The Adjustable Grid: A Grid-Based Cursor Control Solution using Speech Recognition

Tarif Haque[1], Emily Liang[2], Jeff Gray[1]
[1]Department of Computer Science
[2]Departments of Biology and Anthropology
The University of Alabama
{thaque, ejliang, jggray}@crimson.ua.edu

ABSTRACT

Individuals with motor disabilities often find hands-free, speech-based systems useful because they provide an alternative to traditional mouse-centered navigation. A small number of grid-based cursor control systems using speech recognition have been developed. These systems typically overlay a numbered 3x3 grid on the screen and allow the user to recursively drill the cursor down to a target location by speaking a grid number. Though a 3x3 grid remains the standard, it still remains elusive as to which granularity maximizes performance in specific desktop environments, particularly in regard to time delays and error rates of click tasks. The objective of this research is to develop a grid of adjustable granularity both to compare the efficacy of a variety of grid sizes and to provide users with an alternative to current systems which only offer a single default grid.

Categories and Subject Descriptors

H5.0 **[Information Interfaces and Presentation]:** General; K4.2 **[Computers and Society]**: Social issues—*Assistive technologies for persons with disabilities*

General Terms

Human Factors, Performance, Experimentation

Keywords

Cursor control, navigation, speech recognition, grid, human-computer interaction, user interfaces, accessibility

1. INTRODUCTION

The mouse is one of the most widespread and successful input devices for interacting with a personal computer and easily lends to spatial navigation tasks, which are crucial in desktop graphical user interfaces (GUIs) with WIMP (windows, icons, menus, pointing) style interfaces. The unique requirements for motor impaired users can be overlooked when traditional human-computer interfaces are designed. Unfortunately, this broad group of users often has limited or no ability to use a mouse. Motor impairments hindering the use of a mouse may be caused by a – variety of conditions, including Cerebral Palsy, Parkinson's disease, spinal injuries, partial paralysis, arthritis, and strokes.

The remainder of this introduction surveys and compares alternative forms of input interfaces for those with disabilities. The rest of the paper then addresses a specific research question that we investigated regarding the granularity of control for a grid-based cursor control interface.

1.1 Alternative Cursor Control Systems

Because mouse control is a necessity in many modern computer interactions, motor-impaired users must find alternative modes of cursor control. Numerous studies have investigated cursor control techniques and eye trackers, head trackers, tongue operated joysticks, and voice-controlled systems have been explored as possible solutions [14]. More recently, neural prosthetic controls have been studied [4].

Eye gaze tracking (EGT) systems intuitively utilize the user's direction of gaze to move the cursor, but often require expensive peripheral input devices [2]. Electromyogram (EMG) based cursor control systems require the user to attach physical EMG electrodes to the face, which monitor muscle movements that are associated with a particular cursor direction [1]. Recent algorithm improvements to neural prosthetic cursor control allow the user to control a cursor with their thoughts and approach the performance of a real arm, but such systems, when available, will require the implantation of brain sensors [4]. Our interest in speech based systems arises in part because they require no external equipment to facilitate their use, aside from a standard microphone.

1.2 Speech-Based Cursor Control

A comprehensive speech-based control system for personal computers requires both a text-entry mechanism and a cursor control mechanism, which provide an alternative to the standard keyboard and mouse, respectively [10]. Robust dictation (speech-to-text) software comes bundled with many commercial software applications like *Dragon NaturallySpeaking* and *Windows Speech Recognition*, both which use automatic speech recognition (ASR) to generate commands for interacting in a desktop environment. Past research shows the productivity of ASR systems does not differ when used by traditional users as opposed to users with a loss of motor function caused by spinal injuries [12].

Although dictation tools have improved significantly in recent years, mapping speech to a viable cursor control solution has proven more difficult. Speech-based cursor control solutions have diverged to two approaches: direction-based (e.g., "move up three lines") and target-based (e.g., "select *target*").

ACMSE'13, April 4–6, 2013, Savannah, GA, USA.
Copyright 2013 ACM 978-1-4503-1901-0/13/04...$15.00

1.2.1 Direction-Based Solutions

Direction-based solutions can either be discrete or continuous. A discrete methodology asks the user to specify the direction and the number of units to move in that direction. For example, the user could say: “Move Left 3 Units.” Typically, the units are specified for the appropriate context. For example, within the commercial software *Dragon NaturallySpeaking*, the user can say “Move back two paragraphs” in a word-processing application.

A continuous methodology, on the other hand, asks the user to specify a direction for cursor movement until a “Stop” command is reached. There are, however, delays associated with this approach. First, the user must perceive the cursor has reached the target. The user must then issue a stop command, which in turn the computer must process. Because there are several opportunities for delays to arise, these systems tend to be slow and error-prone [11]. Because speech-based commands tend to present a set of inherent delays, predictive cursors that tell the user how far in advance to issue a “Stop” or “Click” command were designed, but showed no significant improvements to task completion times, error rates or user satisfaction compared to the standard cursor [7].

Mihara, Shibayama and Takahashi proposed the Migratory Cursor, which uses both a discrete and continuous specification using ghost cursors [9]. Depending on the direction of movement, a row or column of numbered ghost cursors accompanies the actual cursor. First, the user hones in on the ghost cursor nearest the target by using a discrete command (e.g., “move right, three”). The user then uses a continuous vocalization (e.g., “ahhhh” to move left continuously) to fine tune the position.

Harada et al. proposed the Vocal Joystick, a particularly promising direction-based solution that allows the user to continuously move the cursor by varying vocal parameters such as vowel quality, pitch, and loudness [5]. In their system, vowel sounds are mapped continuously to a 2-D cursor space and the cursor stops when the user stops vocalizing. Theoretically, optimal performance of the Vocal Joystick could be equivalent to that of a hand-operated joystick. This method is quite effective for drawing applications, when fluid cursor movement is valued, but in all direction-based systems, the performance of target selection tasks is dependent on the previous location of the cursor. In other words, the further away the cursor is from the target, selection becomes more difficult and time-consuming [8].

1.2.2 Target-Based Solutions

Target-based systems immediately move the cursor to a specific point on the screen, and unlike direction-based solutions, their performance does not depend on the previous location of the cursor. For example, in a text document, the user could say “Select Friday” which would highlight the word Friday. In another target-based approach using *Windows Speech Recognition*, the user can say “Start” (which would open the start menu) and then a user could launch a program by saying the name of the program on the start menu.

Target-based solutions are useful when there is contextual information associated with each target, particularly word processing. However, moving the cursor to any point on the screen can be difficult with such an approach, when not all items on the screen are associated with a label or when multiple targets have the same name (links in a web search, for example). As interfaces become increasingly complex, mapping every item on the screen to a contextual label will be impractical. For these reasons, we shifted our focus to a solution that allowed a user to move the cursor to a specific point on the screen without any contextual labels.

2. GRID-BASED CURSOR CONTROL

A grid-based system implements a recursive, target-based strategy that splits the screen into multiple labeled regions. Existing solutions traditionally overlay a numbered 3x3 grid on the screen, allowing the user to “drill” down to a target location by splitting each successive target into a 3x3 grid. When the user is sufficiently close to the target, the user issues a click command, which moves the mouse to the center of the selected grid, as demonstrated in Figure 1.

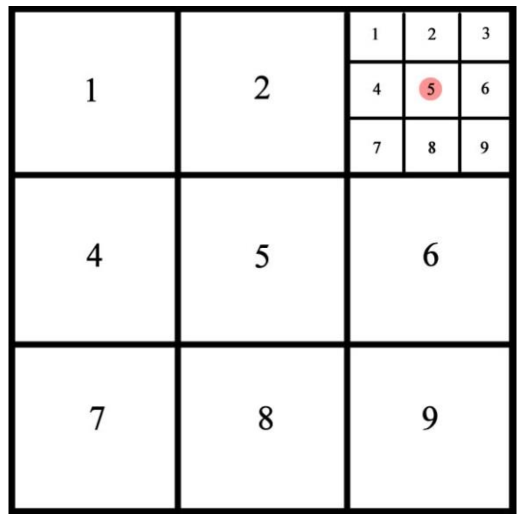

Figure 1: A standard 3x3 grid. To navigate to the red dot, the user says “Three” then “Five” then “Click.”

Compared to other modes of cursor control, grid-based solutions have been understudied. Kamel and Landay first employed a grid-based solution using a standardized 3x3 matrix in drawing applications for blind users [6]. Thereafter, two studies explored the usability of the standard 3x3 grid for desktop cursor control and suggested possible improvements, as summarized in the remainder of this section.

Dai et al. studied the advantages of a nine cursor solution over a single cursor solution [3]. Their results confirmed that multiple cursors, one within each grid block, are more effective at selecting targets than a single cursor placed in the middle of the screen. Furthermore, they concluded that the grid-based system dramatically reduced error rates and eliminated the effect of distance compared to earlier direction based systems. They went on to suggest that grid-based solutions appear to be the most promising generic speech-based cursor control mechanism and advised further research be done to improve the selection of smaller targets. Zhu, Feng and Sears investigated the benefits of fine-tuning and magnification within a 3x3 grid-based system, concluding both fine tuning and magnification improve the user’s ability to select targets [15].

Building on this previous work, we have investigated the benefits of an adjustable grid-based solution that does not universally adopt the 3x3 grid, but allows flexibility in the granularity.

3. THE ADJUSTABLE GRID

The Adjustable Grid (AG) was designed with several goals in mind. First, we sought to design a system that gave the user more flexibility by allowing them to specify the resolution of the grid. In addition, such a system would facilitate further comparisons between grids of various resolutions, which we explored during the experimentation phase of this project.

We hypothesized that the 3x3 grid does not necessarily optimize performance, particularly when compared to grids of higher resolutions. As the granularity of the grid increases, our intuition suggested that the selection time would decrease because the user can specify a target using fewer commands.

However, increasing the resolution of the grid does present a tradeoff. Though targets can be specified with fewer commands, the increased speech vocabulary that accompanies a higher-resolution grid may lead to more recognition errors. For example, in a 4x4 grid, "six" and "sixteen" may sound too much alike and consequently, the cursor may move to the wrong target grid. One of our key goals was to investigate how frequently these recognition errors arose in grids of differing sizes and what measures could be taken to alleviate them.

With these general objectives in mind, we designed and implemented a baseline prototype of the Adjustable Grid.

3.1 Functionality of the Adjustable Grid

The Adjustable Grid first asks the user to specify the width dimension and then the height dimension of the grid using speech commands. To illustrate, if the user says "4" then "5" during these prompts, a 4x5 grid is painted on the screen, as shown in Figure 2.

"one"	"two"	"three"	"four"
"five"	"six"	"seven"	"eight"
"nine"	"ten"	"eleven"	"twelve"
"thirteen"	"fourteen"	"fifteen"	"sixteen"
"seven-teen"	"eighteen"	"nineteen"	"twenty"

Figure 2: Dynamic grammar generation using a 4x5 grid. For illustrative purposes, each vocabulary is shown with its respective grid.

After the grid size has been determined, the speech vocabulary is dynamically generated. Generating the vocabulary after the grid has been specified maximizes performance for each individual grid by limiting the number of speech commands that can be spoken and thereby reducing speech recognition errors. The 4x5 grid shown in Figure 2 generates a grammar from "One" to "Twenty," inclusive.

The selection strategy of the Adjustable Grid works in a similar manner to previous grid-based solutions. The user hones in on a target by speaking a grid number, which is then split into another 4x5 grid in the given example. With each selection, the mouse moves to the center of the current grid. The user issues a "click" command when sufficiently close to the target.

A dynamically generated grid of this nature can be used with a wide range of screen sizes. Smaller screens may benefit from a 2x2 or 3x3 grid, whereas larger screens and more complex interfaces may benefit from a 4x4 or 5x5 grid. Users interacting with widescreen displays may opt for a grid with unequal dimensions (e.g., 4x3). Such flexibility is possibly the greatest advantage of the Adjustable Grid solution.

3.2 Software Implementation

The Adjustable Grid was implemented in Java using the Eclipse IDE on a machine running *Windows 7*. Carnegie Mellon's open-source Sphinx SR Toolkit handled our speech recognition functional needs. The Java *Robot* class allowed the application to programmatically control cursor movement.

4. EXPERIMENTATION

This section outlines the experimental design and procedure for testing the Adjustable Grid. Our experiments were performed using two *Dell Latitude E6520* laptops running *Windows 7* with screen resolutions of 1920x1080.

4.1 Experimental Design

To our knowledge, no previous research compares grids of differing granularities for speech-based cursor control. Two comparison-based studies were designed and performed by the research team. The first evaluated the performance of pure-resolution grids against one another. The second sought to investigate the potential of mixed resolution grids by comparing a 4x3 grid to the standard 3x3 grid.

The trend that motivated our experiments was formulated by *Dai et al.* using a theoretical equation that specified the number of commands needed to select any point on the screen with a grid-based solution:

$$\boldsymbol{N = \log_n(\frac{D}{A})}$$

In this equation, n is the number of columns/rows (in a 3x3 grid, n = 3), D is the largest parameter of the screen resolution (in the experimental case, D = 1920), and A is the size of the target in pixels [3]. From this equation, we see that increasing the granularity of the grid reduces the number of commands needed to select a target. By varying the granularity in our experiments, we were able to determine whether the reduction in commands issued led to better performance using existing speech recognition tools.

4.1.1 Experiment A: Pure-resolution Comparison

The first experiment aimed to determine the differences in performance among pure-resolution grids. A pure-resolution grid is one that retains a distinct granularity throughout the selection process (2x2, 3x3, 4x4, etc.). In contrast, mixed-resolution grids have unequal numbers of columns and rows (5x3, 4x2, etc.). The first experiment tested pure grids with dimensions from 2x2 to 6x6. Performance for each grid involved three parameters:

selection time, the number of commands to select the target, and the frequency of recognition errors.

4.1.2 Experiment B: Mixed-resolution Comparison

The second experiment sought to determine if a sample mixed-resolution grid (e.g., a 4x3 grid) would be more effective for users with widescreen displays than a pure-resolution grid, such as a 3x3 grid, using the same testing protocol outlined in Part A.

4.1.3 Hypotheses of AG Experiments

For Experiment A, we predicted a reduction in selection time as grid granularity increases. In other words, the fewer speech commands of a higher-resolution grid would sufficiently outweigh the increase in recognition errors, as far as performance is concerned.

For Experiment B, we predicted that with widescreen displays, target selection would be improved with a mixed-resolution grid (4x3), as the 4x3 grid divisions are closer to a square in shape compared to a pure-resolution grid (3x3).

4.2 Experimental Procedure

Two members of the research team tested each of the grids as follows: Five targets, 94 x 110 pixels in size, were placed on the screen, as shown in Figure 3. First, the user specified the grid size using speech commands. The selected grid was overlaid on the screen and the user attempted to navigate to the first icon. After clicking the icon, a full screen program launched. The user then navigated to the top-right corner of the screen and issued a "click" command to close the program. After closing the first program, the user attempted to navigate to the second target, repeating the procedure for icons 2, 3, 4, and 5. For each target, the total time to specify the grid size, open, and close the program was recorded. This five-target procedure is then repeated for all the grid-sizes being tested.

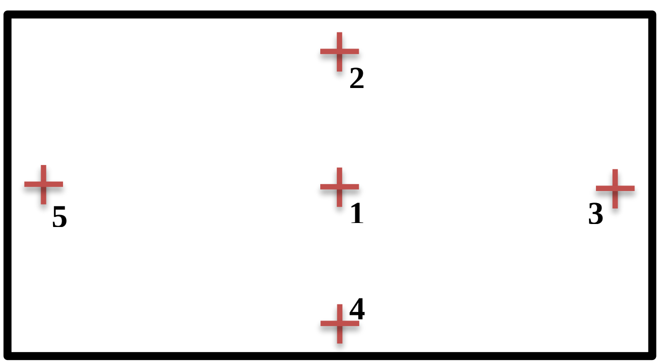

Figure 3. Layout of the experimental screen showing the five locations of the target icons.

4.2.1 Part A: Pure-Resolution Grid Comparisons

The 2x2, 3x3, 4x4, 5x5 and 6x6 grids generated by the adjustable grid prototype were tested on two identical machines with the procedure outlined above. Each machine was given one trial of five targets. Performance times, the number of commands, and recognition errors for selecting and closing each target program were recorded.

4.2.2 Part B: Pure vs. Mixed-Resolution Grids

To evaluate the potential of mixed resolution grids, a 5x3 mixed-resolution grid was tested against a 3x3 pure-resolution grid. Both grids were given one trial of five targets and tested on two identical machines with widescreen displays. Performance times, recognition errors and the number of commands to reach the target were recorded according to the protocol outlined in Part A.

5. RESULTS

This section presents several data tables that summarize our results. Each data table highlights the mean of three dependent variables measured by the experiments: completion time, the number of commands issued, and number of recognition errors.

For each grid, the mean completion time averages the total procedure time for each of the five targets, which includes specifying the grid size, selecting the target program and closing the program. The mean number of speech commands and the mean number of recognition errors during the course of the procedure are given. The standard deviation is given in parentheses beneath each measurement.

5.1 Pure-Resolution Grid Comparisons

Tables 1 and 2 compare the performance of five pure-resolution grids within the Adjustable Grid on two separate machines, used by two separate users.

Table 1. Summarized Pure Grid Performance on Machine 1.

	2x2	**3x3**	**4x4**	**5x5**	**6x6**
Time	26.69 (2.12)	25.76 (9.85)	28.65 (5.81)	19.96 (6.14)	18.87 (2.74)
Commands	13.0 (0.00)	10.40 (2.07)	12.60 (2.19)	10.0 (2.92)	8.60 (1.34)
Errors	0.20 (0.45)	2.80 (2.95)	1.40 (0.89)	1.00 (1.73)	0.20 (0.45)

Table 2. Summarized Pure Grid Performance on Machine 2.

	2x2	**3x3**	**4x4**	**5x5**	**6x6**
Time	30.00 (1.06)	19.04 (1.23)	28.56 (8.42)	20.79 (2.04)	22.63 (4.07)
Commands	13.40 (0.89)	8.80 (0.45)	10.60 (2.50)	8.80 (0.45)	8.00 (0.71)
Errors	0.20 (0.45)	0.20 (0.45)	1.00 (1.41)	0.00 (0.00)	0.20 (0.45)

Our results suggest an improvement for 5x5 and 6x6 pure-resolution grids. The 3x3 grid performed well on Machine 2, but not as well on Machine 1. The 5x5 and 6x6 grids, however, consistently performed well across both machines. Even with their increased vocabulary size, the 5x5 and 6x6 grids appear to work as well, if not better, than the 3x3 grid.

Across the board, the relatively high standard deviations for selection time suggest existing speech recognition tools continue to be unreliable. The 2x2 grid, with a small vocabulary of four words, appeared to be the only grid that consistently performed at error-free levels.

5.2 Mixed-Resolution Grid Comparisons

Table 3 compares the performance of a 4x3 mixed resolution grid to a 3x3 pure resolution grid, used on two separate machines.

Table 3: Mixed-Resolution Grid vs. Pure-Resolution Grid

	4x3 [PC I]	**4x3 [PC II]**	**3x3 [PC I]**	**3x3 [PC II]**
Time	16.44 (1.29)	19.53 (1.2)	25.76 (9.85)	19.04 (1.23)
Commands	9.4 (0.55)	8.4 (0.89)	10.40 (2.07)	8.80 (0.45)
Errors	0.00 (0.00)	0.40 (0.55)	2.80 (2.95)	0.20 (0.45)

Our findings show clear potential for mixed-resolution grids. The 4x3 grid performed better, if not as well, as the 3x3 grid on multiple machines. On Machine I, the 4x3 grid performed significantly better than the 3x3 grid, with a 36% improvement in completion time at error-free levels. In addition, the low standard deviations of the 4x3 grid measurements compared to the 3x3 grid shed light on the grid's reliability.

6. EXTENDING THE ADJUSTABLE GRID

During preliminary testing, we found that several issues arose with grid-systems of pure granularity. For example, a standard 3x3 grid persistently divides each successive grid selection into another 3x3 grid. For higher resolution grids, such a methodology can produce indiscriminate grids after just a few levels of recursion. For a 5x5 grid, imagine splitting the screen into 25 rectangular units, and then splitting one of those units into another set of 25 rectangular units. Very quickly, it becomes difficult to distinguish visually between individual units, much less label them with a useful coding.

As a suggestion for further implementation, we propose a solution that allows for the specification of two grids: the first grid would be of a higher resolution and provide the initial overlay, whereas the second grid would be of a smaller resolution allowing for fine-tuning with fewer recognition errors and subsequent recursive drilling at each new step of the navigation.

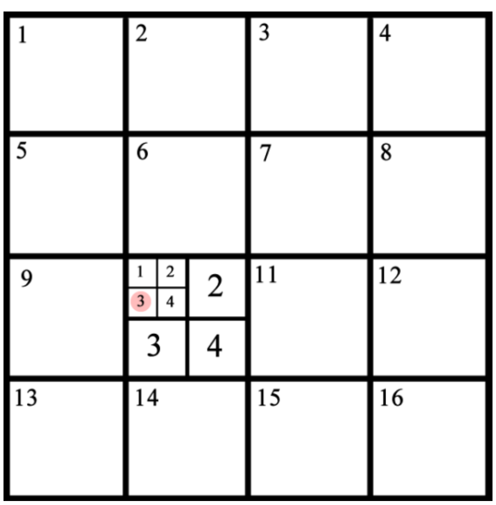

Figure 3: A solution using two different granularities. A 4x4 grid is used for the initial overlay and a 2x2 grid is used for subsequent drilling.

7. CONCLUSION

Grid-based systems are an understudied interface for cursor control. In this paper, we presented a grid of adjustable granularity to facilitate further research among grids of differing resolutions. A cursor control system that allows users to specify the granularity of a grid and then navigate their personal computer using voice commands may prove useful to those with motor impairments, who have limited ability to use a mouse.

Experimentation carried out by the research team shows that speech-recognition errors continue to plague grid-based cursor control solutions. Singular utterances out of context are still difficult for current SR tools to recognize. Even so, grids of higher resolutions did not perform as poorly as predicted, withstanding the errors posed by an increased vocabulary size to a certain degree. A comparison between a singular mixed-resolution and pure resolution grid demonstrated a notable performance gain by the mixed grid over its pure grid counterpart.

In summary, we suggest that future grid-based cursor control systems focus less on a singular 3x3 grid, and take a dynamic, adjustable approach that offers differing granularities, an implementation that grants the user increased flexibility. Multiple grids have shown to match and surpass the performance of the 3x3 grid. As user interfaces become more expansive and demanding, relying on a single grid will be limiting to users.

Speech-based cursor control is a very viable solution that addresses the accessibility concerns of motor-impaired users, and we believe an adjustable, grid-based approach can allow these users to access information more expediently.

8. ACKNOWLEDGEMENTS

This research is sponsored in part by NSF award #117940.

9. REFERENCES

[1] Chin, C. A., Barreto, A., Cremades, J. G., and Adjouadi, M. 2008. Integrated electromyogram and eye-gaze tracking cursor control system for computer users with motor disabilities. *Journal of Rehabilitation Research & Development*, *45* (1), 161-174.

[2] Corno F., Farinetti, L., and Signorile, I. 2002. An eye-gaze input device for people with severe motor disabilities. *Proceedings of SSGRR-2002* (L'Aquila, Italy).

[3] Dai, L., Goldman, R., Sears, A., and Lozier, J. 2005. Speech-based cursor control using grids: modelling performance and comparisons with other solutions. *Behaviour and Information Technology, 24* (3), 219-230.

[4] Gilja, V., Nuyujukian, P., Chestek, C. A., Cunningham, J. P., Yu, B. M., Fan, J. M., Ryu, S. I., and Shenoy, K. V. 2012. A brain machine interface control algorithm designed from a feedback control perspective. *Proceedings of the 34th Annual IEEE EMBS Conference*, (San Diego, CA), 1318-1322.

[5] Harada, S., Landay, J. A., Malkin, J., Li, X., and Bilmes, J. A. 2006. The Vocal Joystick: Evaluation of Voice-based Cursor Control Techniques. *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, (Portland, OR), 197-204.

[6] Kamel, H. M. and Landay, J. A. 1999. The Integrated Communication 2 Draw (IC2D): A Drawing Program for the Visually Impaired, *Proceedings of the ACM SIGCHI*

Conference on Human Factors in Computing Systems (Pittsburgh, PA), 222-223.

[7] Karimullah, A. S. and Sears, A. 2002. Speech-Based Cursor Control. *Proceedings of the 5th International ACM Conference on Assistive Technologies*, (Edinburgh, Scotland, UK), 178-185.

[8] Karimullah, A., Sears, A., Lin, M., and Goldman, R. 2003. Speech-based cursor control: Understanding the effects of variable cursor speed on target selection. *Proceedings of the HCII*, 681–685.

[9] Mihara, Y., Shibayama, E., and Takahashi, S. 2005. The migratory cursor: accurate speech-based cursor movement by moving multiple ghost cursors using non-verbal vocalizations. *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility*, (Baltimore, MD), 76-83.

[10] Oviatt, S. 1997. Multimodal interactive maps: designing for human performance. *Human-Computer Interaction, 12* (1), 93-129.

[11] Sears, A., Feng, J., Oseitutu, K., and Karat, C. M. 2003. Hands-Free, Speech-Based Navigation During Dictation: Difficulties, Consequences, and Solutions. *Human-Computer Interactions, 18* (3), 229-257.

[12] Sears, A., Karat, C. M., Oseitutu, K., Karimullah, A. S., and Feng, J. 2001. Productivity, satisfaction, and interaction strategies of individuals with spinal cord injuries and traditional users interacting with speech recognition software. *Universal Access in the Information Society, 1* (1), 4-15.

[13] Sears, A., Lin, M., and Karimullah, A. S. 2003. Speech-based cursor control: understanding the effects of target size, cursor speed, and command selection. *Human-Computer Interaction; Theory and Practice. Part II,* Stephanidis, C. and Jacko, J. eds. L. Erlbaum Associates, Inc., Mahwah, NJ, 681-685.

[14] Sears, A., Young, M., and Feng, J. 2003. Physical Disabilities and Computing Technologies: An Analysis of Impairments. *The Human-Computer Interaction Handbook*, Jacko, J. and Sears, A. eds. L. Erlbaum Associates Inc., Hillsdale, NJ, 482-503.

[15] Zhu, S., Feng, J., and Sears, A. 2010. Investigating Grid-Based Navigation: The Impact of Physical Disability. *ACM Transactions on Accessible Computing, 3* (1), 1-30.