The Efficiency of Multimodal Interaction for a Map-based Task

November 17, 1999

Topic Area: Spoken Language and multimodal systems, Evaluation of performance of complete NLP systems

Key Words: multimodal interaction, speech, gesture, natural language, graphical user interface

Authors: Philip R. Cohen (pcohen@cse.ogi.edu) & David McGee (dmcgee@cse.ogi.edu), Oregon Graduate Institute of Science & Technology, and Josh Clow of Microsoft Corporation.

Abstract

This paper compares the efficiency of using a standard direct-manipulation graphical user interface (GUI) with that of using the QuickSet pen/voice multimodal interface for supporting a military task. In this task, a user places military units and control measures (e.g., various types of lines, obstacles, objectives) on a map. Four military personnel designed and entered their own simulation scenarios via both interfaces. Analyses revealed that the multimodal interface led to an average 3.5-fold speed improvement in the average entity creation time, including all error handling. The mean time to repair errors also was 4.3 times faster when interacting multimodally. Finally, all subjects reported a strong preference for multimodal interaction. These results indicate a substantial efficiency advantage for multimodal over GUI-based interaction during map-based tasks.

1 INTRODUCTION

Nearly two decades ago at ACL'80, Professor Ben Shneiderman challenged the field of natural language processing as follows:

> In constructing computer systems which mimic rather than serve people, the developer may miss opportunities for applying the unique and powerful features of a computer: extreme speed, capacity to repeat tedious operations accurately, virtually unlimited storage for data, and distinctive input/output devices. Although the slow rate of human speech makes menu selection impractical, high-speed computer displays make menu selection an appealing alternative. Joysticks, light pens or the "mouse" are extremely rapid and accurate ways of selecting and moving graphic symbols or text on a display screen. Taking advantage of these and other computer-specific techniques will enable designers to create powerful tools without natural language commands [20, p. 139]

He also challenged us to go beyond mere claims, but to demonstrate the benefits of natural language processing technologies empirically. Since then, not only has there been a long period of unprecedented innovation in hardware, software architectures, speech processing, and natural language processing, but NLP research has also embraced empirical methods as one of its foundations. Still, we have yet to defend claims empirically that technologies for processing natural human communication are more efficient, effective, and/or preferred, than interfaces that are best viewed as "tools," especially interfaces involving a direct manipulation style of interaction. The present research attempts to take a small step in this direction.

In fact, it has often been claimed that spoken language-based human-computer interaction will not only be more natural but also more efficient than keyboard-based interaction. Many of these claims derive from early modality comparison studies [1], which found a 2-3 fold speedup in task performance when people communicated with each other by telephone vs. by keyboard. Studies of the use of some of the initial commercial speech recognition systems have reported efficiency gains of approximately 20% - 40% on a variety of interactive hands-busy tasks [10] compared with keyboard input. Although these results were promising, once the time needed for error correction was included, the speed advantage of speech often evaporated [18][1]. A recent study of speech-based dictation systems [9] reported that dictation resulted in a slower and more errorful method of text creation than typing. From such results, it is often concluded that the age of spoken human-computer interaction is not yet upon us.

Most of these studies have compared speech with typing. However, in order to affect mainstream computing, spoken interaction would at a minimum need to be found to be superior to graphical user interfaces (GUIs) for a variety of tasks. In an early study of one component of GUIs, Rudnicky [18] compared spoken interaction with use of a scroll bar, finding that error correction wiped out the speed advantages of speech, but users still preferred to speak. Pausch and Leatherby [17] examined the use of simple speaker-dependent discrete speech commands with a graphical editor, as compared with the standard menu-based interface. With a 19-word vocabulary, subjects were found to create drawings 21% faster using speech and mouse than with the menu-based system. They conjectured that reduction in mouse-movement was the source of the advantage. In general, more research comparing speech and spoken-language-based interfaces with graphical user interfaces still is needed.

[1] See also [6, 10] for a survey of results.

We hypothesize that one reason for the equivocal nature of these results is that speech is often being asked to perform an unnatural act —the interface design requires people to speak when other modalities of communication would be more appropriate. In the past, strengths and weaknesses of various communication modalities have been described [2, 6, 13], and a strategy of developing *multimodal* user interfaces has been developed using the strengths of one mode to overcome weaknesses in another. Interface simulation studies comparing multimodal (speech/pen) interaction with speech-only have found a 35% reduction in user errors, a 30% reduction in spoken dysfluencies (which lead to recognition errors), a 10% increase in speed, and a 100% user preference for multimodal interaction over speech-only in a map-based task [14]. These results suggest that multimodal interaction may well offer advantages over GUI's for map-based tasks, and may also offer advantages for supporting error correction during dictation [16, 19].

In order to investigate these issues, we undertook a study comparing a multimodal and a graphical user interface that were built for the same map-based task[2].

1.1 Study[3]

This study compares a direct-manipulation graphical user interface with the QuickSet pen/voice multimodal interface [4] for supporting a common military planning/simulation task. In this task, a user arrays forces on a map by placing icons representing military units (e.g., the 82nd Airborne Division) and "control measures," (e.g., various types of lines, obstacles, and objectives). A

[2] A high-performance spoken language system was also developed for a similar task [12] but to our knowledge it was not formally evaluated against the relevant GUI.

[3] A case study of one user was reported in [3]. This paper reports a fuller study, with different users, statistical analyses, and an expanded set of dependent measures (including error correction).

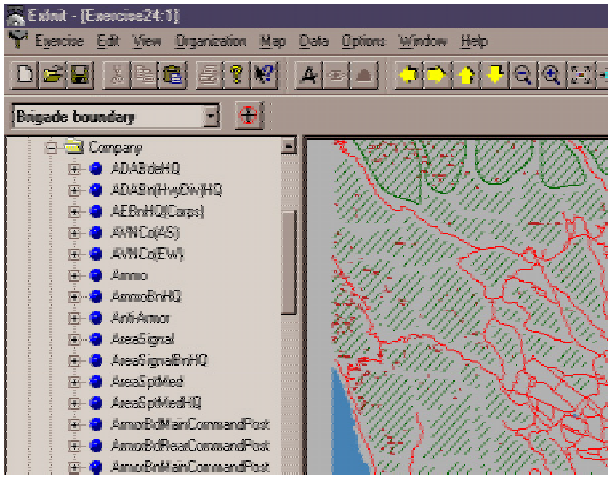

Figure 1. The ExInit GUI

shared backend application subsystem, called ExInit, takes the user specifications and attempts to decompose the higher echelon units into their constituents. It then positions the constituent units on the map, subject to the control measures and features of the terrain.

1.2 ExInit's GUI

ExInit provides a direct manipulation GUI (built by MRJ Corp.) based on the Microsoft Windows suite of interface tools, including a tree-browser, drop-down scrolling lists, buttons (see Figure 1). Many military systems incorporate similar user interface tools for accomplishing these types of tasks (e.g., ModSAF [7]). The tree-browser is used to represent and access the collection of military units. The user employs the unit browser to explore the echelon hierarchy until the desired unit is located. The user then selects that unit, and drags it onto the map in order to position it on the terrain. The system then asks for confirmation of the unit's placement. Once confirmed, ExInit invokes its deployment server to decompose the unit into its constituents and position them on the terrain. Because this is a time-consuming process depending on the echelon of the unit, only companies and smaller units were considered.

To create a linear or area control measure, the user pulls down a list of all control measure types, then scrolls and selects the desired type.[4] Then the user pushes a button to start entering points, selects the desired locations, and finally clicks the button to exit the point creation mode. The user is asked to confirm that the selected points are correct, after which the system connects them and creates a control measure object of the appropriate type.

Finally, there are many more features to this GUI, but they were not considered for the present comparison. The system and its GUI were well-received by the client, and were used to develop the largest known distributed simulation (60,000 entities) for the US Government's Synthetic Theater of War program (STOW).

1.3 QuickSet's Multimodal Interface

QuickSet is a multimodal (pen/voice) interface for map-based tasks. With this system, a user can create entities on a map by simultaneously speaking and drawing [4]. With pen-based,

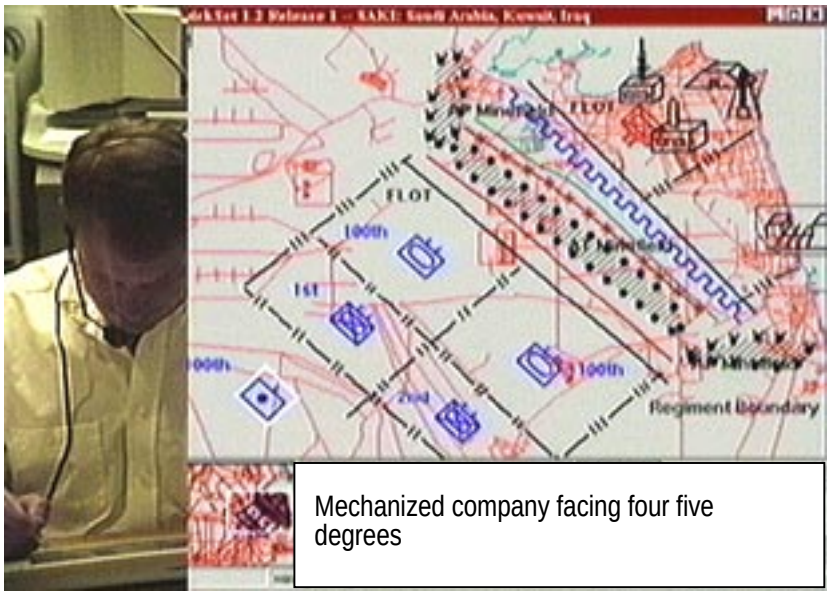

Figure 2. QuickSet

[4] There were 45 entries, viewable in a window of size 9. The entries consisted of linear features (boundaries, obstacles, etc.), then areas.

spoken, or multimodal input, the user can annotate the map, creating points, lines, and areas of various types (see Figure 2). In virtue of its distributed multiagent architecture, QuickSet operates in various heterogeneous hardware configurations, including wearable, handheld, desktop, and wall-sized. Moreover, it controls numerous backend applications, including 3D terrain visualization [5] military simulation, disaster management [15] and medical informatics.

The system operates as follows: When the pen is placed on the screen, the speech recognizer is activated, thereby allowing users to speak and gesture simultaneously. For this task, the user either selects a spot on the map and speaks the name of a unit to be placed there (e.g, "mechanized company"), or draws a control measure while speaking its name (e.g., "phase line green"). In response, QuickSet creates the appropriate military icon on its map and asks for confirmation. Speech and gesture are recognized in parallel, with the speech interpreted by a definite-clause natural language parser. For this study, IBM's Voice Type Application Factory, a continuous, speaker-independent speech recognition system, was used with a bigram grammar and 662-word vocabulary. In general, analyses of spoken language and of gesture each produce a list of interpretations represented as typed feature structures [8]. The language supported by the system essentially consists of complex noun phrases, including attached prepositional phrases and gerunds, and a small collection of sentence forms. Utterances can be just spoken, or coupled with pen-based gestures. Multimodal integration searches among the set of interpretations for the best joint interpretation [8, 22], which often disambiguates both speech and gesture simultaneously [15]. Typed feature structure unification provides the basic information fusion operation. Taking advantage of the system's mutual disambiguation capability, QuickSet confirms its interpretation of the user input after multimodal integration [11], thereby allowing the system to correct recognition and interpretation errors. If the result is acceptable, the user

needs only to proceed; only unacceptable results require explicit disconfirmation. Finally, the multimodal interpretation is sent directly to the ExInit deployment server, effectively bypassing the ExInit GUI.

2 PROCEDURE

The study involved four subjects who were retired US military domain experts, including a US Army National Guard Brigadier General, a US Army Reserve Major, a US Marine Corps Captain, and a US Army communications specialist. Each of the subjects was a frequent computer user, and all had familiarity both with GUIs built around the Microsoft user interface tools as well as with pen-and-paper based drawing of unit symbology and diagrams on maps. Not having used either system before, the subjects were given 30 minutes to learn the ExInit GUI, and the same amount of time to learn QuickSet. The subjects created scenarios of their own design, using entities common to both systems, first on paper, then with each of the two systems. The scenarios had 8-21 units, and 9-33 control measures. The order of interface styles was counterbalanced across subjects in this within-subject design. The systems were run on a Pentium Pro 200MHz computer with an Input Technologies 14” color flat-panel display. Stylus input was used for QuickSet, and keyboard and mouse were employed with the GUI.

The mean time needed for each expert subject to create and position a unit or control measure was calculated for each interface. The time to create an entity began when the mouse entered the relevant interface tool or the time when the microphone was engaged by placing the pen on the map. Mouse “travel” time to the desired interface tool was not included because the pen could not be tracked when it was out of the sensitivity range of the digitizer. Timing ended when the system asked for confirmation of its impending action. Separate creation time calculations were

made for units and control measures because the GUI employed different user interface tools for each. Also, whereas the set of QuickSet units was a subset of the units available to the GUI, the set of control measures was identical for QuickSet and the GUI.

The entity creation times reported in this study include correction of all errors needed for both QuickSet and the GUI. Error correction time was accumulated for each attempt until a user confirmation (explicit or implicit) was achieved, or until the entire entity creation attempt was aborted. Only 4 multimodal interactions (total = 20.7 secs.) and 1 GUI interaction (total = 43.2 secs.) were aborted. Errors for QuickSet included out-of-vocabulary or grammar, procedural errors (e.g., not clicking on the map), disfluencies, and recognition errors. For the GUI, errors involved failure to enter or leave drawing mode, selecting the wrong unit in the browser, disconfirming, etc. Overall, QuickSet provided an 88.5% successful understanding rate.

By saying "multiple Xs," the QuickSet user could enter a “mode” in which he was creating an entity of type X (e.g., a mechanized company). To support this process, the system stored a discourse referent that is unified with subsequent input. The user needed only to say “here” and touch the screen in order to create another entity of that type at that location.[5] In these cases, the time taken to enter the mode was amortized over the entities created. Likewise, the time taken to open the unit browser to show the desired unit was amortized over the units of that type created before the browser was again scrolled.

[5] In general, the user could at that point say anything that would unify with the type of entity being created, such as “facing two two five degrees in defensive posture.” This would add additional data to the type of entity being created. Similar data could be added via the GUI, but it required interacting with a dialogue box that was only created after the unit’s constituents were loaded (a time-consuming operation). Since QuickSet users could supply the data before the constituents were loaded, it was deemed more fair to ignore this QuickSet capability even though it speeds up multimodal interaction considerably, and employs more extensive natural language processing.

3 RESULTS

Analyses revealed that multimodal interaction resulted in a 3.7-fold speed increase in creating units compared to the GUI, paired t-test, $t(3) = 5.791$, $p < 0.005$, one-tailed. In addition, it provided a 3.3-fold increase in creating control measures paired t-test $t(3) = 8.298$, $p < 0.002$, one-tailed (see Table I).[6] Much of this speed differential can be traced to the need to browse the echelons of the US military, scrolling long lists of units with the GUI (e.g., 126 units are in the list of US Army companies), followed by a separate dragging operation to position the selected unit. In contrast, QuickSet users specified the type of entity directly, and supplied its location in parallel. Likewise, the speed differential for the control measures may be attributed to the user's ability to both draw and speak in parallel, where the GUI required separate actions for going into and out of drawing mode, for selecting the type of control measure, and for selecting appropriate points on the map.

Although there were fewer errors on average when using the direct manipulation GUI, they were not significantly fewer than when interacting multimodally. In contrast, the time needed to repair an error was significantly lower when interacting multimodally than with the GUI, paired t-test, $t(3) = 4.703$, $p<0.009$, one-tailed. On balance, the same users completing the same tasks spent 26% more total time correcting errors with the GUI than with the multimodal interface.

[6] It should be pointed out that the paired t-test takes into consideration the number of subjects. Thus, these findings at these significance levels are particularly strong.

Expert Subject	Create Units		Create Control Measures		Repair Errors	
	MM	GUI	MM	GUI	MM	GUI
S1	8.4	25.6	6.5	27.5	12.9	49.3
S2	6.0	14.4	5.2	19.0	7.7	30
S3	6.3	27.2	11.0	24.3	11.6	56.1
S4	4.0	18.5	4.0	17.7	6.3	23.0
Means	6.2	21.4	6.7	22.1	9.6	39.6

Table I: Mean time in seconds required to create various types of entities and to repair errors when interacting multimodally versus with the ExInit GUI

The expert users were interviewed after the study regarding which interface they preferred and why. Multimodal interaction was strongly preferred by all users. Reasons cited included its efficiency and its support of precise drawing of linear and area features.

4 CONCLUSIONS

This study indicates that when the user knows what s/he wants, there can be substantial efficiency advantages of multimodal interaction over direct manipulation GUIs for a map-based taste. Despite having only four subjects, the results exhibited extremely strong statistical significance. These results using stand in contrast to prior research [6, 9, 10, 18] in which speed advantages of spoken input were washed out by the cost of correcting recognition errors. In the present study, not only was multimodal interaction substantially faster than GUI-based interaction, even including error correction times, error correction itself was six times more costly with a GUI than with multimodal interaction. These findings do not support those of Karat et al. [9] who found that for correcting errors in a dictation task, keyboard-mouse input led to a 2.3-fold speed increase over speech. Both sets of findings might be reconciled by noting that

advantages of any type of user interface, especially spoken and multimodal interaction, may be task dependent.

We attribute the findings here to the ability of multimodal interfaces to support parallel specification of complementary parts of a communicative act, as well as direct rather than hierarchical or scrolled access to types of entities. Moreover, because the user can employ each mode for its strengths s/he can offload different aspects of the communication to different human cognitive systems, leading to greater efficiency [21] and fewer user errors [13].

It might be claimed that these results apply only to this GUI, and that a different GUI might offer superior performance. First, it is worth noting that the same pattern of results were found for the two GUI elements (drop-down list and hierarchical browser). Thus, the results cannot simply be attributed to the misuse of a hierarchical tool. Second, we point out that this GUI was developed as a product, and that many military systems use very similar user interface tools for the same purposes (selecting units).[7] Thus, these results may have substantial practical impact for users performing this task. More generally, one study cannot establish results for all possible user interfaces. There will certainly be occasions in which a menu-based GUI will be superior to a multimodal interface – e.g., when the user does not in fact know what s/he wants and needs to browse. Other GUI interface tools, such as a search field with command completion, can be envisioned that would provide direct access. However, it is arguable that such an interface element belongs squarely to *graphical* user interfaces, but draws more on features of language. Also, it would require the user to type, even in circumstances (such as mobile usage) where typing would be infeasible. Given our philosophy of using each modality for its strengths, we

[7] In fact, a recent experiment by the US Marines had mobile combatants using small portable computers with a similar direct manipulation interface as they participated in field exercises. The user interface was generally regarded as the weakest aspect of the experiment.

believe multimodal and graphical user interfaces should be integrated, rather than cast as opposites. Finally, we would expect that these advantages of multimodal interaction may generalize to other tasks and other user interfaces in which selection among many possible options is required.

Obviously, a small experiment only illuminates a small space. But it should be clear that when current technologies are blended into a synergistic multimodal interface the result may provide substantial improvements on some types of tasks heretofore performed with graphical user interface technologies. We conjecture that the more we can take advantage of the strengths of spoken language technology, the larger this advantage will become. Future research should be searching for more such tasks, and developing more general toolkits that support rapid adaptation of multimodal technologies to support them.

5 ACKNOWLEDGEMENTS

This work was supported in part by the Information Technology and Information Systems offices of DARPA under multiple contract numbers DABT63-95-C-007 and N66001-99-D-8503, and in part by ONR grant N00014-95-1-1164. Many thanks to Jay Pittman for the ExInit integration, Michael Johnston for ExInit vocabulary and grammar development, Liang Chen for graphics and military symbology, Sharon Oviatt for advice in experimental analysis, and to our test subjects.

6 REFERENCES

1. Chapanis, A., Ochsman, R.B., Parrish, R.N., Weeks, G. D., Studies in interactive communication: I. The effects of four communication modes on the behavior of teams during cooperative problem solving. *Human Factors*, 1972. 14: pp. 487-509.

2. Cohen, P.R., Dalrymple, M., Moran, D.B., Pereira, F., Sullivan, J., Gargan, R., Schlossberg, J., and Tyler, S., Synergistic use of natural language and direct manipulation, in *Proc. of the Human-Factors in Computing Systems Con-ference (CHI'89)*. 1989, ACM Press: New York, pp. 227-234.

3. Cohen, P.R., Johnston, M., McGee, D., Oviatt, S., Clow, J., and Smith, I., The efficiency of multimodal interaction: A case study, in the *Proceedings of the 5th International Conference on Spoken Language Processing*, Sydney, Australia, 1998, 2: pp. 249-252.

4. Cohen, P.R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., Clow, J., QuickSet: Multimodal interaction for distributed applications, in *Proc. of the Fifth ACM International Multmedia Conference*, E. Glinert, Editor. 1997, ACM Press: New York. pp. 31-40.

5. Cohen, P.R., McGee, D., Oviatt, S., Wu, L., Clow, J., King, R., Julier, S., Rosenblum, L., Multimodal Interaction for 2D and 3D Environments. *IEEE Computer Graphics and Applications*, 1999. 19(4): pp. 10-13.

6. Cohen, P.R. and Oviatt, S.L., The Role of Voice Input for Human-Machine Communication. *Proc. of the National Academy of Sciences*, 1995. 92: pp. 9921-9927.

7. Courtemanche, A.J., Ceranowicz, A., ModSAF Development Status., in the *Proc. of the Fifth Con-ference on Computer Generated Forces and Behavioral Rep-resentation*, Orlando, 1995, Univ. of Central Florida, pp. 3-13.

8. Johnston, M., Cohen, P. R., McGee, D., Oviatt, S. L., Pittman, J. A., Smith., I. Unification-based multimodal integration., in the *Proc. of the 35th Annual Meeting of the Association for*

Computational Linguistics (ACL) and 8th Conference of the European Chapter of the ACL, 1997, pp. 281-288.

9. Karat, C., Halverson, C., Horn, D., and Karat, J., Patterns of entry and correction in large vocabulary continuous speech recognition systems, in the *Proc. of Human Factors in Computing Systems*, New York, 1999, ACM Press, pp. 568-575.
10. Martin, G.L., The utility of speech input in user-computer interfaces. *International Journal of Man-machine Studies*, 1989. 30(4): pp. 355-375.
11. McGee, D., Cohen, P.R., and Oviatt, S.L., Confirmation in Multimodal Systems, in *Proc. of the 17th International Conference on Computational Linguistics (COLING '98) and 36th Annual Meeting of the Association for Computational Linguistics (ACL'98)*. 1998: Montreal, Canada. pp. 823-829.
12. Moore, R., Dowding, J., Bratt, H., Gawron, J., Gorfu, Y., Cheyer, A., CommandTalk: A Spoken-Language Interface for Battlefield Simulations, *Proc. of the 5th Conference on Applied Natural Language Processing*, Association for Computational Linguistics, 1997: Washington, DC. pp. 1-7.
13. Oviatt, S. L., Pen/Voice: Complementary multimodal communication, Proc. of Speech Tech'92, New York, 238-241
14. Oviatt, S.L., Multimodal interactive maps: Designing for human performance. *Human Computer Interaction*, 1997. 12: pp. 93-129.
15. Oviatt, S.L., Mutual disambiguation of recognition errors in a multimodal architecture, in the *Proc. of the Conference on Human Factors in Computing System*, New York, 1999, ACM Press, pp. 576-583.

16. Oviatt, S.L., Cohen, P. R., Wu, L., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J., 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 submission.

17. Pausch, R. and Leatherby, J. H., A study comparing mouse-only vs. mouse-plus-voice input for a graphical editor, *Journal of the American Voice Input/Output Society, 9:2*, July, 1991, pp 55-66

18. Rudnicky, A.I., Mode Preference in a simple data-retrieval task, in *ARPA Human Language Technology Workshop*. March 1993: Princeton, New Jersey.

19. Suhm, B., Myers, B., and Waibel, A., Model-based and empirical evaluation of multimodal interactive error correction, in the *Proc. of the Conf. on Human Factors in Computing Systems*, New York, 1999, ACM Press, 584-591.

20. Shneiderman, B., Natural vs. precise concise languages for human operation of computers: Research issues and experimental approaches. Proceedings of the 18th Annual Meeting of the Association for Computational Linguistics, and Parasession on Topics in Interactive Discourse, Univ. of Pennsylvania, June, 1980, pp. 139-141.

21. Wickens, C., Sandry, D., and Vidulich, M., Compatibility and resource competition between modalities of input, central processing, and output. *Human Factors*, 1983. 25(2): pp. 227-248.

22. Wu, L., Oviatt, S., L. and Cohen, P. R., Statistical multimodal integration for intelligent HCI, in *Neural Networks for Signal Processing*, Y.H. Hu, Larsen, J., Wilson, E., and Douglas, S., Editors. 1999, IEEE Press: New York. pp. 487-496.