Evaluating the cluster scanning system

Pradipta Biswas and Peter Robinson

University of Cambridge Computer Laboratory
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK
E-mail: {pb400, pr}@cl.cam.ac.uk

Abstract. Scanning is the technique of successively highlighting items on a computer screen and pressing a switch when the desired item is highlighted. We have developed a new scanning system for single switch input that works by clustering objects on the screen. This paper presents the underlying algorithm and reports on an evaluation study comparing it with a conventional block scanning system. Motor-impaired users found the cluster scanning system faster and more accurate than block scanning.

1. Introduction

Many physically challenged users cannot interact with a computer through a conventional keyboard and mouse. For example, Spasticity, Amyotrophic Lateral Sclerosis, and Cerebral Palsy confine movement to a very small part of the body. People with these disorders 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.

We have developed a new scanning system that works by clustering screen objects. In a previous paper we compared the cluster scanning system with other scanning systems using simulation [1]. In this paper we validate the results obtained in simulation by evaluating the cluster scanning system through a controlled experiment on motor-impaired users. We describe the scanning systems and results obtained using simulation in the following two sections. In Section 4, we present our study followed by concluding remarks.

2. The scanning systems

In this study we compared two technologies: a conventional block scanning system and our new cluster scanning system.

Block Scanning System

A block scanning system iteratively segments the screen into equally sized sub-areas. The user has to select a sub-area that contains the intended target. The segmentation process iterates until the sub-area contains a single target (Figure 1).

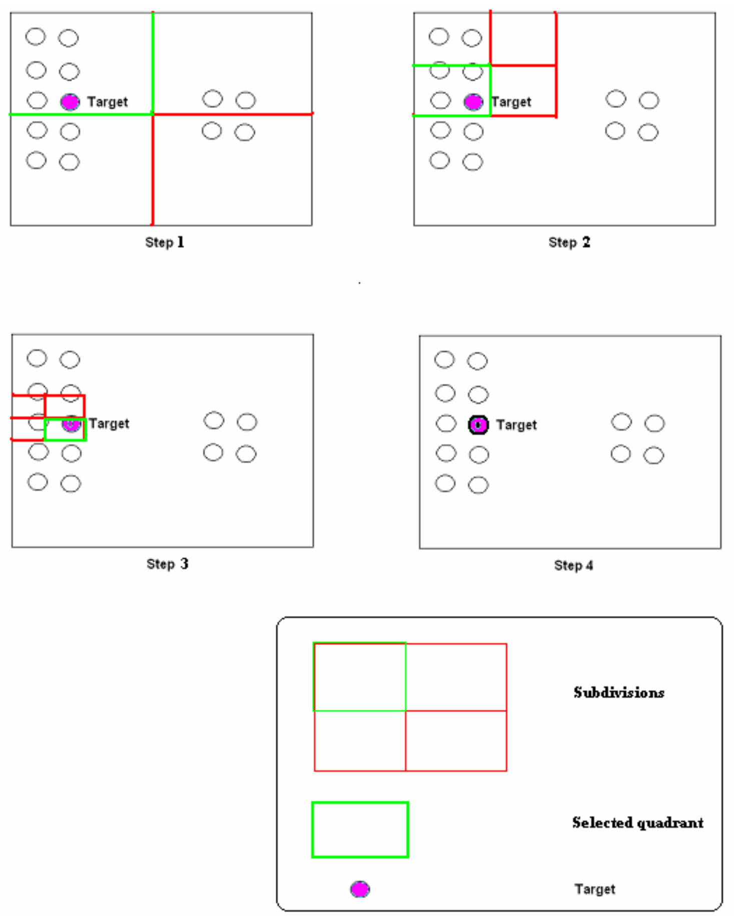

Fig. 1. The Block Scanning System

The cluster scanning system

The cluster scanning system initially collects all possible targets in a screen. Then it iteratively divides a screen into several clusters of targets based on their locations (Figure 2). The user has to select the cluster that contains the intended target. The clustering process iterates until the cluster contains a single target (Figure 1).

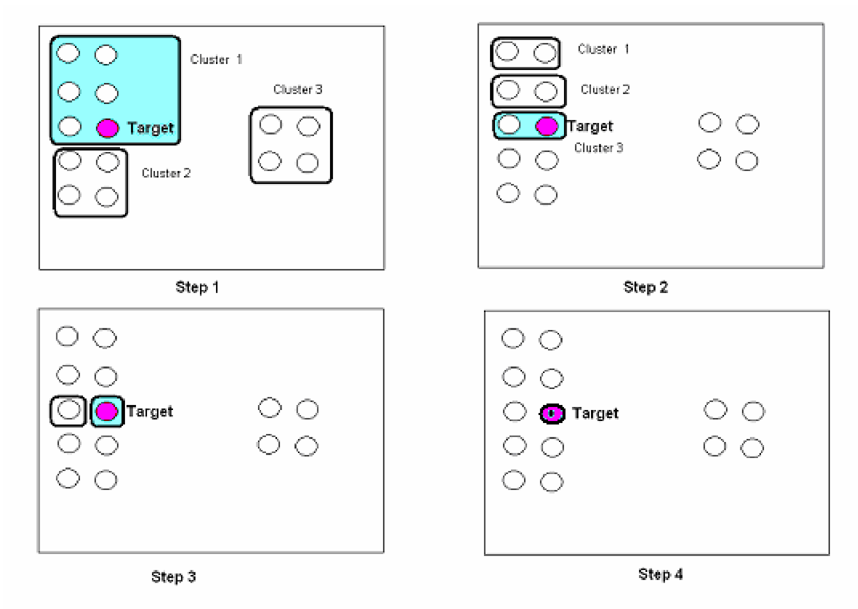

Fig. 2. The Cluster Scanning System

The cluster scanning system works by enumerating objects being shown in the screen and storing positions of windows, buttons, icons and other possible targets. The algorithm starts by considering all the processes running on the computer. If a process is controlling a window, then the algorithm also considers all child and thread processes owned by it. During the enumeration process, the algorithm identifies the foreground window and separately stores the positions of the foreground window and targets within it from the background windows. The algorithm also calculates the area occupied by the foreground window. Then it separately clusters the targets in the foreground window and background windows. The ratio of the number of clusters in foreground and background windows is proportional to the ratio of the area occupied by the foreground window in the whole screen. We used the Fuzzy c-means algorithm [4] to cluster the targets into similarly sized groups. The algorithm is similar to k-means algorithm, but instead of putting the data points into separate clusters, it returns the membership values of data points into different clusters. As a result when the data points are not naturally separated, it returns overlapping clusters.

3. Evaluation through simulation

We recorded sample interactions by two able-bodied users to generate a list of tasks, which were fed to the simulator [2]. The model for the cluster scanning system takes the scan delay, the number of clusters, the intended target and the total number and positions of targets in a screen as input and gives the target acquisition time as output by running the cluster scanning algorithm on the input. The model for the block scanning system takes the scan delay, the number of blocks and the total number of targets in a screen as input and gives the target acquisition time as output which is equal to $s \times \log_k n$, where

s is the scan delay

k is the number of blocks

n is the number of target

It should be noted that for the block scanning system, the target acquisition time is constant for any target in the screen.

We investigated the block scanning system for different numbers of blocks and different numbers of iterations, and the cluster scanning system for different numbers of clusters. The cluster scanning system performed best when the number of clusters was five. However, among the different versions of Cluster and Block scanning processes, we found that a type of block scanning that divided 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 information about target types (e.g. labels are not considered as possible targets) and locations in the clustering process. So we extended the analysis to consider the actual tasks undertaken by our participants. Most of the time our participants used instant messenger software and browsed the World Wide 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. The further study revealed that participants should take less time to complete a task using the cluster scanning system than other scanning systems if the clustering process could include all targets in a screen. Detail of this study can be found in our previous paper [1].

4. The experiment

We validated the results obtained in simulation by a controlled experiment on motor-impaired users. We discuss the detail of the experiment in following sections.

4.1. Procedure

In this experiment, the participants were instructed to press a set of buttons placed in a screen (Figure 3) in a particular sequence. All the buttons were coloured grey except the next target, which was red. The same task was repeated for all the scanning systems. In particular, we evaluated the cluster and block scanning systems. We recorded the cursor traces, target height, width, and task completion time. For internal validity of the experiment, we did not use any scan delay adaptation algorithm. The scan delay was kept constant at 2 sec. for motor-impaired participants and at 1 sec. for the control group. These values were selected after observing their reaction time and were greater than the reaction time. All participants were trained adequately with the scanning systems before undertaking the experiment.

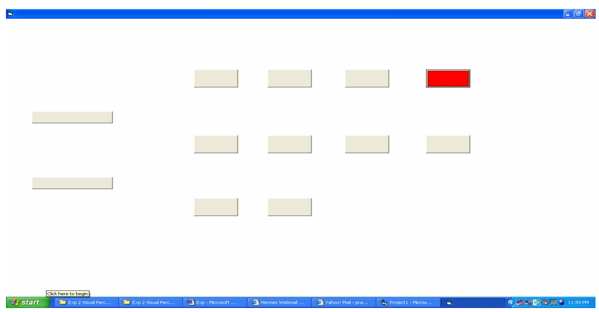

Fig. 3. Screenshot of the experiment

4.2. Material

We used a push button switch [7] and an Acer Aspire 1640 Laptop with 1280×800 pixel screen resolution. We also used the same seating arrangement (same table height and distance from table) for all participants.

4.3. Participants

We collected data from 8 motor-impaired and 8 able-bodied participants (Table 1). The motor-impaired participants were recruited from a local centre, which works on treatment and rehabilitation of disabled people and they volunteered for the study. All motor-impaired participants used computer at least once each week. Able-bodied participants were students of our university and expert computer users. None of them had used the scanning systems before.

Table 1. List of Participants

	Age	Gender	Impairment
C1	27	F	Able-bodied
C2	28	F	Able-bodied
C3	30	M	Able-bodied
C4	30	M	Able-bodied
C5	31	M	Able-bodied
C6	28	F	Able-bodied
C7	30	F	Able-bodied
C8	26	F	Able-bodied
P1	30	M	Cerebral Palsy resulting manual dexterity, wheel chair user.
P2	43	M	Cerebral Palsy resulting manual dexterity, also have tremor in hand, wheel chair user.
P3	30	M	Dystonia, cannot speak, cannot move fingers, wheelchair user.
P4	62	M	Left side (non-dominant) paralysed after a stroke in 1973, also have tremor.
P5	44	M	Cerebral attack, significant tremor in whole upper body part, fingers always remain folded.
P6	46	F	Did not mention disease, hard to grip things, no tremor.
P7	>45	F	Spina Bifida/ Hydrocephalus, wheelchair user.
P8	>45	M	Cerebral Palsy from birth, restricted hand movement, no tremor.

4.4. Results

Initially we measured the total task completion time for the scanning systems (Figure 4). It can be seen that participants took less time to complete the task using the cluster scanning system. The dotted bars in Figure 4 mean that two participants could not complete the task using the block scanning system.

To further investigate the scanning systems, we measured the following three dependent variables:

Number of missed clicks: We counted the number of times the participants wrongly pressed the switch.

Idle Count: The scanning systems periodically highlight the buttons. This variable measures the number of cycles when the participants did not provide any input, though they were expected to do so.

Efficiency: The scanning systems require a minimum time to complete any task which depends on the particular scanning system and not on the performance of the user. We calculated the efficiency as the ratio $\frac{OptimalTime}{ActualTime}$. An efficiency of 100% indicates optimal performance, 50% indicates taking twice the minimal time and 0% indicates failure to complete the task.

We did not find any significant difference between the performances of motor-impaired and able-bodied users by an equal variance paired t-test at p<0.05 level. However the average number of missed clicks and idle count are significantly lower in the cluster scanning system than in the block scanning system in an equal variance paired t-test (p<0.05) (Figure 5). Additionally two participants (P3 and P7) could not complete the task using the block scanning system while all participants could complete the task using the cluster scanning system.

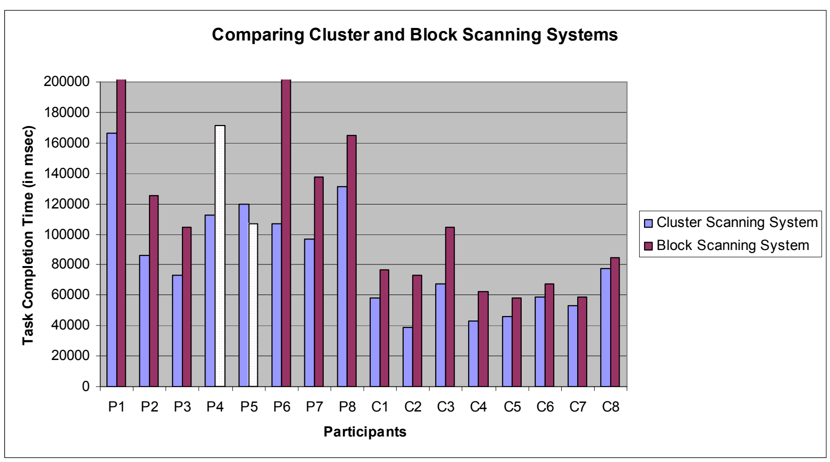

Fig. 4. Task completion times for the scanning systems

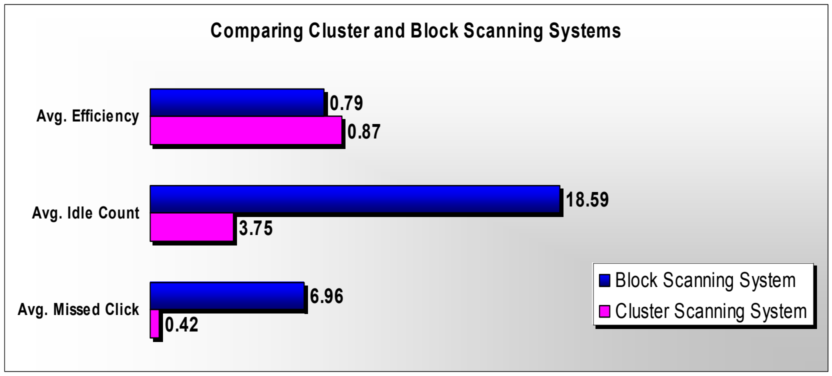

Fig. 5. Comparing the scanning systems

4.5. Discussion

The simulator predicted that the task completion time would be less in the cluster scanning system than the block scanning system when the cluster scanning system can consider all possible targets in its clustering process. The experiment also shows similar results. The total task completion time, sub-optimal task completion time, idle time and number of missed clicks are less in the cluster scanning system than the block scanning system. The efficiency of the cluster scanning system can be attributed to the following factors.

- The cluster scanning system does not introduce any new interface element like a frame or form in the screen as Autonomia [5] or FastScanner [3] systems do.
- The cluster scanning system does not blindly divide the screen in a predefined number of segments as the ScanBuddy [6] or block scanning systems do. 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.

However as the optimal task completion time was higher in the block scanning system than the cluster scanning system, so we did not find the difference in efficiency significant in a paired t-test ($p = 0.07$).

Our study also confirms the value of automatically evaluating assistive interfaces using a simulator [1, 2]. Before running a formal user trial, a system designer may tune interface parameters or select the best design alternative using our simulator. As each alternative design does not need to be evaluated by a user trial, the simulator will reduce the development time significantly.

5. Conclusion

We have developed a new scanning system that works by clustering screen objects. In a previous paper we evaluated the cluster scanning system and found it superior to the block scanning system. In this paper we have presented a study of motor-impaired users to evaluate the cluster scanning system in practice and to validate the results of the simulation. We found that the total task completion time, idle time and number of missed clicks are less in the cluster scanning system than for the block scanning system. The results also in turn validate the simulator. So we can infer that motor-impaired users found the cluster scanning system faster, easier and more accurate than the conventional block scanning system

References

1. Biswas P. and Robinson P., A New Screen Scanning System based on Clustering Screen Objects, Journal of Assistive Technologies, Vol. 2 Issue 3 September 2008, ISSN: 1754-9450
2. Biswas P. and Robinson P., Automatic Evaluation of Assistive Interfaces, ACM International Conference on Intelligent User Interfaces (IUI) 2008
3. Ntoa S., Savidis A. and Stephanidis C. (2004). FastScanner: An Accessibility Tool for Motor Impaired Users, In Proceedings of the ICCHP 2004, LNCS 3118, 2004, pp. 796-803.
4. Ross T. J., Fuzzy Logic with Engineering Application, International Edition, Mcgraw-Hill Inc., 1997
5. Steriadis C. E. and 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)
6. The ScanBuddy system (2007). Available from: www.ahf-net.com/Scanbuddy.htm, Accessed on 21st May 2007
7. The Super-Switch (2007), Available at: http://rjcooper.com/super-switch/index.html, Accessed on: 1st July, 2007