing complexity; imaginary head/wrist and upperarm/wrist pointing direction at gestural stroke

3 Multimodal Input

3.1 Gestural Input

Currently, the system supports the recognition of four kinds of gestures: pointing gestures, hand twisting about the index finger, rotating the hand about the wrist, and pushing with the palm up or down. Recognition is based on a body model [9] we use to track human movements and a set of rules for those movements, that extend the rules in [8]. These rules were derived from an evaluation of characteristic patterns we identified by analyzing sensor profiles of the movements underlying the various gestures.

Table 1. Average upper-/lower-arm angle (left) and head/wrist angle (right) for each subject (bars 1-10) along with the average of these values (bar 11). Blue bars have been determined using all collected data, red bars after dropping 10% (both higher and lower 5%) of the data



An Empirical Study. Pointing is probably the most common kind of gesture and human beings can easily recognize it. It is an intentional behavior controlled both by the pointer's eyes and by muscular sense (proprioception), which aims at directing the listener's visual attention to either an object or a direction [10,17,21]. Pointing gesture recognition by computers is much more difficult and in fact to date there is no system able to reliably recognize such class of gestures under any conditions.

We conducted an empirical experiment to expand the set of rules, we used in a previous gesture recognizer [8], wrt the geometrical relationships within our body model while a pointing gesture occurs. We invited ten subjects to both point at and describe any object in our laboratory. They were attached the four FOB sensors and then instructed to ignore anyone in the laboratory. Furthermore, they were required to keep the whole body in a kind of 'frozen' state at the end of the stroke for any pointing gesture performed (see Fig. 1). Once in that position, by averaging the FOB reports, we determined the pointing direction, the direction of the upper arm, that of the wrist and the direction of the eyes (see Table 1). The subjects were requested to perform one hundred pointing at as many objects or entities within the room, on the floor or the ceiling. Objects or entities were allowed to be pointed at more than once.

Deictic Gestures. Based on the above empirical study and on the partial analysis of collected gesture data [9], we characterize a pointing gesture as: 1) a hand movement between two stationary states, 2) lasting between 0.5 and 2.4 seconds, 3) whose dynamic patterns are characterized by a smooth and steady increase/decrease in the spatial component values, 4) whose head direction and pointing direction (see Fig. 1) form an angle below a heuristically determined threshold, and 5) whose pointing direction and the direction determined by the upper arm (see Fig. 1) forms an angle below some certain threshold. The first condition is quite artificial as it was introduced by us, rather than extrapolated from the data, for facilitating data stream segmentation. In this way, start and stroke phases of the pointing gestures can be used to trigger pen up/down events while for ink production. The fourth condition about the angular value between pointing and head direction, implicitly assumes that head orientation is a reliable indicator of the direction of the user's attention. Estimating where a person is looking at based on his solely head orientation is a plausible simplification [25] used to determine the focus of attention of the user without having to perform eye gaze tracking. The experimental study reported in the previous

subsection provides a useful estimate for the threshold used for checking this condition and the fifth one as well.

In natural human pointing behavior, the hand and index finger are used to define a line in space, roughly passing through the base and the tip of the index finger. Normally, this line does not lie in any target plane, but it may intersect one at some point. When a user interacts with the system on any shared map, it is this point of intersection that we aim to recover from within the FOB's transmitter coordinate system. A pointing gesture phase is dealt with only if the imaginary line described by the sensor in the space intersects some of the virtual papers. Since in the context of ink production any recognition errors lead frequently to incorrect messages we implemented a substitute way to trigger pen down, and pen up events using a PinchGlove [2]. At the begin of the interaction, the user can choose which method to use. Drawings are passed on from the remote system to the central command and control for recognition.

Rotational Gesture. A similar rule-based analysis of hand twisting, hand rotating and hand pushing can be given using the quaternion components provided by the sensor.

All three kinds of gesture are very similar rotational movements, each one occurring about different orthogonal axes. To recognize such gestures, we analyze the hand rotation information using the quaternion components provided by the sensor. We characterize a rotational gesture as a hand movement for which: 1) the gesture takes place between two stationary states, 2) it is a rotation, 3) the unit vector along the axis of rotation is constant over the movement, 4) the rotation is smooth and takes place about the axis of rotation by at least N (a tunable parameter) degrees, 5) the upper arm does not move during the rotation, and 6) the hand moves during the rotation.

According the axis of rotation considered, each of the three rotational gestures can be recognized. While in the current system hand pushing and hand rotating are not attached to any command, a hand twisting gesture is required anytime the user wishes to pan over the map. In such case, the pointing direction with respect to the center of the virtual paper determines the direction of the panning. An additional twisting gesture makes the system to switch back to normal mode.

3.2 Speech and Gesture Integration

Voice is an essential component of natural interaction. Spoken language allows operators to keep their visual focus on the map environment, leaving their hands free for further operations.

For speech recognition, we use Dragon 4.0, a commercial off-the-shelf product. Dragon is a Microsoft SAPI 4.0 compliant, speaker independent speech engine. Any user can immediately interact via voice without having to train the system for his/her voice. Spoken utterances are sensed either by microphones in the room or by (wireless) microphones that users wear. Sensed utterances are sent to the speech recognition engine, which receives the audio stream and produces an *n*-best list of textual transcripts, constrained by a grammar supplied to the speech engine upon startup. These are parsed into meaningful constituents by the natural language processing agent, yielding an *n*-best list of alternative interpretations ranked by their associated speech recognition probabilities. Speech recognition operates in a click-to-speak microphone mode, i.e. the microphone is activated when a pen down event is trigger.

Concerning modality fusion, we exploit the original version of QuickSet that already provides temporal integration of speech and gesture. These modes are constrained to either have overlapping time intervals or to have the speech signal onset occur within a time window of up to 4 seconds [23] following the end of the gesture. The multimodal integrator determines which combinations of speech and gesture interpretations can produce an actual command, using the approach of [16]. The basic principle is that of typed feature structure unification [15], which is derived from term unification in logic programming languages. Unification rules out inconsistent information, while fusing redundant and complementary information via binding of logical variables that are values of 'matching' attributes. The matching process also depends on a type hierarchy. A set of multimodal grammar rules specify, for this task, which of these speech and gesture interpretations should be unified to result in a command.

4 Conclusion and Future Directions

This report describes a working implementation of a 3D hand gesture recognizer to extend the existing digital-ink input capabilities of a fully functional real-time, multimodal speech/pen architecture. The architecture presented is flexible and easily extensible to provide more functionality. It exploits QuickSet's capabilities for recognition and fusion of distinct input modes while maintaining a history of the collaboration.

While drawing using free hand movements allows for non-proximity and transparency to the interface, creating detailed drawings is not easy as human pointing is not accurate [7]. Executing detailed drawings takes training and practice on the user's part, and relies on a precise calibration of both the tracking device and the geometrical model. The accuracy of drawings decreases with both increasing symbol complexity (see Fig. 1) and increasing distance from the virtual paper, thus this might cause lower recognition rates. We use the PinchGlove to signal pen-up/pen-down. As speech recognition in non click-to-speak mode in QuickSet becomes more reliable, such pen-up/pen-down gestures could also be entered by voice. Alternatively PinchGlove signaling could be replaced by defining a specific hand shape to trigger these events as an extension to our current 3D gesture recognition system.

For right now, hand rotation and pushing gesture are not attached to any command because of the lack of a natural intuitive mapping between these movements and any entity behavior on the map. For instance, while an entity can be put on, moved about, and removed from the map, it cannot be given an orientation (e.g. to visually indicate the direction of movement of a unit on the map) i.e. it is always displayed using the same symbol. Once such an entity will be given an orientation, it will be natural and straightforward to attach e.g. a hand rotation to a 2D rotation of the entity on the map.

Acknowledgments. This research has been supported by the ONR grants N00014-99-1-0377, N00014-99-1-0380 and N00014-02-1-0038. We are thankful to Phil Cohen, Ed Kaiser, and Richard Wesson for programming support and fruitful suggestions.

References

1. http://www.ascension-tech.com
2. http://www.fakespace.com
3. Cheyer A., and Julia, L., “Multimodal Maps: An Agent-based Approach", Multimodal Human-Computer Comm., Lecture Notes in AI #1374, Springer Verlag, 1998, 111-121
4. Coen, M.H., “The Future of Human-Computer Interaction or How I learned to stop worrying and love My Intelligent Room”, *IEEE Intelligent Systems*, March/April 1999
5. Cohen, P.R., et al., "An Open Agent Architecture", Working Notes of the AAAI Spring Symposium on Software Agents, Stanford, CA, March 21-22, 1994, 1-8
6. Cohen, P.R., et al., "QuickSet: Multimodal Interaction for Distributed Applications", Proceeding of the 5th International Multimedia Conference, ACM Press, 1997, 31-40
7. Corradini, A., and Cohen, P.R., "Multimodal speech-gesture interface for hands-free painting on virtual paper using partial recurrent neural networks for gesture recognition", Proc. of the Int’l Joint Conf. on Artificial Neural Networks, Honolulu (HI), 2002, 2293-2298
8. Corradini A., et al., “A Map-based System Using Speech and 3D Gestures for Pervasive Computing”, Proc. IEEE Int’l Conf. on Multimodal Interfaces, 2002, 191-196
9. Corradini A., and Cohen P.R., “On the Relationships Among Speech, Gestures, and Object Manipulation in Virtual Environments: Initial Evidence”, Proc. of Int’l CLASS W’shop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Sys., 2002,.52-61
10. Efron D., “Gesture, Race and Culture”, Mouton and Co., 1972
11. Ellis, C.A., Gibbs, S., and Rein, G., "Groupware: Some Issues and Experiences", *Communications of the ACM*, 34(1):39-58, 1991
12. Flanagan, J., et al., "NSF – STIMULATE: Synergistic Multimodal Communication in Collaborative Multiuser Environments", Annual Report, 1998
13. Holland, R., "QuickTurn: Advanced Interfaces for the Imagery Analyst", DARPA/ITO Intelligent Collaboration & Visualization Program PI Meeting, Dallas, TX, Oct. 10, 1996
14. Johansen, R., ”Groupware: Computer Support for Business Teams’’, The Free Press, 1988
15. Johnston, M., et al., “Unification-based multimodal integration”, Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, Madrid, Spain, 1997
16. Johnston M., “Multimodal Language Processing”, Proceedings of the 5th International Conference on Spoken Language Processing, Sydney, Australia, 1998
17. Kendon A., “The Biological Foundations of Gestures: Motor and Semiotic Aspects”, Lawrence Erlbaum Associates, 1986
18. Kumar, S., et al., “The Adaptive Agent Architecture: Achieving Fault-Tolerance Using Persistent Broker Teams”, Proc. of 4th Int’l Conf. on Multi-Agent Systems, 2000, 159-166
19. McGee, D.R., Cohen, P.R, "Exploring Handheld, Agent-based Multimodal Collaboration", Proceedings of the Workshop on Handheld Collaboration, Seattle, WA, 1998
20. McGee, D. R., et al., "A Visual Modality for the Augmentation of Paper", Proceedings of the Workshop on Perceptive User Interfaces, ACM Press: Orlando, FL, Nov. 14-16, 2001
21. McNeill, D., “Language and Gesture: Window into Thought and Action”, David McNeill, editor, Cambridge: Cambridge University Press, 2000
22. Mellor, B.A., et al., "Evaluating Automatic Speech Recognition as a Component of Multi-Input Human-Computer Interface", Proc. of Int’l Conf. on Spoken Lang. Processing, 1996
23. Oviatt, S.L., "The Multimodal Interfaces for Dynamic Interactive Maps", Proc. of the Conference on Human-Factors in Computing Systems, ACM Press, 1996, 95-102
24. Sharma, R., et al., “Speech-Gesture Driven Multimodal Interfaces for Crisis Managment”, Proceedings of IEEE, special issue on Multimodal Human-Computer Interface, 2003
25. Stiefelhagen, R., “Tracking Focus of Attention in Meetings”, Proc. IEEE Int’l Conference on Multimodal Interfaces, Pittsburgh, PA, USA, October 14-16, 2002, 273-280