Pradipta Biswas
Research Student, Computer Laboratory, University of Cambridge

Peter Robinson
Professor of Computer Technology and Deputy Head of the Department
Computer Laboratory, University of Cambridge

A new screen scanning system based on clustering screen objects

Abstract
Many physically challenged users cannot interact with a computer through a conventional keyboard and mouse. They may interact with a computer through one or two switches with the help of a scanning mechanism. In this paper we present a new scanning technique based on clustering screen objects and then compare it with two other scanning systems by using a simulator. The analysis shows that the best scanning system is a type of block scanning that divides the screen in four equal sized partitions for four iterations and then switches to eight-directional scanning. However with a more accurate target acquisition process, the cluster scanning technique is found to outperform other scanning systems.

Key words
Human computer interaction; assistive technology; single switch scanning; simulation.

Many physically challenged users cannot interact with a computer through a conventional keyboard and mouse. For example, spasticity, amyotrophic lateral sclerosis (ALS) and cerebral palsy confine movement to a very small part of the body. These people may interact with a computer through one or two switches with the help of a scanning mechanism. Scanning is the technique of successively highlighting items on a computer screen and pressing a switch when the desired item is highlighted.

Most work on scanning has aimed to enhance the text entry rate of a virtual keyboard. In these systems the mechanism is usually block-row-column-item based scanning (Simpson & Koester, 1999; Lesher *et al*, 2002). However, navigation to arbitrary locations on a screen has also become important as graphical user interfaces are more widely used. Two types of scanning mechanism are commonly used for general navigation. Cartesian scanning moves the cursor progressively in a direction parallel to the edges of the screen, and polar scanning selects a direction and then moves along a fixed bearing. A particular type of polar scanning that allows movement only in eight directions is commonly used (Steriadis & Constantnou, 2002; Ntoa *et al*, 2004; and in a wheelchair mobility interface, O'Neill *et al*, 2002). In both Cartesian and polar scanning systems, the interaction rate of users remains very low. Consequently recent scanning systems have tried to combine two or more types of scanning to get better performance.

Examples of some existing systems in the same discipline are the Autonomia System (Steriadis & Constantnou, 2002), the FastScanner system (Ntoa *et al*, 2004), the Gus! Scanning Cursor (Gus Scanning Cursor, 2007), the ScanBuddy system (The ScanBuddy system, 2007) and the SSMCI (single switch mouse control interface) system (Moynahan & Mahoney, 1996). The Autonomia system (Steriadis & Constantnou, 2002) replaces the windows and widgets of a typical Microsoft Windows interface by frames and wifsid (widget for single-switch input devices) respectively. The system consists of different frames such as cursor frame, virtual keyboard frame, console frame etc. The cursor frame provides eight-directional scanning whereas the frame itself and other frames are scanned using the block-row-item based scanning approach. The FastScanner system starts the scanning process by showing a list of currently open applications and asks the user to choose an application. The scanning procedure then restarts itself to the selected application. The objects of an interface are scanned sequentially based on a predefined order. Screen navigation is done by eight-directional scanning. Additionally, the objects of an interface are divided into four classes (text entry objects; simple objects; selection objects; container objects) and the user input is interpreted according to the type of the object that has received the input. The Gus Scanning Cursor provides different types of navigation strategies (like Cartesian, polar, eight-directional) at a single screen and the screen itself is scanned by row-item based scanning. The user has to choose a particular scanning type to navigate through the screen. The ScanBuddy system scans the screen by iteratively dividing it into two equal parts up to four times. Finally it scans the smallest part using Cartesian scanning. In the SSMCI system, an intelligent agent operates to guess the target and moves the cursor accordingly. If the guess is incorrect the user has to signal the agent, which then re-evaluates the situation and comes up with a new solution. There also exists some scanning application for some specialised tasks like text selection (Shein, 1997) and menu selection process (Evreinov & Raisamo, 2004).

Most of these scanning systems (except Gus and SSMCI) have a similar structure. They start by dividing the screen into several blocks and then introduce either Cartesian or polar scanning within a block. As a result, users can traverse shorter distances using Cartesian or polar scanning and the time needed to reach a target from long distances is reduced. However, an arbitrary screen layout cannot always be evenly divided into blocks, rows or columns, so different scanning systems define blocks differently. The Autonomia system introduces blocks by providing different frames. The FastScanner system defines blocks based on the hierarchy of objects in the Microsoft Windows operating system. The ScanBuddy system defines blocks just by dividing the screen in two equal segments.

There is little published work comparing performances of different scanning techniques. Angelo (1991) compared automatic, inverse and step scanning and concluded (unsurprisingly) that the choice of scanning depended on the type of disability of the user. Birch (2000) compared Cartesian and polar scanning approaches and found that: '*the Cartesian method was shown to be faster over the entire screen when compared to the rotational method*'. Blackstien-Adler *et al* (2004) compared continuous Cartesian, discrete Cartesian, rotational, and hybrid quadrant/continuous Cartesian scanning techniques and found that Cartesian scanning (both continuous and discrete) was not only preferred, but also the most effective scanning strategy. However, these works were confined to polar and Cartesian scanning. Additionally, they compared the performance of the scanning systems on some synthesized situations. In this paper we propose a new scanning technique based on clustering the screen objects and then compare its performance with two other scanning approaches. We did not restrict our participants to any particular task and evaluated the scanning systems for natural set of interactions. So our results can be considered more general than the previous attempts.

In the following section we describe the three different scanning systems that are considered in the present study. In Section 3, we discuss our cluster scanning system in detail. The performance of the scanning systems is compared in Section 3. Finally we make concluding remarks in at Section 4.

The scanning systems

In the present study we have considered the following three types of scanning systems.

Eight-directional scanning system

In this scanning technique the pointer icon is changed at a particular time interval to show one of eight directions (up; up-left; left; left-down; down; down-right; right; right-up). The user can choose a direction by pressing the switch when the pointer icon shows the required direction. When the pointer reaches the desired point in the screen, the user has to give another key press to stop the pointer movement and make a click. ***Figure 1*** shows a probable cursor trace using the eight-directional scanning system.

Block scanning system

In the block scanning system the screen area is iteratively segmented into equally sized sub-areas (***Figure 2***). The user has to select a sub-area that contains the intended target (the green rectangle in ***Figure 2***). The segmentation process runs a certain number of iterations and after that eight-directional scanning is initiated in the selected sub-area.

Cluster scanning system

The cluster scanning system collects all possible targets (eg. icons, buttons, combo-boxes etc) by enumerating window processes (currently it operates only for Microsoft Windows operating systems). Then it iteratively divides a screen into several clusters of targets based on their locations (***Figure 3***) we used fuzzy c-means algorithm (Ross, 1997) to cluster the targets. The user has to select the appropriate cluster

Figure 1: The eight-directional scanning system

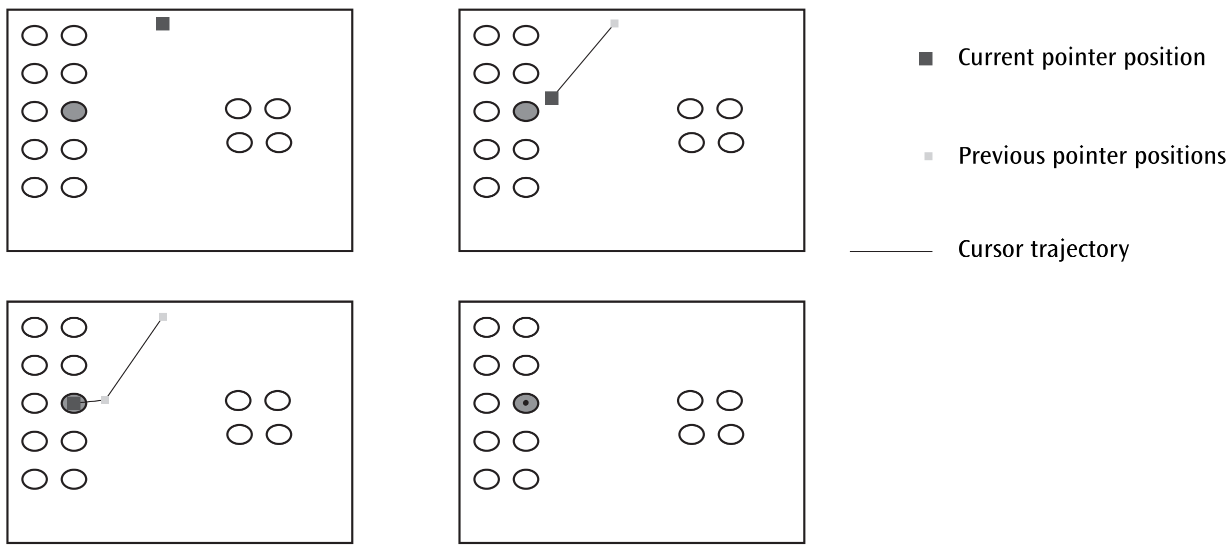
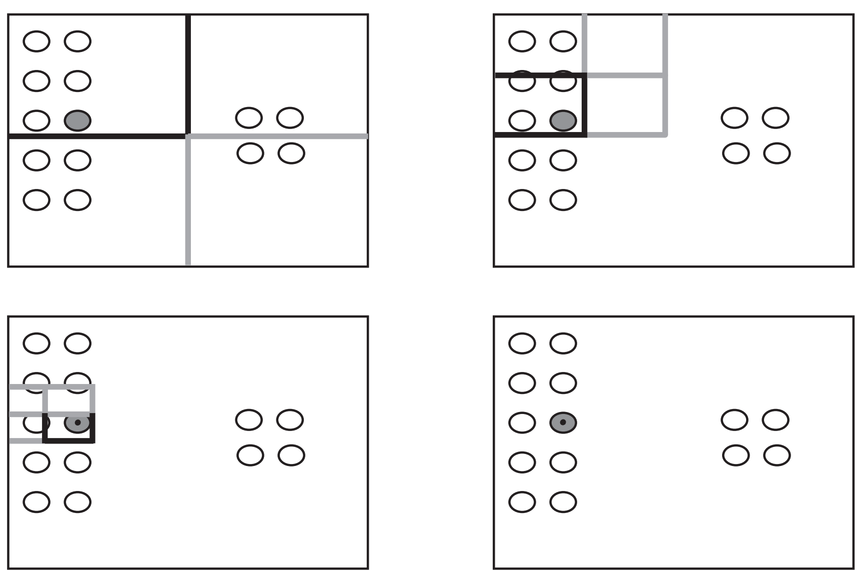


Figure 2: The block scanning system



that contains the intended target. After reaching a relatively small cluster, the system switches to eight-directional scanning. The user can select the target or can navigate through the screen using eight-directional scanning mechanism.

This particular system does not introduce any new interface element (like a frame or form) in the screen as Autonomia or FastScanner system do. So we can expect users to take less time to learn this system than existing ones. Additionally, the system does not blindly divide the screen in a predefined number of segments (as the ScanBuddy system does). It clusters the target so that the targets are evenly divided into blocks and a block is not drawn in a region that does not contain any target. As a result it can minimize the target selection time.

A demonstration of these scanning systems can be downloaded from:

- EightD Scanning: www.youtube.com/watch?v=0eSyyXeBoXQ&feature=user

Figure 3: The cluster scanning system

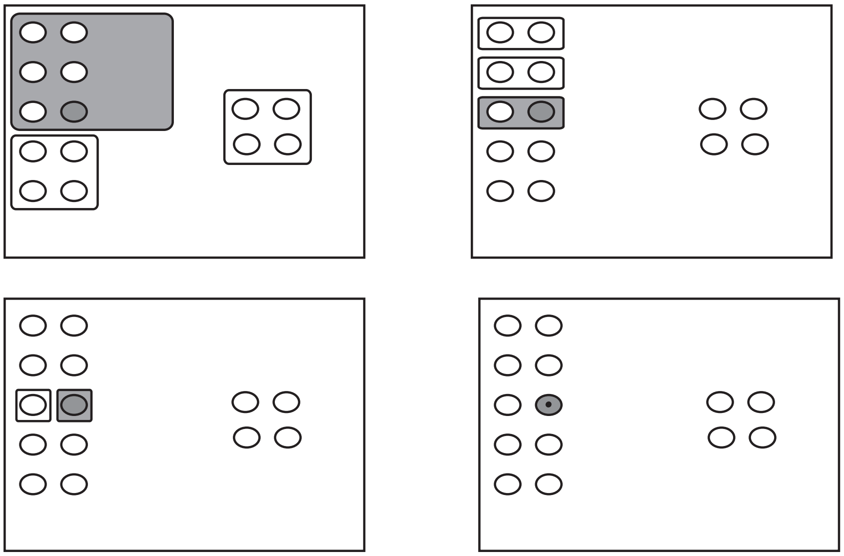
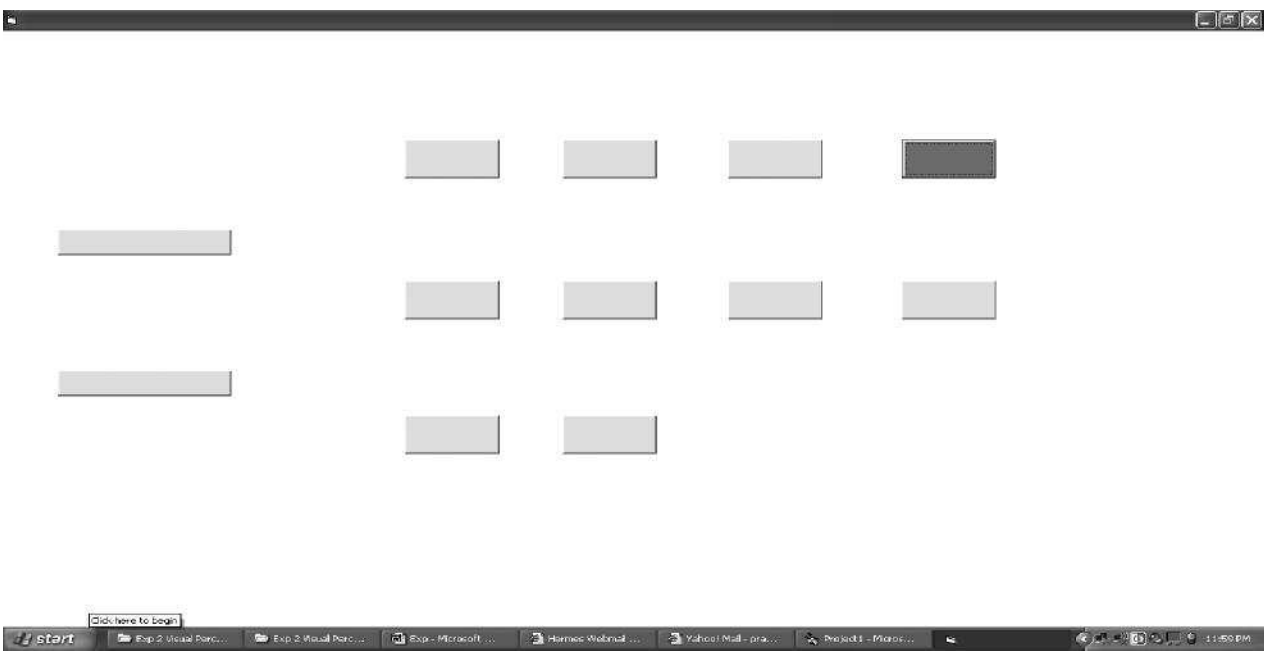


Figure 3: Screenshot of the demonstration for scanning interfaces



- Block Scanning: www.youtube.com/watch?v=UTmMstrGDZY
- Cluster Scanning: www.youtube.com/watch?v=tRtbsn2LfeA&feature=user.

The demonstrations involved a task of pressing a set of buttons placed in a screen (*Figure 4*) in a particular sequence. All of the buttons were coloured grey except the next target, which was red. The same task was repeated for all the scanning systems.

Performance comparison of the scanning systems

We have developed a simulator to investigate the interaction patterns of different scanning systems. The simulator takes a mouse cursor trace for undertaking a task as input and predicts the equivalent cursor trace and completion time for the same task using a scanning system. The details about the architecture of the simulator can be found in our earlier publications (Biswas & Robinson, 2008; Biswas & Robinson, 2007). Our initial evaluation calibrated and tested the simulator for an eight-directional scanning system (Biswas & Robinson, 2007).

In the present study, sample interactions by two able-bodied users were recorded to generate a list of tasks, which were fed to the simulator to evaluate different scanning techniques. The users were expert computer users and they were not instructed to use any particular application or to do any specific task. These can therefore be taken as representative of natural interactions. The simulator estimated the time needed to undertake the same set of tasks using different scanning systems.

Results

We investigated the naïve eight-directional scanning, block scanning for different numbers of blocks and different numbers of iterations, and cluster scanning for different numbers of clusters. The estimated task completion times are shown in *Table 1* and *Figure 5*. The fact that some of these tasks would take over two hours to complete indicates the value of simulation over user trials.

Discussion

The results clearly show that both the cluster scanning and block scanning processes perform better than eight-directional scanning and thus support the use of screen segmentation in recent scanning systems. The cluster scanning system performs best when the number of clusters is five. However, among the different versions of cluster and block scanning processes, we found a type of block scanning that divides the screen into four equal sized partitions for four iterations performed best.

We had expected that the cluster scanning process would perform better since it uses the information about target types (eg. labels are not considered as possible targets) and locations in the clustering process. So, as part of a post-hoc analysis, we studied the actual tasks undertaken by our participants. Most of the time, our participants used instant messenger software and browsed the web. The present version of the clustering process does not consider locations of hyperlinks in the target acquisition process and so it might miss possible targets during web surfing. To test our hypothesis we again collected some sample cursor traces in two different conditions: in the first condition we asked users not to browse the web, while in the second there was no such restriction. The estimated time for block scanning (with branching factor four and four iterations) and cluster scanning (with five cluster centres) are shown in *Figure 6*.

We found that that the cluster scanning process performed far better than the block scanning process when it considered all possible targets in its clustering process (ie. in tasks without web browsing). The intended audience of the scanning systems (motor-

Table 1: Estimated task completion time for different scanning systems

Scanning type	Branching factor (#clusters or #blocks)	Number of iterations (for block scanning)	Estimated task completion in seconds
Eight-directional scanning			8676
Cluster scanning	2		6943
	3		5996
	4		5842
	5		5707
	6		5937
	7		5965
Block scanning	2	2	7595
	2	4	7859
	2	8	7781
	4	1	7206
	4	2	7116
	4	4	5374
	16	1	8201
	16	2	6961

impaired users) can use special browsers customised for them (Stephanidis *et al*, 1998; IBM Web Adaptation, 2007). In those browsers, a web page is pre-processed before presentation and the hyperlinks are arranged in a fixed location of screen. In that case, the cluster scanning process will have no problem locating hyperlinks and should perform better than other scanning systems.

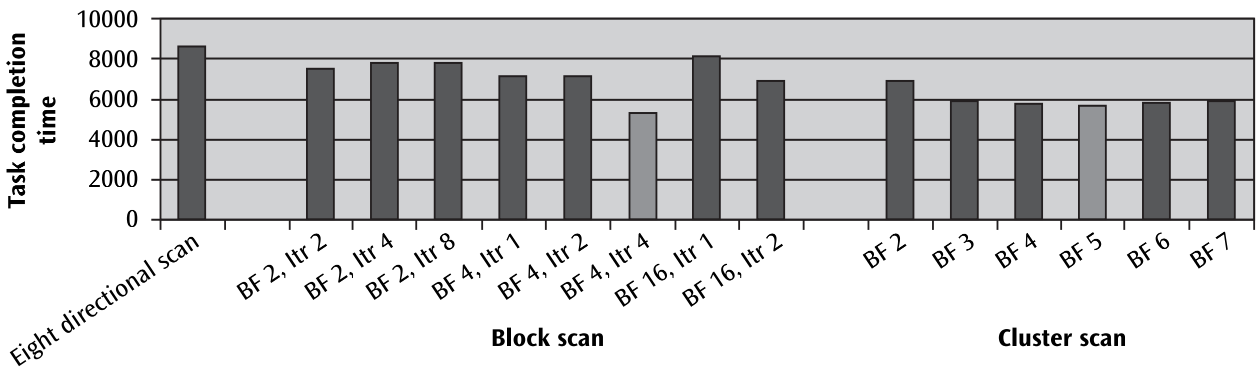
Implications of the study

We can summarise the implications of our study in the following two points.

Extending the scope of scanning

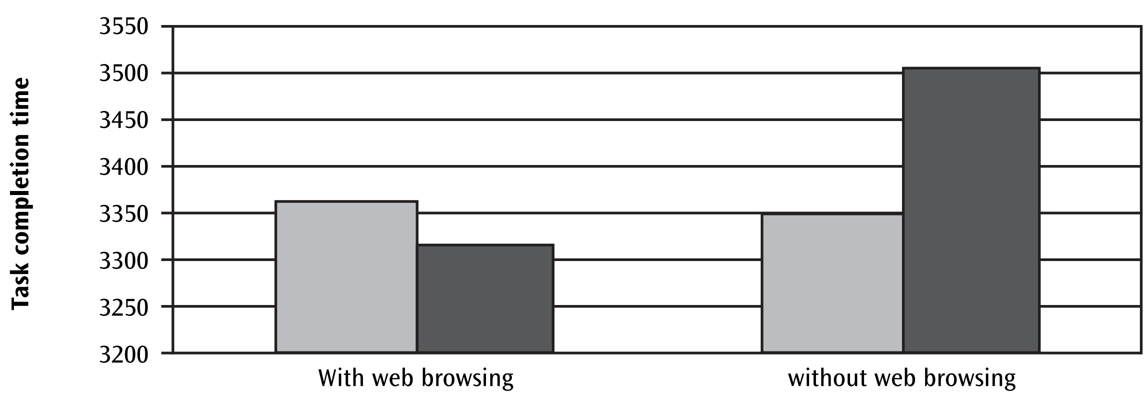
In the field of assistive technology, scanning systems are mainly used for alternative and augmentative communication (AAC) devices. However, AAC systems are not sufficient to engage a user more fully with the world – other software like internet browser, word processor etc are also quite useful for people with disabilities. Recent studies (Steriadis & Constantnou, 2002; Ntoa *et al*, 2004; Gus Scanning Cursor, 2007; The ScanBuddy system, 2007) already used scanning systems for other software. Our study keeps up that trend and provides a new scanning system that can be used to access any software. It will also be equally applicable for AAC systems since block-row-item based scanning can be considered as a special type of cluster scanning, where the items are manually clustered into blocks and rows. Researchers of AAC systems can think about clustering unequal number of items into blocks or rows based on their frequency of use or position in the screen.

Figure 5: Performance comparisons of different scanning systems



BF: Branching factor
ITR: Number of iterations

Figure 6: Comparing cluster scanning and block scanning for tasks using and not using internet



	With web browsing	without web browsing
Cluster scanning	3359.567	3349.592
Block scanning	3316.63	3504.617

Automatic evaluation

Our study also pioneers an idea of automatically evaluating assistive interfaces using a simulator (Biswas & Robinson, 2008). Before running a formal user trial, a system designer may tune interface parameters (as we did to find out the optimum number of clusters or blocks) or select the best design alternative using our simulator. For example, a researcher can find out the best set of parameters for a scanning adaptation algorithm using our simulator and then evaluate the algorithm with that set of parameters by a proper user trial. Similarly a researcher of AAC can initially record some conversational text from able-bodied users (as we collected interaction patterns for this study) and then evaluate different arrangements of items (letters, words or phrases) in a screen to produce the same set of conversation. In this way, the time and cost of evaluation can be reduced by using simulated situations.

Conclusions

Scanning is the technique of successively highlighting items on a computer screen pressing a button when the desired item is highlighted. People with severe motor-impairment (like ALS, cerebral palsy etc) use this technique to access a computer. In this paper, we have compared different scanning systems on some real life tasks. We introduced a new scanning technique based on clustering screen objects and compared it with two other scanning systems using a simulator. The simulator is based on the mathematical models of the scanning systems and it can predict the possible interaction patterns and task completion time for different scanning systems. Initially it was found that the best scanning system is a type of block scanning that divides the screen in four equal sized partitions for four iterations and then switch to eight-directional scanning. However with a more accurate target acquisition process, the cluster scanning technique was found to have the potential to outperform other scanning systems. The analysis and results also encourage the use of modelling and simulation for evaluating performances of different assistive interfaces.

Addresses for correspondence

Pradipta Biswas
Research Student
Computer Laboratory, University of Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD
UK
Email: pb400@cam.ac.uk
Website: www.cl.cam.ac.uk/~pb400

Peter Robinson
Professor of Computer Technology and Deputy Head of the Department
Computer Laboratory, University of Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD
UK
Email: pr10@cam.ac.uk
Website: www.cl.cam.ac.uk/~pr

Acknowledgement

We would like to thank the Gates Cambridge Trust for funding this work. We are also grateful to the students of Computer Laboratory and Trinity College, Cambridge, for taking part in our experiments.

References

Angelo J (1991) Comparison of three computer scanning modes as an interface method for persons with cerebral palsy. *The American Journal of Occupational Therapy* **46** (2) 217–222.

Birch S (2000) *Single-switch Mouse Emulation: A tale of two methods*. Proceedings of the RESNA 2000 Annual Conference 28 June–2 July 2000. Orlando, FL, USA.

Biswas P & Robinson P (2007) *Simulation to Predict Performance of Assistive Interfaces*. Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'07). Tempe, Arizona, USA.

Biswas P & Robinson P (2008) *Automatic Evaluation of Assistive Interfaces*. In Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI 2008) January 2008. Canary Islands, Spain.

Blackstien-Adler S, Shein F, Quintal J, Birch S & Weiss PL (2004) Mouse manipulation through single-switch scanning. *Assistive Technology* **16** (1) 28–42.

Evreinov G & Raisamo R (2004) Optimizing menu selection process for single-switch manipulation. Proceedings of the ICCHP 2004 (pp836–844).

Gus Scanning Cursor (2007) Available from: www.turningpoint technology.com/Sw/GS/ScanningCursor.htm [accessed 23 July 2008].

IBM Web Adaptation (2007) Available from: www.webadapt.org [accessed 23 July 2008].

Lesher G, Moulton B and Higginbotham DJ (2002) *Acquisition of Scanning Skills: The use of an adaptive scanning delay algorithm across four scanning displays*. In Proceedings of the Annual Conference of RESNA 2002.

Moynahan AJ & Mahoney RM (1996) *Single Switch Mouse Control Interface*. In Proceedings of the Annual Conference of RESNA, 1996.

Ntoa S, Savidis A & Stephanidis C (2004) *FastScanner: An accessibility tool for motor impaired users*. In Proceedings of the ICCHP 2004, LNCS 3118 (pp796–803).

O'Neill P, Roast C & Hawley M (2002) *Evaluation of Scanning User Interfaces Using Real Time Data Usage Logs*. In Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility (ASSET 2002) (pp137–141).

Ross TJ (1997) *Fuzzy Logic with Engineering Application*. International Edition. New York: McGraw-Hill.

Shein F (1997) *Towards Task Transparency in Alternative Computer Access: Selection of text through switch-based scanning*. PhD Thesis. Toronto: Department of Mechanical and Industrial Engineering, University of Toronto.

Simpson RC & Koester HH (1999) Adaptive one-switch row-column scanning. *IEEE Transactions on Rehabilitation Engineering* **7** (4) 464–473.

Stephanidis C, Paramythis A, Sfyrakis M, Stergiou A, Maou N, Leventis A, Paparoulis G & Karagiannidis C (1998) Adaptable and adaptive user interfaces for disabled users in the AVANTI project. *Intelligence in Services and Networks* **1430** 153–166.

Steriadis CE & Constantnou P (2002) *Using the Scanning Technique to Make an Ordinary Operating System Accessible to Motor-impaired Users. The Autonomia system*. In Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility (ASSET 2002).

The ScanBuddy system (2007) Available from: www.ahf-net.com/Scanbuddy.htm [accessed 23 July 2008].