Performance Comparison of Different Scanning Systems using a Simulator

Pradipta BISWAS[1], Peter ROBINSON
Computer Laboratory, University of Cambridge

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 switch to eight directional scanning. However with a more accurate target acquisition process, the cluster scanning technique is found to have the potential to outperform other scanning systems.

Keywords. Human Computer Interaction, Assistive Technology, Single Switch Scanning, Simulation

Introduction

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 [1] [2]. However, navigation through 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 [3] [4](and in a wheelchair mobility interface [5]). In both Cartesian and polar scanning systems, the interaction rate of users remains very low. So 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 [3], the FastScanner system [4], the Gus! Scanning Cursor [6], the ScanBuddy system [7] and the SSMCI

[1]Corresponding Author: Pradipta Biswas, Computer Laboratory, University of Cambridge, E-mail: pb400@cam.ac.uk

system [8].

The Autonomia system [3] replaces the windows and widgets of a typical Windows interface by Frames and Wifsid (Widget for Single-switch input devices) respectively. The system consists of different frames like 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 [4] starts the scanning process by showing a list of currently open applications and asks the user to choose an application. The scanning procedure operates on the selected application. The objects of an application are scanned sequentially based on a predefined order. Screen navigation is done by eight-directional scanning. Additionally, the controls in an application are divided into four classes (viz. Text entry objects, simple objects, selection objects and container objects) and the user input is interpreted according to the type of the object that has received the input. The Gus Scanning Cursor [6] provides different types of navigation strategies (viz. Cartesian, Polar, eight-directional) at a single screen and the screen was scanned by row-item based scanning. The user needs to choose a particular scanning type to navigate through the screen. The ScanBuddy system [7] scans the screen by iteratively dividing it into two equal parts up to 4 times. Finally it scans the smallest part using Cartesian scanning. In the SSMCI (Single switch Mouse Control Interface) system [8], 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 reevaluates the situation and comes up with a new solution.

Most of these existing scanning systems (excepts [6] and [8]) have a generic structure. They start with dividing the screen into several blocks and then introduce either Cartesian or polar scanning within a block. As a result, users need to 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 (like cursor frame, console frame etc.). The FastScanner system defines blocks based on the hierarchy of objects in the Windows operating system. The scan buddy system defines blocks just by dividing the screen in two equal segments.

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. In the next section we describe each of these scanning systems. The performance of these scanning systems is compared in section 2.

1. The 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 (viz. 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 keypress to stop the pointer movement and make a click.

Cluster Scanning System: The cluster scanning system collects all possible targets by enumerating window processes (currently it operates only for Microsoft Windows operating system). Then it iteratively divides a screen into several clusters of targets based on the their locations. The user has to select the appropriate cluster that contains the intended target. After reaching the intended target (or the nearest possible position of it), the system switches to eight-directional scanning. The user can select the target or can navigate through the screen using eight-directional scanning mechanism.

Block Scanning Process: In the block scanning process the screen area is iteratively segmented into equal sized sub-areas. The user has to select a sub-area that contains the intended target. The segmentation process runs a certain number of iterations and after that eight-directional scanning is initiated in the selected sub-area.

2. Performance Comparison of the Scanning Systems

We have developed a simulator to simulate the interaction pattern 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. 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 aware that their interactions were being recorded. 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.

2.1. 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 time is shown in table 1 and fig. 1.

2.2. Discussion

The result clearly shows 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. 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 have expected that the cluster scanning process would perform better since it uses the information about target types (e.g. labels are not considered as possible targets) and locations in the clustering process. So as a part of a post-hoc analysis we studied the actual tasks undertaken by our participants. Most of the time our participants used chatting software and browsed the Internet. 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 Internet surfing. To test our hypothesis we again collected some sample cursor traces in two different conditions: in the first condition we forbade users from using the Internet while in the second there was no such restriction. The estimated time for block scanning (with branching factor 4 and no. of iterations 4) and cluster scanning (with no. of cluster centers 5) is shown in fig. 2.

Table 1 Estimated Task Completion Time for different scanning systems

Scanning Type	Branching Factor (#Clusters or #Blocks)	Number of Iterations (For Block Scanning)	Estimated Time in sec.
Eight-Directional Scanning			8676.316
Cluster Scanning	2		6943.452
	3		5995.769
	4		5842.424
	5		5706.539
	6		5936.574
	7		5965.293
Block Scanning	2	2	7595.362
	2	4	7859.33
	2	8	7780.873
	4	1	7206.349
	4	2	7116.49
	4	4	5374.18
	16	1	8201.309
	16	2	6961.48

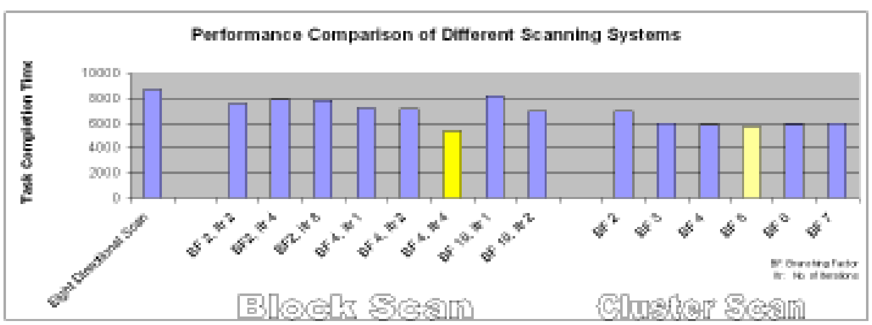

Figure 1 Performance Comparison of Different Scanning System (B.F.: Branching Factor, Itr: #Iterations)

We found that that the cluster scanning process performed far better than the block scanning process when it can consider all possible targets in its clustering process (i.e. in tasks without web Browsing). The intended audience of the scanning systems (motor-impaired users) can use special browsers customized for them ([9][10]). In those browsers, a web page is preprocessed 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.

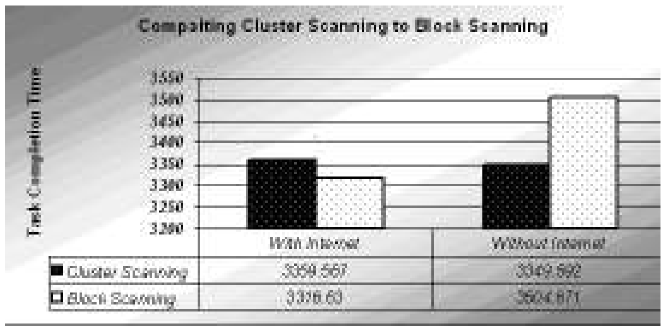

Figure 2 Comparing Cluster Scanning and Block Scanning for tasks using and not using Internet

3. Conclusions

In this paper, we compare different scanning systems on some real life tasks. We introduce a new scanning technique based on clustering screen objects. We have developed a simulator that can predict the possible interaction pattern and task completion time for different scanning systems. The cluster scanning technique is compared with two other scanning systems. Initially it has been 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 is found to have the potential to outperform other scanning systems.

References

[1] Simpson R. C., Koester H. H., Adaptive One-Switch Row-Column Scanning, IEEE Transactions on Rehabilitation Engineering, Vol. 7, No. 4, Dec 1999, pp. 464 -473

[2] Lesher et. al., Acquisition of scanning skills: The use of an adaptive scanning delay algorithm across four scanning dispays, ,Proc. of Annual Conference of RESNA, 2002

[3] Steriadis C. E., Constantnou P., Using the scanning technique to make an ordinary operating system accessible to motor-impaired users. The Autonomia system, Proc. of ASSET 2002

[4] Ntoa S., Savidis A., Stephanidis C., FastScanner: An Accessibility Tool for Motor Impaired Users, ICCHP 2004, LNCS 3118, 2004, pp. 796-803.

[5] O'Neill P., Roast C., Hawley M., Evaluation of scanning user interfaces using real time data usage logs, Proc. of ASSET 2002 pp 137-141

[6] Gus Scanning Cursor, Available at: www.turningpointtechnology.com/Software/GS/ScanningCursor.htm

[7] The ScanBuddy system, Available at: www.ahf-net.com/Scanbuddy.htm

[8] Moynahan A. J., Mahoney R. M., Single Switch Mouse Control Interface, Proc. of Annual Conference of RESNA, 1996

[9] Stephanidis, C. et. al., "Adaptable and Adaptive User Interfaces for Disabled Users in the AVANTI Project" Intelligence in Services and NetworksLNCS-1430, Springer-Verlag 1998, pp 153-166

[10] IBM Web Adaptation, Available at: http://www.webadapt.org/