Vanishing windows: a technique for adaptive screen management

This item was submitted to Loughborough University’s Institutional Repository by the/ an author.

Additional Information:


Metadata Record: [https://dspace.lboro.ac.uk/2134/27081](https://dspace.lboro.ac.uk/2134/27081)

Publisher: © Tunu Miah

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 2.5 Generic (CC BY-NC-ND 2.5) licence. Full details of this licence are available at: [http://creativecommons.org/licenses/by-nc-nd/2.5/](http://creativecommons.org/licenses/by-nc-nd/2.5/)

Please cite the published version.
This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/
Pilkington Library

Author/Filing Title .... M . A . H ........................................
Vol. No. ..........   Class Mark .........................

Please note that fines are charged on ALL overdue items.

[Signature]

0402153871
VANISHING WINDOWS: A TECHNIQUE FOR ADAPTIVE SCREEN MANAGEMENT

By

Tunu Miah

Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of

Doctor of Philosophy at Loughborough University

October, 1998

© Tunu Miah 1998
Abstract

Windowing systems offer many benefits to users, such as being able to work on multiple tasks concurrently; or working with a number of windows, each connected to different remote machines or applications. Unless these windows are managed efficiently, users can easily become overwhelmed by the number of currently open windows and lose their way round the desktop. This can lead to a state where the desktop is cluttered with windows. At this stage “window thrashing” occurs, as users begin to perform window management operations (move, resize, minimise etc.) in order to locate relevant pieces of information contained in one of several open windows.

This thesis proposes a novel technique known as Vanishing or Fading Windows to reduce window manipulation and aid the user in search activities. One of the features of this technique is that some of the burden of the window management operation is taken over by the system, hence allowing the user to focus more on application domain activities. The technique involves progressively reducing the size of inactive windows whilst the user is interacting with the active window thereby reducing the clutter on the desktop. If a reduced window is reactivated, it immediately returns to its original size. Two basic reduction techniques have been examined – cropping and scaling. Experiments have been performed to determine the effect that screen size and presentation styles have on the users’ ability to recognise the content of a window.

A number of techniques have been investigated to increase the recognition rate of a reduced window, the most useful of which packs the window with an increased number of important keywords as it reduces. These keywords are derived automatically by comparing the word frequencies in the document with a standard corpus.
Finally, empirical evaluation of the Vanishing Windows system has been conducted and subjective views of the system collected from users. The results from the final experiment are encouraging and show significantly lower user window management operations when perform a task using Vanishing Windows.
Acknowledgements

I would like to thank all those who helped with the work undertaken in this thesis especially:

- My Supervisor, Professor James Alty, and my Director of research Dr. Chris Hinde, for their valuable guidance, and encouragement.

- Miss Karen Mackenzie-Mills and Dr. Ghada Kadoda for their encouragement and support during times of despair.

- My family who have been a constant source of support.

- Colleagues from LUTCHI Research Centre and from the department of Computer Studies.
# Table of Contents

ABSTRACT I

ACKNOWLEDGEMENTS III

TABLE OF CONTENTS IV

LIST OF FIGURES XI

LIST OF TABLES XIII

LIST OF GRAPHS XVI

CHAPTER 1
GENERAL INTRODUCTION 1

1.1 Background 1
1.2 The Objectives of the Research 7
1.3 Research Methodology 7
1.4 Main Achievements 7
1.5 Guide to the Thesis 10

CHAPTER 2
REVIEW OF WINDOWING SYSTEMS AND ADAPTIVE USER INTERFACES 14

2.1 A Brief History of Window Systems 14
2.2 Characterising Windowing Systems 15
  2.2.1 Interaction Styles 15
  2.2.2 Set of Operations 18
  2.2.3 Presentation Style 20
    2.2.3.1 Tiling Windowing Systems 20
    2.2.3.2 Overlapping Windowing Systems 22
    2.2.3.3 Comparison of Tiled and Overlapping Windows 24
2.3 Clutter 29
  2.3.1 What is Clutter? 29

2.4 Complexity Theory 33
  2.4.1 How Does Shannon’s Theory of Communication relates to GUI’s? 34
  2.4.2 Complexity Theory – How Does it Apply to Window Clutter? 38
## 2.4.3 Clutter Analysis Applied to Windowing Systems
- 2.4.3.1 Window Size Determination
- 2.4.3.2 How Clutter Analysis May Benefit Windowing Systems

## 2.5 Adaptive and Intelligent Window Managers
- 2.5.1 Adaptive Systems
- 2.5.2 Window Managers
- 2.5.3 Related Work
  - 2.5.3.1 Window Layout Constraint Based Approach
  - 2.5.3.2 Multiple Window Co-ordination Based Approach
  - 2.5.3.3 Miscellaneous Approaches

## 2.6 The Need to Manipulate Windows

## 2.7 Evaluation Methods
- 2.7.1 Common Approaches to Evaluation
  - 2.7.1.1 Analytic Evaluation (Formal modelling as an approach to evaluation)
  - 2.7.1.2 Expert Evaluation (Walkthrough approaches)
  - 2.7.1.3 Observational Evaluation
  - 2.7.1.4 Survey Evaluation
  - 2.7.1.5 Experimental Evaluation
- 2.7.2 Selecting an Evaluation Method
- 2.7.3 Usability testing

## 2.8 Summary

### CHAPTER 3
**HOW ARE WINDOWS USED: AN EMPIRICAL STUDY**

## 3.1 Introduction

## 3.2 Approaches to Window Usage
- 3.2.1 Generic Windows Tasks

## 3.3 Experiment: Observing Users Working with Windows
- 3.3.1 Objective of the Experiment
- 3.3.2 Experimental Design
  - 3.3.2.1 The Task
  - 3.3.2.2 Set-up and Procedure

## 3.4 Analysis of Results
- 3.4.1 Notation Used to Transcribe the Video Recordings
- 3.4.2 Observations

## 3.5 Overall Observation

## 3.6 Summary
CHAPTER 4
THE CONCEPT OF VANISHING WINDOWS

4.1 Theoretical Considerations

4.2 The Vanishing Windows Technique: Concept and Issues
  4.2.1 When is a window no longer needed?
  4.2.2 How does the system reduce the window size?
  4.2.3 Vanishing Windows Parameters
    4.2.3.1 Rate of Progress to Iconification
  4.2.4 Vanishing Strategies

4.3 Algorithms for Vanishing Strategies
  4.3.1 Top Level Algorithm
  4.3.2 Vanishing Operations
    4.3.3 Operations Employed in the Vanishing Windows System
      4.3.3.1 Migrate - Formulation of Rules for Direction of Migration
      4.3.3.2 Shrink – Formulation of Rules for Direction of Reduction
      4.3.3.3 Shunt – Formulation of Rules
      4.3.3.4 Evade – Formulation of Rules
  4.3.4 Intelligent switching between Shrink and Migrate Operations

4.4 Metrics for Evaluating Vanishing Algorithms
  4.4.1 Distance From Original Position and Size
  4.4.2 Make maximum use of the Screen Real Estate

4.5 Some Associated Problems of Vanishing Windows

CHAPTER 5
VISUAL RECOGNITION OF WINDOWS: EFFECTS OF SIZE VARIATION AND PRESENTATION STYLES

5.1 Overview

5.2 Introduction

5.3 Experiment 1: Scaled Verses Cropping
  5.3.1 Experimental Procedure

5.4 Results of Analysis

5.5 Target Document Identification Accuracy
  5.5.1 Removal of Redundant Information
  5.5.2 Recalibrated Results

5.6 Response Time for Correct Identification of Target Document
  5.6.1 Screen Utilisation and Response Times

5.7 Further Strategies for Improving Visual Cueing

5.8 Summary of Results

5.9 What Next?
CHAPTER 7
VANISHING WINDOWS: AN EMPIRICAL STUDY

7.1 Introduction 188
7.2 Evaluating the Vanishing Windows System 188
7.3 Experimental Design 190
  7.3.1 The Hypothesis 190
  7.3.2 The Task 190
  7.3.3 Measurements 193
7.4 Experimental Procedure 193
7.5 Results 194
  7.5.1 An Analysis of Vanishing Windows Performance 194
    7.5.1.1 Search Time 194
    7.5.1.2 Number of Window Operations 198
  7.5.2 Analysis of Questionnaire 200
    7.5.2.1 Usefulness of Highlighted Keywords 201
    7.5.2.2 Usefulness of Toolbars Being Removed 201
    7.5.2.3 Usefulness of Vanishing Windows 202
    7.5.2.4 Loss of Control 203
    7.5.2.5 Disruption of Automatic Movement (Auto-Adaptation) 203
    7.5.2.6 Preference of Windowing System 204
  7.5.3 Subjective Comments 205
7.6 Summary 206

CHAPTER 8
SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH

8.1 SUMMARY OF THESIS 208
  8.1.1 CROPPED VERSES SCALING 209
  8.1.2TRANSFORMING WINDOW CONTENTS 210
  8.1.3 VANISHING WINDOWS - USER PERFORMANCE 211
  8.1.4 ECOLOGICAL VALIDITY 212
8.2 THESIS IN RETROSPECT 212
8.3 CONCLUSIONS 214
  8.3.1 POSSIBLE FUNCTIONALITY FOR FUTURE VERSIONS OF VANISHING WINDOWS 215
  8.3.2 POTENTIAL APPLICATIONS 216
8.4 FURTHER RESEARCH 217
  8.4.1 ADDRESSING PROBLEMS OF SYSTEM LIMITATIONS 217
  8.4.2 EXTEND THE EVALUATION 219
APPENDIX A. ANCILLARY INFORMATION FOR CHAPTER 3

A.1. Task Sheet for Observation of Windows Usage
A.2. Questionnaire to Collect User Details
A.3. Transcribed Videotaped Data

APPENDIX B. ANCILLARY INFORMATION FOR CHAPTER 5

B.1. Objective and Instruction for Subjects for Experiment 1: Cropped Verses Scaled Presentation Styles

APPENDIX C. ANCILLARY INFORMATION FOR CHAPTER 6

C.1. Objective and Instruction for Subjects for Experiment 2: Adapted Presentation Styles
C.2. Data Exploration: Descriptive Statistics for Experiment 2 – Accuracy of Identification
C.3. Data Exploration: Descriptive Statistics for Experiment 2 – Response Time for Correct Identification

APPENDIX D. ANCILLARY INFORMATION FOR CHAPTER 7

D.1. Outline and Instruction for Subjects for Experiment 3: Empirical Study of Vanishing Windows
D.2. Task Sheet for Experiment 3
D.3. Questionnaire for Experiment 3
APPENDIX E. PAPERS

E.1. Published Papers


List of Figures by Chapter

Chapter 1
Figure 1-1: Representation of user activity. (a) As represented by Bannon et al. 1983a. (b) In terms of phase transition, Card and Henderson 1987. 3
Figure 1-2: Thesis layout diagram 9

Chapter 2
Figure 2-1: Microsoft’s Win95 window, with direct manipulation controls embedded in its border 17
Figure 2-2: Typical window configuration in a one-dimensional tiling system 21
Figure 2-3: Typical window configuration in a two-dimensional tiling system 21
Figure 2-4: A hierarchical window layout 22
Figure 2-5: Non-hierarchical window layout 22
Figure 2-6: Typical window configuration in an overlapping system 23
Figure 2-7: A task suitable for overlapping 25
Figure 2-8: Diagram of GUI communication system (after Shannon and Weaver, 1959) 35
Figure 2-9: Hypothetical displays (Funke et al., 1993): 40
Figure 2-10: Clutter analysis applied to the desktop window 41
Figure 2-11: Basic classification of adaptive systems 43
Figure 2-12: A simple model of a window manager 45

Chapter 3
Figure 3-1: Tiled approach to task 3 85
Figure 3-2: Overlapping approach to task 3 86
Figure 3-3: Screen shot of screen layout for two windows being used concurrently 87
Figure 3-4: Screen shot of screen layout for the third window 88

Chapter 4
Figure 4-1: Vanishing parameters 96
Figure 4-2: Parameters for Vanishing Windows 97
Figure 4-3: Vanishing Windows by means of window resizing 99
Figure 4-4: Hybrid System (tile/overlapping) 100
Figure 4-5: Top level pseudo code 102
Figure 4-6: State transition diagram for Vanishing Operations 103
Figure 4-7: Two overlapping windows at a diagonal 108
Figure 4-8: Some possibilities for the direction of the Evade operation 110
Figure 4-9: Two strategies for manoeuvers 111
Chapter 5
Figure 5-1: Examples of partial and scaled views 115
Figure 5-2: Reducing window size - (a) original window (b) reducing whilst scaling content of window and (c) reducing whilst cropping content of window 116
Figure 5-3: A simple model of reduction strategies 117
Figure 5-4: Toolbars wasting valuable workspace - (a) window without toolbars and (b) same window with toolbars. 125
Figure 5-5: Model of menu and toolbar size and its effect on visible window area 126
Figure 5-6: Visual cues provided by a document 137

Chapter 6
Figure 6-1: Presenting keywords in the form of a LIST 146
Figure 6-2: Presenting keywords - COMPRESSED SPACES 147
Figure 6-3: Presenting keywords by HIGHLIGHTING 148
Figure 6-4: Switching views to accommodate keywords 149
Figure 6-5: Response Ttime histograms with normal curve for Compressed, Highlighted, List and Normal presentation styles 166
Figure 6-6: Normal probability plots for response time for correct identification 167
Figure 6-7: Histogram for the four presentation styles at complexity level 2 177
Figure 6-8: Histogram for the four presentation styles at complexity level 3 178
Figure 6-9: Histograms for the four presentation styles at complexity level 4 179
Figure 6-10: Convention used for boxplots 180
List of Tables by Chapter

Chapter 2
Table 2-1: Window operations for the creation phase of a window 19
Table 2-2: Window operations for the manipulation phase of a window 19
Table 2-3: Advantages for using tiled and overlapped windows 27
Table 2-4: Disadvantages for using tiled and overlapped windows 28
Table 2-5: Levels of the communication process (from Comber and Maltby, 1996) 35
Table 2-6: Differences among the five evaluation methods (Preece, 1993) 70
Table 2-7: Advantages and disadvantages of the evaluation methods. Preece (1993) and Jeffries et al. (1992) 71

Chapter 3
Table 3-1: Phase for task 1 80
Table 3-2: Phase for task 2 80
Table 3-3: Phase for task 3 81

Chapter 4
Table 4-1: Maneouers available for the Migrate operation 105

Chapter 5
Table 5-1: Sample Screen Distribution 118
Table 5-2: Percentage of Correct, Incorrect and Don't Know answers for both Reduction Strategies 120
Table 5-3: Descriptive statistics for response times for cropped and scaled presentation styles 130
Table 5-4: Descriptive Statistics for Response Time – Cropped Correct Identification 132
Table 5-5: Descriptive Statistics for Response Time – Scaled Correct Identification 133

Chapter 6
Table 6-1: Composition of the dictionary file 143
Table 6-2: Sample words and their associated frequencies 145
Table 6-3: Distribution of trials 151
Table 6-4: Test of homogeneity of variance for number of correct identification across all screen sizes for individual presentation styles 152
Table 6-5: Tests of normality for number of correct identification 153
Table 6-6: Percentage of correct, incorrect and don't know answers for the four presentation techniques 156
Table 6-7: ANOVA table for number of correct identifications for the presentation styles

Table 6-8: Paired two-tail t-test for identification accuracy

Table 6-9: 10x4 Repeated Measures ANOVA table for number of correct identifications

Table 6-10: Repeated Measures ANOVA for identification accuracy for all screen utilisation groups

Table 6-11: Paired two-tail t-test for identification accuracy for all screen utilisation groups

Table 6-12: 4x2 Repeated Measures ANOVA table for number of correct identifications for presentation styles and data set

Table 6-13: Repeated Measures ANOVA for identification accuracy for all screen utilisation groups

Table 6-14: Paired two-tail t-test for identification accuracy for all screen utilisation groups

Table 6-15: User ranking of presentation styles

Table 6-16: Test of homogeneity of variance for response time across all screen sizes for individual presentation styles

Table 6-17: Tests of normality for response time for correct identification

Table 6-18: One way ANOVA of response time for presentation styles and complexity levels

Table 6-19: Bonferroni post hoc multiple comparison test to determine statistical differences between complexity levels

Chapter 7

Table 7-1: Task Ordering

Table 7-2: Schedule for experiment 3

Table 7-3: Descriptive statistics for search time

Table 7-4: Paired 2-tailed t-test for search times

Table 7-5: Descriptive statistics for number of windows operation

Table 7-6: Paired 2-tailed t-test for number of windows operation required to locate required window

Table 7-7: Usefulness of Highlighted keyword
Table 7-8: Usefulness of toolbars being removed 202
Table 7-9: Usefulness of Vanishing Windows 202
Table 7-10: Responses for loss of control 203
Table 7-11: Disruption caused by auto-adaptation 204
Table 7-12: Windowing system preference 204
List of Graphs by Chapter

Chapter 5
Graph 5-1: Identification accuracy for the scaled reduction style 121
Graph 5-2: Identification accuracy for the cropped reduction style 121
Graph 5-3: Accuracy of identification as windows size varies for the crop reduction strategy 121
Graph 5-4: Accuracy of identification as windows size varies for the scaled reduction strategy 122
Graph 5-5: Screen utilisation % against number of correct identification for both reduction strategies 123
Graph 5-6: Comparison of cropped reduction strategy with toolbar and without toolbar (calculated) 128
Graph 5-7: Comparison of scaled reduction strategy with toolbar and without toolbar (calculated) 129
Graph 5-8: Average response times for correct, incorrect and unidentified (DK) results for both presentation styles 131
Graph 5-9: Screen utilisation percentage against response time for both cropped and scaled presentation styles for correct identification 133
Graph 5-10: Response time against screen utilisation percentage for both cropped and scaled presentation styles for the unidentified case 134
Graph 5-11: Response time against screen utilisation percentage for both cropped and scaled presentation styles for incorrect identification case 135

Chapter 6
Graph 6-1: Average number of correct identifications against screen utilisation percentage 158
Graph 6-2: Average response time for correct identification against screen utilisation percentage 169
Graph 6-3: Average number of correct identification against screen utilisation 0-30% step size = 3 174
Graph 6-4: Boxplots grouped by presentation styles - response times for correct identification of target document for presentation styles at Complexity level 2,3, and 4. 181
Graph 6-5: Boxplots grouped by complexity levels - response times for correct identification of target document. 182
Graph 6-6: Mean response time for correct identification for the four presentation styles at the 3 complexity levels 183
Chapter 7
Graph 7-1: Boxplots grouped by windowing system - search time to locate target window for three complexity levels 195
Graph 7-2: Average search times for Low, Medium, and High complexity environment for both windowing systems 197
Graph 7-3: Box plot grouped by windowing system - number of windowing operations for each complexity level 198
Graph 7-4: Average number of windows operation performed for a search task 200
Chapter 1

1 GENERAL INTRODUCTION

This thesis addresses the problem of automating and assisting the task of managing and efficiently accessing windows in a window management system. It is based on the premise that, by freeing the user's cognitive and temporal resources from the task of managing the interface, more of these resources will be available for application domain activities. As the problems and application tasks confronting users become more complex and information intensive, the potential of this approach for improving overall human system performance is enhanced. Bly and Rosenberg (1986) found that, for a database management task, almost half of the user's time is spent in managing the window-based interface. If their findings are representative of all or most computer based tasks that use windowing systems, the concept of automated window management, or adaptive window management, offers considerable potential for increasing human computer effectiveness on these tasks.

1.1 Background

Consider the following:

"Imagine the task of writing a paper using a dining-room table. Drafts, figures, references, outlines, and notes can be spread on the table in a way that makes them easily accessible as the writer proceeds. Imagine the same task now done on an office desk. There is still considerable workspace, but perhaps some of the information must be piled together, necessitating occasional flipping through the piles. Now imagine the task on a very tiny desk. Much of the writer's time must be devoted to 'thrashing' about searching through the papers: New papers dredged up top will cover other papers still in active use, and these, in turn will need to be dredged up and
will cover other papers in use. In addition to the immediate time consequences of 'thrashing', the ensuing chaos will tend to alter the task itself, pushing the writer towards more formal methods of accessing information, such as file folders and note cards, which have their own overheads: time to categorise, memory for categories, and conceptual ambiguities.” (Henderson and Card, 1986).

This basic example identifies the window management problem clearly and has a direct bearing on the way work is carried out in a windowing system.

The window metaphor is appealing because it supports the way people really work. Office based information workers, for example, routinely monitor and manipulate data from a wide variety of sources; they typically spend a great deal of time synthesising, summarising, and reorganising information (Card, 1985). In an investigation of methods used to perform these tasks in a non-computing environment, Malone (1983) studied the way people arranged materials on their desktops. He observed that people tended to position papers to reinforce the way they had categorised tasks. This spatial mapping helped them to structure their work, and served to remind them of uncompleted tasks. As work proceeded, task materials were frequently rearranged to reflect changing priorities.

Bannon et al. (1983) demonstrated that when knowledge workers use a computer screen, rather than paper, they must also deal with constantly changing task priorities. Their analysis of the typical activities and task flows of people using a traditional (non-windowing) computer system showed complex patterns in the way the data and applications were accessed. People seldom completed a single application in one continuous time frame. Instead, they moved from application to application, changing the focus of their work in response to events and interruptions inside or outside the computing environment (see Figure 1-1). Interruption examples might include a knock at the door, a colleague asking them for a coffee or an important email from their manager requesting the financial report for the year end.
Both representations shown in Figure 1-1 illustrate the switching between tasks performed by users. Bannon et al. (1983) went further and identified a number of reasons why users switch from one task to another:

1. Digressing to do tasks that users are reminded of while performing another task ("While I am at it tasks")
2. Time sharing among concurrent demands
3. Tasks with long waits
4. Subtasks
5. "Snags" such as running out of file space.
Card and Henderson (1987), extended the list of reasons to include the following:

6. Interruptions from outside ("please complete your section by 5pm")

7. Shifting to another higher priority task (because it's scheduled now or because some relevant mail came in)

8. Shifting to a specialised environment or application (say, to draw a figure or graph)

Current windowing systems offer some support for this style of working. By allowing users to view more than one window at a time, and by giving users control over some aspects of the way windows are configured, such systems provide the following:

- They allow users to arrange their computer desktop spatially.
- They make it possible to change focus from one task to another.
- They provide a visible memory cache. This feature makes it particularly useful for tasks in which users need to:
  - Integrate information from a secondary file or application into the primary task domain.
  - Monitor changes in a secondary process while they perform a primary task
  - Transfer information from a specific location in one file to a specific location in another (Bury et al., 1982; Card et al., 1984).

It is easier for users in a windowing environment to compare or integrate information from two or more sources, than in a non-windowing environment. In a non-windowing environment they must rely on their own memory, hand written notes, printed copies of computer files, or other sub-optimal technique for working with multiple sources.

The development and implementation of the window metaphor has added a new dimension to the way people think about, and use computers. It is now possible, for
example, to perform multiple tasks in parallel, and to view the results of one task whilst performing another. It is even possible to manipulate objects and parameters in one window and simultaneously view the results of that manipulation, perhaps from a different perspective, in another window. These advantages of windowing environments have led to their rapid and wide-ranging acceptance.

However, there are costs associated with window systems as well. The user must assume the burden of managing the windowing environment. Windows must be created and displayed, placed at desired locations, moved to uncover needed information on other windows, re-sized, exposed and de-exposed, rearranged to meet changing needs and desires, and put “out of the away” (or left to clutter the screen) when no longer needed. Since less of the window is visible, users spend a substantial amount of their time moving, resizing, and scrolling windows (Bury et al., 1985; Card Henderson, 1987; Sandberg-Diment, 1984; Steinbrecher, 1984). These tasks are added to the user’s existing application domain tasks. They do not contribute to user productivity. Of course, many of these operations are performed in a non-windowing environment as well, but they are less complex and usually take less overall time. Johnson-Laird (1985:p.37) put it this way “we are now (with the introduction of windows) challenging a not completely confident or competent user to control, what is in effect, several different computers.”

It is generally accepted that the added power achieved through the use of window systems more than offsets the burden of window management imposed on the user. But research on this question is not completely supportive of this belief. Bury, Davies and Darnell (1985) investigated the impact of windows on completion times for a series of information retrieval tasks. Contrary to the author’s predictions, subjects took a significantly longer amount of time to complete the task using windows than in a non-windowing environment. A detailed analysis of the results suggested that time spent doing the task was less in the window condition, but the time spent on window management operations increased the overall task time.
Experiments by Bly and Rosenberg (1986) also suggested that a large determinant in the time taken to solve a problem using a window system is the time spent manipulating the windows themselves.

In another study, Davies, Bury and Darnell (1985) found that, for tasks requiring supplemental information relative to the primary task, user performance was more error-free in a window environment than in a non-window environment. However, subjects took significantly longer time to complete the tasks. Their study indicated that the additional time spent was due to window management operations, and that the reduction in errors was not simply the result of having spent more time on the task. A significant time differential was evident even when all errors had to be corrected. Apparently, the overhead of window management added a significant time burden.

With the increase in the number of windows, visualising simultaneously all the necessary information for a task has become difficult. As the number of windows per task increases, task switching becomes more time consuming since more windows need to be opened/closed, moved/resized (located) under the independent overlapping windows approach. Longer delays due to housekeeping further increase task completion time because of the loss of the users' mental task context, kept in short term memory. Large numbers of windows also prevent users from seeing the overview of their desktop due to overlapping windows. This may delay users in switching to unfinished tasks.

The contents of short-term memory are not only effected by the time that passes, but also by the type of work carried out during that period. Since window housekeeping is an activity related to the computer domain and not to the user's task, the time spent on window management substantially increases the disruptive effect on short-term memory. Thus implies a non-linear cost curve as the number of windows per task increases (Shneiderman, 1992).
1.2 The Objectives of the Research

This thesis proposes a novel technique known as Vanishing Windows for reducing the window manipulation required by a user and to aid the user in search activities. The technique gradually reduces the screen real-estate requirements for inactive windows. The reduction of inactive window size progressively increases the overall visibility of windows on the desktop (less overlap). However, the visibility of individual window content will also reduce progressively.

The objective of the research is to develop the concept of Vanishing Windows and explore its potential for the intelligent management of screen real estate.

1.3 Research Methodology

The methodology used for the research is based upon an empirical approach. Ideas and techniques are developed and tested empirically. Feedback of the results are used to modify the system and further testing takes place.

1.4 Main Achievements

The Vanishing Windows system has been successfully developed and implemented. The final prototype system is the product of the results of the various experiments conducted as part of this thesis. Three major experiments were carried out:

The first experiment investigated the effect that presentation styles had on users' ability to correctly identify a document. Two presentation styles were considered for this experiment; Cropped vs. Scaling. The overall findings showed an improved performance in terms of correct identification using the cropped style compared with the scaling style. However, as the window size increased, the performance between the two systems began to converge. So, at large screen sizes, both presentation styles performed equally well.

From the results of the first experiment, the Vanishing Windows system was designed to use cropping views when reducing the size of inactive overlapping windows.
Investigations were carried out to derive techniques that could be used to further reduce window size and still retain a high recognition rate. A solution was to identify keywords contained in the document and to “pack” them into a small window size. Users needed only a few of these keywords to identify a window, which contained the target document. A technique for automatic extraction of keywords was developed. This technique is based upon frequency of words contained in the document. This frequency is compared with a frequency contained in a language corpus, which tabulates the frequency of words as used in normal written English Language.

Experiment 2 used this automatic extraction of keywords, and documents were modified to provide one of four presentation styles. The four presentation styles are compressed, highlighted, list and normal. Compressed is where all non-keywords from the document are removed and replaced with spaces, leaving only keywords. The spaces are compressed to occupy minimum space. Highlighted is where all keywords are highlighted, making it easier for users to scan the keywords. List is where all the keywords are listed in the format of a table and Normal is where no adaptation occurs. Like experiment 1, subjects were presented with all four adapted presentation styles and were asked to identify a target document at various sizes.

The results of experiment 2 showed that subjects performed better at identification using the compressed presentation followed by the highlighted presentation. However, when subjects were asked which adapted presentation they preferred, they indicated that the highlighted presentation was preferable because it retained the structure of the document. Both the compressed and highlighted presentation performed better than the normal presentation styles where no adaptation took place.

During the course of the experiment it was observed that the toolbars were of no value when windows were inactive. Therefore, the toolbars were removed
automatically for inactive windows, thereby releasing valuable screen real estate within a window.

A prototype Vanishing Windows system was designed and implemented taking the results of experiment 1 and 2 into consideration. The system used a cropped presentation style and adapted the document contained within the inactive window by highlighting keywords. Also, the toolbars were removed from inactive windows. When a window was reactivated, the document was re-established in its original form, with the toolbars displayed.

Inactive overlapping windows were made to vanish or reduce in size using a number of strategies. Inactive windows migrated, shrunk, shunted or evaded depending on the layout of the windows. The intelligent adaptation of windows was empirically evaluated in experiment 3. This experiment set a number of tasks to be completed using both the final Vanishing Windows System and a non-Vanishing Windows System (Microsoft's Windows 95). The result of the experiment showed that, where a high number of windows (>4) are visible on the desktop, subjects perform better using the Vanishing Windows system for a search task. User questionnaires showed that the obtrusive adaptation feature of the Vanishing Windows system did not distract users.
1.5 Guide to the Thesis

Figure 1-2: Thesis layout diagram

The thesis is divided into 8 chapters. The layout of these chapters in the thesis is illustrated in Figure 1-2. A brief description of each chapter is included here to summarise the thesis.
CHAPTER 1 - This chapter presents a brief background to the research. It examines the need for carrying out research in this subject area; detailing the research aims and objectives and the steps followed to realise these objectives. It outlines the main achievements of this research.

CHAPTER 2 - This chapter contains a literature review of the current work in the area of adaptive window managers.

CHAPTER 3 - This chapter reports on an empirical study carried out on users working on a particular windowing system. Some general observations are also made.

CHAPTER 4 - The chapter details the concept of Vanishing Windows and discusses the design of the system.

CHAPTER 5 - This chapter details the design of an experiment to investigate recognition performance of users using two presentation techniques (Scaled vs. Cropping). The investigation also reports on the effect that screen size has on user recognition rate. This is an important aspect since the Vanishing Windows system reduces window sizes of inactive windows. Therefore, the effect of window size on recognition is investigated.

In this experiment subjects were asked to read a document and become familiar with it. They were then presented with a one of four documents on the screen at various screen sizes and one of the two presentation styles. They were simply asked whether the document presented on screen was the document they were asked to read. The subject responded by selecting one of three options: “yes” this is the document; “no” this is not the document or “don’t know”. The result was then analysed to determine the effect of screen size and presentation styles on the user’s ability to recognition the target document.
CHAPTER 6 - This chapter further develops the techniques for reducing window size whilst retaining recognition accuracy. The use of keywords in the identification process was observed during the experiment detailed in chapter 5. A technique for automatic extraction of keywords from a document is developed and utilised. Four methods for presenting these keywords were developed and an experiment carried out to test the performance achieved by each method. The results of this experiment are also presented in this chapter.

The use of keywords to identify a document observed in chapter 5 was utilised to reduce the screen size further and to still retain high levels of accuracy. A technique for automatically identifying keywords was developed, whereby the frequency of a word appearing in the document is compared to the frequency of this word in normal written English. If the frequency was greater than that in the corpus then this word was taken to be a keyword (a word that can be used to identify the document).

Four techniques were developed to present these keywords and experiments carried out to determine which technique achieves greater recognition at low screen sizes. The four techniques were compressed (all keywords listed as they appeared in the document and replacing non-keywords by spaces); list (all keywords listed as a table), highlighted (keywords are highlighted so they stand out) or normal (no change made to the document).

Again, a similar experiment to that described in chapter 5 was used for the experiment. The results were analysed based on accuracy of identification and response times. This chapter details three related experiments. The first was to determine which one of the presentation style resulted in greater accuracy. The second experiment concentrated on the small screen sizes (i.e. window size < 30% of the screen size) and smaller increases in step size for the samples of windows used in the experiment. The third experiment contained multiple documents in the search
space. This was used to test the effect of multiple documents on the user’s ability to recognise the target the document.

CHAPTER 7 – A final prototype of the Vanishing Windows System incorporating the results of chapters 5 and 6 was built and empirically tested. Again, an experiment was designed to compare user performance for a given task on the Vanishing Windows System and a non-Vanishing Windows System (like Microsoft Windows). The results of the experiment are presented in this chapter.

CHAPTER 8 – Conclusions from this research are presented in this chapter. Areas for further research are identified and discussed.
Chapter 2

2 REVIEW OF WINDOWING SYSTEMS AND ADAPTIVE USER INTERFACES

This chapter reviews the current literature on windowing and adaptive systems. In particular, it examines techniques used in adaptive user interfaces and their applicability to adaptive window managers. Also addressed in this chapter, is a review of current adaptive window managers.

2.1 A Brief History of Window Systems

Multiple tiled windows were demonstrated in Engelbert's NLS in 1968 (Engelbart and English, 1968). Early research at Stanford on systems like COPILOT (Swinehart, 1974) and the EMACS text editor at MIT (Stallman, 1979) also demonstrated tiled Windows. Kay proposed the idea of overlapping windows in his 1969 University of Utah PhD thesis (Kay, 1969) and they first appeared in 1974 in his Smalltalk system (Goldberg and Robson, 1979) at Xerox PARC, and soon afterwards in the InterLisp system (Teitelman, 1977). SmallTalk's overlapping windows and direct manipulation interaction style made it very different from more typical systems of the time, which relied primarily on command-line textual interaction (Markoff, 1984).

Some of the first commercial uses of windows were on Lisp Machines Inc. (LMI) and Symbolics Lisp Machines (1979), that grew out of MIT AI Lab projects. The Cedar Window Manager from Xerox PARC was the first major tiled window manager (Swinehart et al., 1986), followed soon by the Andrew window manager (Morris et al., 1986; Palay, 1988) from Carnegie Mellon University's Information Technology Centre. The main commercial systems popularising windows were the Xerox Star (1981) (Smith et al., 1982), the Apple Lisa (1982), and most importantly, the Apple Macintosh (1984). Early versions of the Star and Microsoft Windows were tiled, but eventually they supported overlapping windows like the Lisa and the Macintosh. The
Chapter 2

X Window System, a current international standard, was developed at MIT in 1984 (Scheifler and Gettys, 1986).

2.2 Characterising Windowing Systems

Billingsley (1988), characterised windowing systems with three primary attributes. These attributes, when taken together, define the basic nature of the user interface for any windowing system. These attributes are:

*Interaction Style.* This attribute defines the ways in which users can input window management requests and how the system can respond to those requests. Typical interactions include direct manipulation, indirect manipulation, and combinations of these styles.

*Set of Operations.* Within the constraints of the chosen presentation and interaction styles, this attribute defines the particular set of windowing operations supported by the system, and describes the specific way in which those operations have been implemented.

*Presentation Style.* This attribute specifies the possible spatial relationships between windows and, to some degree, the types of operations that can be performed on them. Presentation styles include tiling, overlapping and mixed display.

2.2.1 Interaction Styles

One of primary attributes of a window system is the set of techniques provided for communicating window management requests to, and receiving feedback from, the system. These techniques can be categorised according to their basic ‘interaction style’; direct manipulation, indirect manipulation, or a mixture of these two styles.

Shneiderman (1982) first used the term ‘direct manipulation’ to describe a class of interactive systems characterised by:

- Continuous representation of the object of interest.
- Physical actions or labelled button presses instead of complex syntax.
• Rapid incremental reversible operations whose impact on the object of interest is immediately visible.

This definition implies that, in a direct manipulation environment, all the software structures that the users have the power to access or modify, will be displayed as objects. It also implies that each operation will have a visible component that reinforces the physical linkage between the operation and the object on which it is being performed.

In the context of direct manipulation systems, windows are objects that can be accessed and modified by users. The physical connection between a window, and the operations performed on it, are usually provided by a selection cursor. The cursor serves as an electronic equivalent to the human hand: it can point at, select, and manipulate objects on the display. Since the cursor, unlike the human hand, provides no tactile feedback to the user, the system may incorporate various types of visual and auditory cues to inform users that operations are proceeding as expected.

Using these techniques, it is possible to replicate some of the characteristics of real world actions on physical objects. For example, to change the position of a piece of paper on the desktop, it must first be grasped and then placed at the new location. In a direct manipulation interface, moving a window might be accomplished in the following parallel way:

• The user moves a mouse (or other cursor control device) to position the selection cursor in the interior of the required window, and presses a mouse button to initiate the “move” operation.

• The system verifies that the operation has been initiated by changing the shape of the cursor. For the duration of the operation, the window behaves as though it is attached to the cursor.

• The user moves the mouse to move the cursor. The cursor (attached to the window) is moved to a new location on the display.
When the user releases the mouse button, the system terminates the operation and displays the window at the new cursor position.

Some windowing operations have no analogy in the physical world of the paper and desktop. Pieces of paper, for example, are of a constant size, while windows can be instantly expanded to fill the screen, minimise to an icon, or made any size in between. Therefore, a number of decorations (or control icons) are added to the window and functions attached to them. Thus, to initiate windowing operations that would never be performed on paper in the physical world, users can directly manipulate the symbolic switches and buttons on the window border. In the Microsoft's Windows 95 interface, for example, control devices in the window border include the title bar (which must be selected to move the window), the size box; the maximise, minimise, and close box; and the horizontal and vertical scroll bars (see Figure 2-1).

![Figure 2-1: Microsoft's Win95 window, with direct manipulation controls embedded in its border](image)

Indirect Manipulation techniques are characterised by a reliance on language rather than graphics, and on typing rather than pointing. Examples include traditional menus, command lines and data input forms. These techniques identify the system
more as an obvious intermediary between the user and the windowing system; the user specifies a windowing operation, and the system executes it.

Some interaction techniques commonly found in windowing systems are interesting hybrids of direct and indirect manipulation. For example, in many systems, users must initiate a windowing operation by selecting it by name from a pull-down or pop-up menu. Typically, they use direct manipulation technique of seeing and pointing to access the menu and for selecting one of the windowing operations listed on it. At this point, the process takes on the characteristics of indirect manipulation: the system interprets the menu selection as a command and executes the desired operation.

2.2.2 Set of Operations

As windowing technology has matured, the set of window management operations provided to users has become relatively stable. Although these operations have been implemented very differently across windowing systems, most systems provide a predictable type and degree of control over window configuration. Window management operations fall into two general categories, corresponding to two sequential stages when windows are displayed. In the first category, windows are created and displayed and in the second category, currently displayed windows are manipulated. However, during the late eighties, the operation required to specifically create a window, define a window location, and to name or rename a window, were no longer required. These were all achieved by the open operation. This operation opens an application window and is now usually a single operation. That is, a single operation creates a window, starts the application and opens the required document. For example, double clicking on a word document called sales.doc would create a window, load the word application and open the sales document in a single operation.

Some of the early operations that were required for windowing system designed in the seventies and early eighties were:
For the second phase of display preparation in early windowing systems, users had to position displayed windows appropriately, and ensures that the information or objects required to perform the task were visible inside those windows. Operations that supported this fine-tuning of the window configuration included: move, resize, zoom, re-scale, scroll, name/rename, and make active (see Table 2-2).

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>Display an entirely new window</td>
</tr>
<tr>
<td>Delete</td>
<td>Remove a window from the display</td>
</tr>
<tr>
<td>Open</td>
<td>Replace an iconic window with the full size window it represents</td>
</tr>
<tr>
<td>Close</td>
<td>Replace a full size window with an iconic representation of itself</td>
</tr>
<tr>
<td>Bring to front</td>
<td>Move a window to the most forward plane of the display</td>
</tr>
<tr>
<td>Push to back</td>
<td>Move a window to the most rearward plane of the display</td>
</tr>
</tbody>
</table>

Table 2-1: Window operations for the creation phase of a window

The re-scale and scroll aspects of window manipulation are regarded as application activities rather than activities of the window manager. These activities or events are handed at the application level as opposed to the window manager level.

Over the last decade, advances in the interaction styles, in particular direct manipulation techniques, have enabled some windowing operations to be integrated into other functions, or removed from the activity of the user and transferred to the
system. For instance, it is no longer necessary for the user to manually name/rename windows. The window manager automatically assigns a window name (title) (usually the name of the application and/or the document). Similarly, it is no longer necessary to position the window when created manually. Instead, the window manager positions the window in a random position, or at the position where it was located at the last use of the window. Create and destroy operations are obsolete and have been integrated into the open and close operations. The open and close operations, as defined in Table 2-2, have been renamed and are more commonly known as minimise and maximise.

Although the lists above are comprehensive, they are not exhaustive. Other types of windowing operations, and variations on those presented here, have been implemented in some systems. The names used here have also been subject to considerable variation across systems. Overall, however, the listed operations represent the basic functional set.

2.2.3 Presentation Style

Billingsley (1988), suggests that for any windowing system, there are two alternative presentation styles, tiling and overlapping. Most systems use either style exclusively, but some incorporate both. For example, tiling is often used as the primary window presentation style, whilst overlapping is used for windows whose contexts are of only transitory interest such as menus, dialogue boxes and help displays.

2.2.3.1 Tiling Windowing Systems

In a tiling system, all windows lie on the same plane like a mosaic of tiles on a flat surface. Many different tiling strategies are used in different computer systems. These strategies are listed below:

One dimensional or single column system

Windows are linked up vertically in a single column. This gives users' control over only one-dimension of the window size, typically the height (Card et al., 1984). In such systems, as illustrated in Figure 2-2, each window is the full width of the
display, and multiple windows are arranged vertically. Most EMACS-like editors fall into this category. So does the virtual terminal manager of the RIG system (Lantz and Rashid, 1979).

![Typical window configuration in a one-dimensional tiling system](image)

**Figure 2-2: Typical window configuration in a one-dimensional tiling system**

Two dimensional or multi-column system
Most tiling system are two-dimensional or multi-column system like the example in Figure 2-3. They allow users to adjust both the height and width of each window. This model is used by Xerox PARC’s Ceder System (Teitelman, 1984) with a limit of two columns and by Microsoft Windows 3.1 (with a limit of 12 columns).

![Typical window configuration in a two-dimensional tiling system](image)

**Figure 2-3: Typical window configuration in a two-dimensional tiling system**

Hierarchical
The screen is divided either vertically or horizontally into partitions, and each partition is then recursively divided into sub-partitions. Windows correspond to the final set of undivided partitions. Each partition corresponds to a node in a hierarchy, with windows at the leaves (see Figure 2-4). The CMI/ITC Andrew System uses this window organisation.
Non-hierarchical

Any rectangular tiling can appear in this system (see Figure 2-5). Example of such a system is the Siemens Research and Technology Laboratories Rectangular Tiled Layout - built on top of PERQ systems Sapphire Window Systems (Myers, 1984).

2.2.3.2 Overlapping Windowing Systems

In an overlapping system, windows are handled like pieces of paper on a desktop; they can lie directly on top of the stationary background surface or on top of other windows, as shown in Figure 2-6. Because each window lies in its own plane, and planes are allowed to overlap, windows appear to exist in three-dimensional space. Users can typically control the height, width, and location of each window in its own plane, as well as the window's position relative to the foreground and background of the display.
Bly and Rosenberg (1986), define two user requirements for a multi-window system:

- The ability of windows to conform to their contents so as to maximise the visibility of those contents, and
- The ability of the system to relieve the user of having to manage the size and location of the windows.

Overlapping window systems maximise the first user requirement. In an overlapping system, the user has more control over the placement of the window, i.e. moving, resizing and overlapping windows anywhere on the screen. However, such systems do very little to aid the user in window management. Tiled window systems maximise the second user requirement. Because most tiled systems attempt to satisfy the conformance requirement by using all of the available screen space, windows often change size and location when other windows are opened or closed. Tiled systems typically do very little to conform to window contents, and as the number of open windows increases, the size of each window decreases.

Bly and Rosenberg (1986), also point out that systems that maximise neither requirement are unusable, and those that maximise both must be advantageous to a degree but unattainable by current techniques.

Both of these requirements can now partially be achieved using an overlapping system with some intelligent or adaptive window management features, which relieve the user of having to control the window management operation. For the system to maximise all the available screen space and still conform to the first
requirement i.e. maximise the visibility of windows content, a hybrid approach must be adopted. A hybrid system utilises the features of both tiling and overlapping window system.

2.2.3.3 Comparison of Tiled and Overlapping Windows
Authors over the years have speculated on the relative strengths of tiled and overlapping windows, but there has been little research done to support their speculations. Bly and Rosenberg (1986), carried out a study to explore the proposition that there are tasks and users for which tiled windows are superior to overlapping windows.

To test their proposition, the authors devised tasks that they predicted would be easier to perform using either a tiling system or an overlapping system. A task, which favoured a tiled window system to aid user performance, would require that the contents of the windows have maximum visibility within the default window location and size. That is, when these windows are tiled, or their default window location and size is used, they reveal the maximum content of a window. In this case a user would have no need for further window management operation and hence could proceed with the task.

Tasks favoured by the overlapping approach would involve windows that did not conform to a regular pattern (see Figure 2-7). In order to achieve maximum visibility of the contents, the user would have to arrange the windows in some irregular manner, exploiting the ability to manage and overlap the windows.
In tasks designed to favour tiling, almost all the information required to perform the task appeared in the upper left corner of each windowed document, and could thus be viewed when windows first appeared in their default sizes and locations. Completing the task required only one windowing operation: the user had to either move one window once, or scroll one window slightly. The task could also be performed using overlapping windows, but to take full advantage of the layout of information, users had to reposition and resize the default overlapping arrangement into one similar to that provided automatically by the tiling system. This required a larger, although unspecified, number of windowing operations.

In tasks designed to favour overlapping, the required information did not appear in the upper left corner of the document to be viewed, and was therefore not visible in the default arrangement of the tiling system. To complete the task using the tiling system, users had to perform several operations to open, close and scroll windows. However, because of default arrangement of the overlapping windows, users of overlapping windows also had to perform a series of window management operations to complete the task.

In the actual study, each of the 22 participants was tested individually in one of four conditions:

1. task optimised for tiling - using a tiling system
2. task optimised for tiling - using an overlapping system
3. task optimised for overlapping - using a tiling system
4. task optimised for overlapping - using an overlapping system.

The task completion data showed that, for the task optimised for tiling, participants who used the tiling system were significantly faster than those who used the overlapping system. This result is not surprising, since users of overlapping windows were required to perform many more window operations to complete the task.

The results were somewhat more complex for the task, which favoured the overlapping system. The performance distribution for the subjects using overlapping windows contained two groups or clusters. One group completed the task faster than the other group. The fast group took approximately 30% less time to complete the task than the group using the tiled system. This is not surprising, as the task was optimised for the overlapping system. However, the slower group took 30% more time to complete the task than subjects using the tiling system. Upon further examination of the subjects within the two groups, the authors (Bly and Rosenberg) found that, all but one of the subjects in the fast group were programmers and considered to be experienced in the overlapping system. In contrast, all but one of the subjects in the slow group were non-programmers and were generally less familiar with overlapping. Thus, it appears that the level of expertise distinguished the performance of the two groups.

The authors concluded that tasks, which require little window management, can be carried out more quickly using tiled windows, and tasks which require more manipulation can be carried out more quickly with overlapping windows. However, user experience needs to be taken into consideration to qualify this statement.

Bury et al. (1985) investigated the impact of windows on completion times for a series of information retrieval tasks. They found that subjects took a significantly longer amount of time to complete the task in a windowing environment than in a non-windowing environment. A detailed analysis of the results indicated that the time spent doing the task was less in the windowing environment, but that the time spent on window management operations increased the overall time. Experiments by Bly and Rosenberg (1986) have also suggested that a large part of the time it takes to solve
a problem using window systems is the time spent in manipulating the windows themselves.

In another study, Davies et al. (1985) found that, where tasks required supplementary information relative to a primary task, user performance was more error free in a windowing environment than in a non-windowing environment. However, subjects took significantly longer time to complete the tasks. Their study also suggested that the additional time spent was due to window management operations.

There are therefore trade-offs to be made between using a tiled approach and an overlapped approach when using window systems. These advantages and disadvantages are listed in Table 2-3 and Table 2-4.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Overlapped</th>
<th>Tiled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows can be efficiently sized to fit the displayed information</td>
<td>Fewer management operations for the user</td>
<td></td>
</tr>
<tr>
<td>Many windows can be displayed on one screen</td>
<td>Windows do not get lost because they are covered</td>
<td></td>
</tr>
<tr>
<td>Offers the user more control over the display</td>
<td>Window locations easily standardised</td>
<td></td>
</tr>
<tr>
<td>Supports high degree of multi-tasking on the part of the user</td>
<td>Uncluttered display appearance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>All windows fully visible</td>
<td>Easier for novices to use</td>
</tr>
</tbody>
</table>

*Table 2-3: Advantages for using tiled and overlapped windows*
Disadvantages

<table>
<thead>
<tr>
<th>Overlapped</th>
<th>Tiled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window management operations are more difficult</td>
<td>Size and number of windows is restricted</td>
</tr>
<tr>
<td>Display can become cluttered with windows</td>
<td>Difficult to add new windows without disrupting the display</td>
</tr>
<tr>
<td>Difficult to standardise window location</td>
<td>Cannot always size the window to optimally fit information</td>
</tr>
<tr>
<td>Some windows can become covered by other windows and forgotten</td>
<td>Restricts the degree of multi-tasking that can be achieved</td>
</tr>
</tbody>
</table>

Table 2-4: Disadvantages for using tiled and overlapped windows

Lane et al. (1997) conducted another experiment comparing tiled and overlapping windows. Multi-Window searches were performed using the two windowing styles and they were compared on the basis of search completion time and error rates. Results with 35 novice users showed faster completion times for the tiled windowing style than for the overlapping style. This quicker completion time may be due to the lack of overlap in the tiled interface. Subjects spent less time reorganising the screen and were able to complete the trials faster. By not allowing overlapping, subjects could quickly scan the screen to see if a window existed. With the overlap windowing environment, many subjects lost windows underneath the cluttered desktop. During the experiment, one subject spent minutes moving and searching windows to find the last window necessary to complete a trial. The completion times did reduce as the trials progressed, suggesting that some learning of the task was occurring.

Two future areas of work identified by Lane et al. (1997) are:

- Further experimentation is needed to determine to what extent the “Cluttered Desktop” increases the completion time.
Further research is needed to see whether independent variables such as number of windows opened, percentage of windows hidden, and frequency of subtasks, will affect user completion times and error rates.

There is clearly a need for more research in understanding the influence of presentation styles on user performance in windowing system. However, even without solid research results, the trend in the industry is away from tiling and towards overlapping systems, as exemplified by the decision of Microsoft to switch from tiling to overlapping for its future window based operating system (Angus and Mace, 1987).

2.3 Clutter

It is evident from the literature already reviewed that user performance is affected by "clutter". To get an understanding of clutter, a number of questions need to be addressed. These are as follows:

- What is 'clutter'?
- Why does clutter occur?
- How can clutter be managed?
- If clutter is reduced will that increase user performance?

2.3.1 What is Clutter?

Collins English Dictionary defines clutter as:

"(1) to strew objects about (a place) in a disorderly manner. (2) a disordered heap or mass of objects. (3) a state of disorder."

The dictionary definition places a strong emphasis on disorder. That is, there is no apparent existence of relationships within the organisation of the objects.

Funke et al. (1993) define clutter as:

"... the condition of highly dense and overlapping information"
This definition of clutter only takes into consideration proximity of objects and ignores other possible variables, such as the distinctness of objects. In fact, clutter can occur in either condition - high overlap or high density, and the co-presence of both increases clutter. In the overlap case, the user has to perform additional operations to retrieve the required information. In the highly dense case; actually locating the required information can be a problem even when there is no overlap.

Clutter, as defined by the dictionary definition, can also be viewed as describing an unordered set of objects. Take a room as an example. If objects in a room are not ordered or grouped in some logical manner then the room can be seen as being cluttered. Similarly, if items are piled onto a desk without any form of ordering then the desk might be seen as being cluttered. With a cluttered desk, the retrieval task becomes more difficult. However, it can be argued that although a person's desk is cluttered, they often know where a certain piece of information is located. This is because people have ways of ordering the clutter on the desktop (it is not totally random). They introduce spatial relations such as 'piles' of papers to induce relationships.

Another approach in defining and measuring clutter is to examine the difficulty in locating a piece of information or an object. If a space is cluttered then a user will have more difficulty in locating a piece of information or an object. So one measure of clutter might be the amount of work necessary to locate a piece of information or an object. This leads to the question of how work is defined.

The amount of work necessary to locate a piece of information can simply be defined (for a windowing system) as the number of windows operations required to locate a piece of information or identify a particular window. With respect to overlapping windows, if the windows are overlapping then the work required to locate a piece of information has two components. Firstly the amount of work necessary to unoverlap the windows and secondly the work required to locate the desired piece of information from the set of objects. This latter factor requires substantial cognitive
processing to compare what is required to what is available in the set of objects in the search space.

Clutter is also caused by the close proximity of similar objects. If dissimilar objects are close together, the user need not necessarily have a problem in distinguishing between them and selecting the appropriate object. This similarity and dissimilarity of objects when searching for objects is also supported by Treisman (1980; 1982). Triesman reports that if a target differs from the distracters in some simple property, such as orientation, colour or shape, the target is identified easily. Furthermore, the visual search for a target distinguished along a single stimulus (for example, colour or shape) is conducted in parallel, whereas the search for an item defined by the conjunction of two stimulus dimensions is conducted serially. In this respect a difference from Funke et al., (1993), is made as they do not consider similarity or dissimilarity and concentrate only on proximity.

If a user is working with four overlapping windows each containing text, then selecting the appropriate window might involve a considerable reading effort for each window. On the other hand, if the text in the four windows was about very dissimilar subjects, a cursory glance at each window might suffice. In addition, if the windows themselves were readily distinguishable in form as well, the task would be even easier. The text might have distinct layouts, or the windows might be showing different applications such as PowerPoint, Word and Excel, which are easily distinguishable by their gross features.

When a target search area has a number of sets of objects, we firstly have to find the appropriate set of objects of interest, and then find the required object within that chosen set.

Human beings use memory when searching for previously used information (Lansdale, 1991). Some simple generalisations about visual information processing are beginning to emerge, one of which is the distinction between two levels of processing (entire and focused) (Treisman, 1986). Some aspects of visual processing are
accomplished simultaneously (that is, for the entire visual field at once) and automatically (that is, without attention being focused on any one part of the visual field). Other aspects of visual processing depend more on focused attention and are performed serially. Treisman (Treisman, 1980; Treisman, 1982) has also reported that the visual search for a target distinguished along a single attribute (for example, colour, shape or size) is conducted in parallel. For such targets, the target 'stands out' or 'pops out' and the search time is independent of the number of irrelevant items in the set. In contrast, the search for an item defined by the conjunction of two attributes is conducted serially, and the search time increases, as the set becomes larger. Treisman (Treisman, 1980; Treisman, 1982) has concluded that the visual system is incapable of conducting parallel search over two conjunct attributes simultaneously.

Foster and Coles (1977) studied the typographic cueing of printed text. They found that typographic cueing (visual distinctions between different levels of text) yielded higher scores than non-cued information.

Mori and Hayashi (1993) conducted some experiments to determine the effect multi-window system has on users task performance. Their experiments indicated that peripheral windows did interfere with the users' main task activity.

It is reasonable to assume that as the number of items in the search space increases, so will the search time. In addition, as the clutter in the search space increases, we would expect to see an increase in the search time.

Springer (1987) conducted a number of experiments to assess user performance on retrieval of information from a search space. The screen format of the search space was varied in terms of the degree to which information fields were aligned in columns, the ordering of information on the screen and the amount of blank space on the screen (i.e. the density). User performance was measure by the target search time. The domain for the experiments was based on the screen format for directory assistance. Users had to identify the correct name and address from a database, for a
give request presented on an alphanumeric screen. Springer found that by lowering the density of the screen (by suppressing the appearance of redundant surnames) target identification times were reduced. However, she did not go further in identifying the reason for this.

The result can be explained in terms of the clutter concepts presented here. The Information Density has been reduced, and the search space has been broken down into sets of similar and dissimilar objects. By suppressing the appearance of redundant surnames - the target search area now became easier to process. The user is able to locate the surname first, scan all surnames until desired surname is located, and then search within this group for the desired address. If the surname does not match a particular group, the whole group of surnames can be skipped and search resumed at the next surname.

2.4 Complexity Theory

Another reason for manipulating the interface is to reduce the complexity of the interface. Complexity theory has been utilised by many researchers in interface design and many conclusions have been drawn. This section reviews the application of complexity theory to interface design and draws out principles that could be applied to window system.

Shannon and Weaver (1959) proposed a mathematical approach for measuring the amount of information produced by a communication process consisting of \( n \) classes of event, where an event is the transmission of a specific unit of information. In an English language communication, for example, we might consider the letters of the alphabet to be the communication units, in which case \( n=26 \) (slightly more if we include spaces and punctuation symbols). Shannon obtained a formula for \( H \), the measure of uncertainty in the occurrence of a specific event in a sequence of events:

\[
H = -K \sum_{i=1}^{n} P_i \log P_i
\]

*Equation 2-1: Measure of uncertainty*
Where:

\[ K = \text{a positive constant} \]
\[ n = \text{number of events} \]
\[ P_i = \text{probability of occurrence of the } i^{\text{th}} \text{ event} \]

Shannon pointed out that the form of \( H \) is identical to that of entropy in statistical mechanics, where entropy is a measure of disorder of a system that can be arranged in a large number of different ways. Similarly, clutter can be considered as being related to disorder. If a system is ordered in some way then the system is likely to be considered as less cluttered, hence the entropy of the system will be lower.

The meaning of \( H \) is best described by considering a system with 2 event classes (equivalent to a 2-letter alphabet or a 2-word language). If in such a system the probabilities of each class of event are \( p \) and \( q \), then putting \( K=1 \) for simplicity, the formula for \( H \) reduces to:

\[
H = -(p \log p + q \log q)
\]

Equation 2-2: Measure of uncertainty for a 2-letter alphabet

Where \( q = 1-p \)

For this relationship, if we plot \( H \) as a function of \( p \), we get an inverted ‘U’ shape. The peak of the inverted U represents maximum entropy and this is where the maximum uncertainty occurs (where all events occur with equal probability).

2.4.1 How Does Shannon’s Theory of Communication relates to GUI’s?

Weaver (Shannon and Weaver, 1959) expanded Shannon’s theory of communication and related it not only to speech but pictures and music as well. A GUI can be viewed as a communication system between the computer (CPU) and the user (Figure 2-8).
Weaver (Shannon and Weaver, 1959), defined three levels of communication problem: these being the technical, semantic and effectiveness problem.

*Technical Problems* relates to the accuracy of transmission from sender to the receiver.

*Semantic Problems* are concerned with the interpretation of meaning by the receiver.

*Effectiveness Problems* are concerned with the success with which the meaning conveyed to the receiver leads to the desired conduct on the receiver's part.

These problems, as outlined by Weaver, have been expanded to GUI problems by Comber and Maltby (1996) as shown in Table 2-5.

<table>
<thead>
<tr>
<th>Weaver</th>
<th>GUI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical</td>
<td>Accurate transmission of data</td>
</tr>
<tr>
<td>Semantic</td>
<td>Attachment of meaning to the data</td>
</tr>
<tr>
<td>Effective</td>
<td>Changes in the recipient by the data</td>
</tr>
</tbody>
</table>

Table 2-5: Levels of the communication process (from Comber and Maltby, 1996)

Comber and Maltby (1996) go further and says that:

"It may appear that the theory only applies at the technical level but closer thinking reveals that the problems at the technical level affect the semantics and effectiveness of communication. For example, a button with too small a font may not convey meaning and thus prevent the user from completing a task."

35
The fact that entropy can be interpreted statistically as a measure of the disorder of a system, has provided a justification for Bonsiepe (Bonsiepe, 1968) to use Shannon's formula to measure the order or complexity of the typographic design of printed pages.

He identifies two types of order, system order and distribution order. System order is determined by classifying objects according to common typographic widths and heights, and distribution order is determined by classifying objects by their distances from the top of the page and from the left side of the page.

Bonsiepe then used an adapted formula from Shannon's information theory to calculate the complexity 'C' of the printed page. This 'C' corresponds to Shannon's 'H', the measure of the uncertainty in the occurrence of an event. Bonsiepe's formula states that the complexity 'C' of a system is given by:

\[
C = -N \sum_{i=1}^{n} P_i \log_2 P_i
\]

Equation 2.3: Bonsiepe's Measure of Complexity

Where:
\[ P_i = \frac{n_i}{n} \]

And:
N = total number of objects (widths or heights, distances from top or side of page)
n = number of classes (number of unique widths, heights or positions)
n_i = number of objects in the i_th class
\[ p_i = \text{proportion of the objects in the } i_{th} \text{ class} \]

He tested the applicability of this formula by comparing two versions of a printed catalogue and found that the new version was 39% more ordered than the original version using his formula.

Tullis (1983) reviewed the literature dealing with computer generated, alphanumeric displays to understand how formatting affected the processing of information by the viewer. One metric he used was Bonsiepe's layout complexity.
Tullis was attempting to lower the entropy of the system (lowering the freedom of choice of the viewer). When Tullis applied Bonsiepe's technique to screens that had been identified in an earlier study (Tullis, 1981) as narrative and structured, he found that the structured screens returned a lower complexity value than the narrative screens. Users were tested on both the narrative and structured screens. He found that, after practice, the mean time to answer a question about the display was 8.3 s for the narrative format and 5.0 s for the structured format, and this was found to be statistically significant difference.

Tullis (Tullis, 1988) later decided to determine if the complexity measure was a useful usability metric. Again using alphanumeric data, he prepared 26 formats that were viewed by ten subjects in different trials. He found that the layout complexity did not help in predicting the time it took for a user to find information. This is an interesting result, as less uncertainty about the placement of objects should make it easier to find the information. However, he did find that it was an important predictor of users' ratings of the usability of screens.

In a second experiment using different displays and subjects, Tullis (Tullis, 1988) attempted to predict the subjective ratings. He found that, along with other measures, layout complexity helped to predict a user's rating of the usability of different screens.

Comber and Maltby (1996; 1997) performed a pilot experiment in which they construct four versions of a simple application (Launcher), which performs a similar function to that of Microsoft's Program Manager. In the four version of Launcher, the layout of the interface objects was varied and for each version, the complexity was calculated based on Shannon's measure of information flow.

Seven experienced computer users were tested on all four versions of the interface. The test involved a simple task: selecting a file, adding to a list, and changing its name. At the end of the experiment subject rated each of the interfaces. In addition, task completion time and error rates were recorded.
Their result showed that, for screens with the highest and lowest complexity, subjects took longer to complete the task. Again when subjects were asked to choose the most preferred screen, the screens with the highest and lowest complexity rated poorly, So it was not always the case that users preferred low complexity. No statistical significance of the results was presented and therefore this result should be treated with caution.

2.4.2 Complexity Theory – How Does it Apply to Window Clutter?

Kahn and Charnock (1995) identified two types of complexity for windowing systems, complexity within a window and complexity across windows. A better terminology for these two types of complexities would be Intra-Window Complexity and Inter-Windows Complexity. When objects are placed in the same window, the user will logically conclude that there are relationships among them. The more objects and types of objects there are, the more possible types of relationships could be inferred to exist, and the more difficult it might be for the user to make sense of it all. The metric used by Tullis (1983) measures this complexity within a window (Intra-Window Complexity), which included overall density, local density, grouping, and layout complexity as defined by information theory. Strategies that can reduce the complexity of a screen include reducing the number of objects, distributing the objects, and ordering the objects by task order or task/functionality groupings.

A metric for complexity across windows (Inter-Windows Complexity) is likely to include the following two components (taken from Bowers and Snyder, 1990):

- The number of windows that must be opened simultaneously in order for the user to not have to use short term memory to transfer information from one window to another.
- The total number of windows needed to complete the task.

However, these measures can be very difficult or impossible to compute in real-time for an adaptive windowing system, which does not have a detailed task model. Since without prior knowledge of what the task is, the number of windows required for
the task can not be calculated. In addition, the order in which a task is completed need not be fixed and may vary from user to user. With a task model some of these variables can be calculated to determine the complexity across windows.

2.4.3 Clutter Analysis Applied to Windowing Systems

2.4.3.1 Window Size Determination

The CUBRICON system (Funke et al., 1993) uses an algorithm to reduce clutter in a window displaying a map, by resizing the window to a minimum acceptable size to keep the overall clutter within the limits of a threshold value. This technique can be adapted to minimise screen clutter generated by too many windows being displayed at any one time.

The system displays information in either Maps or Tables. It used clutter analysis to determine minimum acceptable sizes for these windows. Table windows are sized by the system based on the number of columns they contain. Map size determination, however, is more complex. The minimum map size calculations are based on a heuristic approach to clutter analysis, which considers the number of displayed objects and the density of these objects within the display.

The clutter analysis algorithm is based on the premise that, if some sub region of the map is cluttered, then the entire map should be considered cluttered. That is, it is not sufficient simply to compute the percentage of the whole map that is occupied by iconic display objects. Rather, every sub region of the map (ideally) should be examined to determine whether it exceeds the clutter threshold (maximum allowable value of the clutter measure based on the density of displayed objects). Furthermore, the clutter threshold for sub-areas of a map display varies inversely with the size of the sub-areas.

Clutter should not be restricted to an analysis based on sub-areas of constant size. Consider the hypothetical display shown below. Quadrant 1 does not appear to be cluttered while quadrant 2 seems more cluttered. Yet, the equally sized-circled areas in these quadrants shown an uncluttered appearance when considered in isolation.
Conversely, the circle in quadrant 3 appears to be cluttered, while the overall quadrant is relatively uncluttered when taken as a whole. Quadrant 4 is definitely cluttered.

![Figure 2-9: Hypothetical displays (Funke et al., 1993):](image)

The CUBRICON system uses three sizes of circular area, called critical area sizes, each with a corresponding clutter threshold, to consider whether a map planned for presentation is too cluttered. If a planned map exceeds the clutter threshold for any of the three critical area sizes, then the size of the map window is increased to a size that will reduce the clutter to below threshold values for any of the critical area sizes.

2.4.3.2 How Clutter Analysis May Benefit Windowing Systems?

This sections expands some of the ideas developed by the CUBRICON system and develops ways in which similar techniques can be applied to windowing systems.

The CUBRICON system applies clutter analysis to a specific window, namely a window displaying geographic information (i.e. a map). However, the clutter analysis technique used by the CUBRICON system could be extended and applied to any windowing system. Based on the analyses of clutter, decisions can be made as what to do with the windows to reduce clutter. Windows may be moved, resized or minimised to bring the windows clutter rating within a threshold value.
Literature Review

The following variables need to be assessed in order to perform clutter analysis on the desktop window: number of windows, number of overlapping windows and the number of windows completely visible. The desktop window can be segmented and analysis performed on a segment by segment basis. Each segment can then possibly be sub-segmented in to circular areas for further analysis. These different analysis strategies need to be implemented and tested to find the most suitable strategy that can be adopted for the system.

After clutter analysis has been performed on the desktop window, the system can then act based on this analysis. For example, it can automatically bring the window clutter level to that of a defined threshold value. This can be achieved in a number of different ways. Firstly, overlapping windows can be moved away from each other to achieve minimum overlap. Secondly, we can start shrinking the window until the clutter level is below the threshold value at which time it ceases shrinking.

A consequence of such auto-adaptation is that continuous window movement may distract users. Furthermore, repositioning windows may disorientate users and they may lose the whereabouts of a particular window. During these activities, the user can be given some control over the process by setting various parameters (such as shrinking rate, time to start shrinking etc.) They can also be provided with statistics of window usage to provide them with information on how to select the best course of action on window movement.
2.5 Adaptive and Intelligent Window Managers

2.5.1 Adaptive Systems

There are many kinds of adaptive interfaces. Dieterich et al. (1993) suggest a basic classification of adaptive interfaces according to stages and agents. They define the following categories, based on the agent performing each of the stages of adaptation.

- **Self-Adaptation** - The presentation of information to the user is adapted in response to the behaviour of the individual user. The adaptation takes place continuously during the session. Meyer (1994) concluded that adaptation performed entirely by the system could be effective, primarily when the adaptation does not change the action set available to the user.

- **User Controlled Self-Adaptation** - Similar to Self-Adaptation but the final decision as to whether or not adaptation takes place is under the user's control. Adaptation only takes place when the user decides to adapt.

- **User Initiated Self-Adaptation**

- **Computer Aided Self-Adaptation** (Computer Aided Adaptation) focuses on enabling users to express their needs and preferences effectively. Kühme et al. (1992), carried out a survey of approaches to adaptivity and concluded that the most promising approaches are those which give the users more control over the adaptation process and extensively support them in adapting the interface themselves.

- **System Initiated Adaptation**

- **Adaptation** - the user manually adapts the system through customisation.

This categorisation is very useful and defines the degree of control the user has over the adaptation. This is plotted in Figure 2-11, which shows user customisation of an application as being a User Controlled Self-Adaptive System, whereby the user adapts the system manually through customisation features for particular needs.
Brown et al. (1990) makes a valid distinction between Adaptable and Adaptive systems:

**ADAPTABLE:** A system is considered as being adaptable if it provides the end user with tools that make it possible to change the system characteristics. The system has customization functionality enabling the user to customize to their particular need (Figure 2-11, point A).

**ADAPTIVE:** A system is considered adaptive if it offers the ability to change its own characteristics automatically, possibly after consulting the user, thereby adapting itself to the user's needs (Figure 2-11, point B).

Guidelines for adaptable and adaptive systems are beginning to emerge. Although work is required to validate these guidelines, they can be used to build a first generation of adaptive systems and serve as a further basis for development of the guidelines. Some initial, general guidelines offered by Thomas and Krogsæter (1993) for adaptive and adaptable systems are:

- For each adaptive feature there must also be a corresponding adaptable one because the users must be able to do at least everything the system can do.
- There should be several ways of accessing the adaptation environment otherwise these features will be of little use.
- The user should be in complete control of the system, the system only acting as assistant, otherwise the user will be forced into a specific working style.
- Suggestions from the system should not be dramatic and should not disturb users unnecessarily in their work.
- When possible, more than one adaptation possibility should be offered.
- There must be a way to undo adaptations on the user interface. Additionally, there should be a simple way of resetting all adaptations, so the interface will not be overloaded with adaptations, which are no longer needed or have no relevance to the task.

2.5.2 Window Managers
Using X System terminology (Myers, 1995), User Interface software consists of two major components: the window system (described in section 2.2) and the window manager. The window system supports the separation of the screen into several windows whilst the window manager allows the user to move windows around, resize them, minimise or maximise them. It is also responsible for displaying the title lines and borders around the windows. A window manager thus allows the user to organise the presentation of information on the screen to meet the user’s need.

Computers operating systems are usually multi-tasking these days, that is they allow the user to run a number of different jobs (sometimes called processes) at the same time. For example, a user might be compiling some code in one window whilst editing text in another.

In most conventional systems, if a user runs more than one processes, the input and output from the various processes can be confusingly intermixed on the screen. For example, the output from one process may appear while another process is listing a
file and be missed entirely, or the output might appear while running a word processor or graphics program and thereby clutter the screen.

When two or more processes request input at the same time, the user may give the input to the wrong program. A window manager solves these problems by simulating several terminals on the same physical screen. The window manager keeps the input and output for the different "virtual terminals" separate. A simple model of a window manager is shown in Figure 2-12.

![Figure 2-12: A simple model of a window manager](image)

2.5.3 Related Work

Researchers have acknowledged (Sandberg-Diment, 1984; Steinbrecher1984; Bury et al., 1985; Bly and Rosenberg 1986; Card and Henderson, 1987) the fact that current information visualisation techniques used in multiple windows systems result in time consuming window management operations by the user. Since these activities are not directly related to the users' task domain, time spent on window management can result in loss of the users' mental context and an increase in the actual task completion time. Designers have attempted to reduce the time spent on window management operations by developing adaptive or intelligent systems. These systems and techniques are reviewed in the next section.
The next section has been broadly categorised by the approach used to reduce window management operations. They are:

- Window Layout Constraint Based Approach - System uses logic or constraint to control the layout in order to reduce the window management operations.

- Multiple Window Co-ordination Based Approach - Reduce window management operations by means of automatic co-ordination of windows and rapid manipulation of windows.

- Miscellaneous Approach - all other approaches, such as transparent windows and 3D windows.

2.5.3.1 Window Layout Constraint Based Approach

RTL/CRTL (Cohen et al., 1986)

The RTL/CRTL (Siemens Research Technology Laboratories/Constrained Rectangular Tiled Layout) system uses constraints to control the layout of the windows. Constraints can be specified in several ways. A constraint language can be used by an application to specify a default window layout. Constraints can also be specified dynamically so that the layout can be adjusted as execution of the application proceeds. In addition the user implicitly asserts constraints when resizing or moving a window.

Constraint have priorities, and the system attempts to satisfy the constraints in priority order. Constraints that are asserted by a user action (such as explicitly moving or resizing a window) initially have very high priorities to ensure that the window remains as the user indicated.

Constraints specify:

- Which windows must be present on the screen
- Their size and location
- Adjacency and alignment requirements
- Limits on the ways in which the system may change the layout
- Particular organisations of the windows on the screen.

Two primary constraints are:

PRESENCE CONSTRAINT – In a tiled window system, not all windows may be able to fit on a screen, so some of the windows are not displayed and are iconised. A variety of presence constraints can be applied to help the system determine which windows should be displayed.

Two such presence constraints are \textit{Window Priorities} and \textit{Required Presence}.

\textit{Window Priorities} – The simplest form of presence constraint specifies the priority of a window. When a visible window is explicitly enlarged, or new windows are opened, there may no longer be space for all the remaining windows. The window with the lowest priority is usually iconised. Window priorities are determined by an ageing function. The window currently in use has the highest priority. When not in use, its priority is reduced by the ageing function.

Users can influence the operation of the ageing function, and define ‘use’. The ageing function will normally be linear, in which case the user specifies the rate down to some specific minimum value. ‘Use’ can either mean input, so that ageing should start just after a window has lost its input focus or it could also include output, so that the user may want the window to remain present as long as output is being directed.

\textit{Required Presence} – Often the presence of one window requires the presence of another. The RTL/CRTL system allows the specification of required window presence relationships.

LAYOUT CONSTRAINT – Window layout is controlled by a group of layout constraints that constrain the appearance of an individual window (Appearance constraints), or that establish geometric relationships amongst windows (Association constraints).
In the RTL/CRTL, system conflicting constraints are resolved by the assigned priorities. Constraints are satisfied in priority order. Any number of low priority constraints may remain unsatisfied to satisfy higher order priorities.

**CUBRICON System (Funke et al., 1993)**

The CUBRICON is a complex system utilising several knowledge bases for a specific application (air tactics control presentation system). However, there are concepts developed in the CUBRICON system that can be adapted to other applications, such as its intelligent window manager (CIWM).

The CUBRICON system uses several knowledge sources during processing. These knowledge sources included:

1. a lexicon,
2. a grammar defining the language defining used by the system for multimedia input and output,
3. a discourse model,
4. a user model,
5. a knowledge base for output planning strategies to govern the composition of multimedia response to the user,
6. a knowledge base of information about generally shared world knowledge, and
7. a knowledge base of information about the specific task domain of tactical air control.

These knowledge sources are used both for understanding input to the system and planning/generating output system.

Like the RTL/CRTL system, the CUBRICON system (Funke et al., 1993) applies constraints to achieve intelligent window management. The CUBRICON Intelligent Window Manager (CIWM) is a knowledge based system that automates windowing operations. It automatically performs window creation, sizing, placement, removal,
and organisation. These operations are accomplished without direct human inputs, although the system provides for user override of CIWM decisions.

Three types of dependency are identified for information presentation; the nature of the information (e.g. spatial, verbal, sequential), the task that the user is performing, and the function that information will serve within that task (e.g., comparison, correlation). Different types of visual presentation are provided which serve the communication requirements of a wide variety of applications include tables, maps, text, forms, and graphs. These presentation types define the window types available in the CUBRION system. The dependencies are used by CIWM to make informed decisions about the presentation layout.

CIWM combines tiled and overlapping layout approaches to form a hybrid window configuration management methodology. It uses a tiled windowing approach as default, but allows the tiled windows to overlap adjacent windows when necessary, based on window contents and the task at hand. As the number of windows increases, a limited number of the most useful windows are kept open, while those of lesser importance are removed and transformed to labelled icons.

One of the important issues in windowing systems is the size of windows. A window must be at least large enough so that the information shown is clearly legible. In the CUBRICON system, the size of a map window is determined by an algorithm that computes the minimum necessary size, based on an analysis of clutter (see earlier section on Window Size Determination). Using this algorithm, the size of a window may exceed the default size of the predefined window position, causing the window to overlap adjacent windows.

The CUBRICON system is a task dependent system and adaptation relies heavily on task domain data. However, two techniques used by the CUBRICON system - the Window Importance Algorithm (as discussed earlier) and the Window Size Determination algorithm could be used for more general task independent use.
CUBRICON Window Importance Algorithm

In information-based tasks, not all information is equally valuable to the user. Users predominantly focus on information critical to the task. Often they categorise and organise information to match priorities (Malone, 1983).

The CUBRICON system uses a composite parameter, based on several variables, to quantify the relevant importance of windows. This parameter, called window importance, is used to determine which window to place in the pre-defined window positions and which windows to remove from the screen and iconise, when necessary.

Since the CUBRICON system has a pre-defined window position for particular types of window, the window importance algorithm is used only when all the window positions have been occupied and a new window is to be displayed. The system compares the importance of the new window with the importance of the windows already displayed. If there are windows of less importance on the screen than the new window, then the system removes (and iconises) the least important window, putting the new window in its place. However, if the importance of the new window is less than the importance of any of the windows already placed on the screen, then the new window is not displayed and the system looks for a different means of communicating the information to the user.

Adaptive Window Manager (Stille et al., 1997)

Stille et al. (1997) have proposed an Adaptive Window Manager (AWM) based on their earlier work on Adaptive Automatic Display Layout (A^2DL) (Stille et al., 1996). The system automates the layout of the windows on the display screen according to the current user and task domain, and gradually learns the users' layout requirements. The A^2DL system observes a user's interaction with the computer system and maintains a profile for each specific user. A layout characteristic for each learnt context is associated with each user's profile. The layout features consist of the window/icon (object) attributes and layout parameters. Object attributes are size, width/length ratio, object position etc. and the Layout Parameter determines the
characteristics of the display layout like, a window always appears on top of another window, or the window is always in the top left corner of the screen. The layout features are inputs to the layout algorithms for generating the layout. Thus, when the user is in a same or similar working context that has been learnt by the A2DL system and has been incorporated in the profile for that user, the layout algorithms take into consideration the layout features for that context to present the layout.

The system uses a neural network to classify and determine the user's task domain or working context. The current working context is checked against a reference database of all currently stored contexts for that user. If no match is found then this context is stored for future reference. However, if a match is found, the current scenario stored in the database is used as a reference layout. Changes between the current and stored object attributes are calculated and the current layout is adapted to bring about the change in the display layout as per new object attributes to suit the current working context of the user.

This is a sophisticated approach, which learns layout preferences of users for different context and offers greater potential than CIWM where no learning was applied. However, no user evaluation of the system has been carried out.

2.5.3.2 Multiple Window Co-ordination Based Approach

Taxonomy of Multiple Window Co-ordination (North and Shneiderman, 1997)

Apart from managing windows in terms of their layout, another way of improving user performance, when working with windows, is to perform multiple window co-ordination. Consider the task of authoring a large web document. Scrolling a web page window might also produce a synchronised scroll of the corresponding HTML source view to show the same position of the document, providing simultaneous views of WYSIWYG (what you see is what you get) layout and a detailed layout description. Research indicates that potential gains from co-ordination strategies can be significant, dramatically improving user performance (browsing software, Shneiderman et al. (1986); the handling of large 2D spaces, Beard and Walker (1990); managing windows, Kandogan and Shneiderman (1997)). One reason for increased
performance is because in current windowing environments, windows are treated as independent and isolated objects, leaving users with the tedious job of individually manipulating one window at a time, even when windows are related by task or content. The coordination between windows avoids this and hence improves performance. North and Shneiderman (1997) have proposed the following categorisation of multiple window co-ordination.

A window consists of two components, a *view* and a collection of information *items*, each of which has an associated basic user action.

- **Selecting items**: Windows contain collections of information items of any of seven basic data types (e.g. characters, words, paragraphs of 1D text, or pixels, regions of 2D images, or nodes of a directory tree). In direct manipulation, the basic user action is selection.

- **Navigational views**: A window’s view provides a visualisation of the collection of items. Since the view might show only a small portion of large collections, users navigate through the view to see other parts. Different navigation actions apply to different data types (e.g. scroll, pan, zoom, slice, rotate, ascend/descend tree, follow link, open file, etc.).

Co-ordination on pairs of windows tightly couples an action in the first window to another in the second window. The primary dimension of the taxonomy classifies co-ordinations in three possible ways; Selecting Items to Selecting Items; Navigating Views to Navigating Views; and Selecting Items to Navigating Views. The second dimension classifies whether the collections of information items contained in the co-ordinated window pair are the same or different. For example, when a user selects (highlights or paints) an item (or set of items) in one view, and the system immediately highlights the equivalent item (or set), representing the same underlying data elements, in the other view, this is a “same” case. An example from Microsoft Word of a “different case” is user selection of an annotation in the annotation window, which highlights its corresponding pointer in the document text window to help the user locate it in the text.
One major implication of multiple window co-ordination is in the design of window managers/operating systems. Systems can be designed, in which end users can establish co-ordinations, on the fly, by interactively linking windows and specifying co-ordination types and details. North and Shneiderman go further and state:

"Such interaction across window borders may be the next big step in escaping the limitations of the current independent windows approach"

Elastic Windows (Kandogan and Shneiderman, 1996)

Kandogan and Shneiderman (1996) used a different approach to reduce window manipulation. Rather than developing an adaptive or intelligent system, they added features to the windows system so that multiple windowing operation can be applied rapidly. They proposed a system called “Elastic Windows” to address the following requirements:

- Support users to promote organisation and co-ordination of windows according to task
- Allow fast task switching and resumption
- Free users cognitive resources to work on task related operations rather than on window management operations.
- Use screen space efficiently and productively for the tasks
- Provide a spatial layout that indicates the relationship between windows.

Their system was based on three principles: hierarchical window organisation, multi-window operations and space filling tiled layout.

Hierarchical window organisation supports users structuring their work environment according to tasks. The hierarchical organisation of windows allows users to map their task hierarchy onto a nested rectangle tree structure.

Multiple window operations in the elastic window system can be achieved in two ways:
• By applying the operation to a group of windows at any level of the hierarchy. The results of the operations are propagated to windows inside that group recursively. This way a group of windows can be packed, resized, or closed with a single operation.

• Select a window operation and apply it to windows rapidly in a serial manner.

The space filling tiled approach is employed to make maximum use of screen space, avoiding the wasted background of the overlapped windows approach. Groups of windows stretch like an elastic material as they are being resized, and other windows shrink to make space.

Due to the space-filling tiled nature of the layout, window size changes affect the size of other windows. Dragging a border in a space-filling tiled layout causes some windows to be either pushed or pulled. As a window is pushed or pulled other windows stretch or squash proportionally according to their window size. However, each window has a default minimum size, but users can set a different value for each window. This way a user can protect a window from unwanted size updates.

Elastic Windows support a number of standard windows operations which can be applied to a individual window or a group of windows. The standard operations are resize, open/close, and maximum. Additionally it supports what they refer to as “multiple pack” and “unpack” operations. Packed windows appear in the same location, but with only their title shown as a bar appropriately placed in the layout. This avoids the spatial disorientation which is typically the case in the iconify operation in most of the windowing system. The pack/unpack operation is very similar to a function provided on the MacOS 7.0 and higher operating systems. Users can double click on the title bar of any window causing the window to scroll up and only display its title bar. So spatial orientation is preserved. However, users have to perform this operation individually to each window and no group operation facility is provided.
The placement of the packed windows in the same position of the layout keeps the same spatial cues formed by the user when these windows are all open. This helps users to locate these windows easily. The Pack/Unpack operations are primarily used to abandon a task for a while and open space for other tasks.

One unique feature of the elastic windows system is its ability to resize a window in all directions simultaneously by the pump operation. Pumping a window resizes the window pushing all the surrounding windows to the sides. This operation is invoked by pressing either the left or the right button of the mouse on the pump gadget. Pressing the left (right) button causes window size to enlarge (reduce) in all directions according to the duration of press.

Kandogan and Shneiderman (1997) proposed an evaluation metric for user performance measurement to analyse window management strategies. Their performance measurement is based on environment set-up tasks, environment switching tasks and task execution times.

*Environment Set-up* is the act of accessing information objects needed for the task, opening windows for them, and arranging the layout. Programmers opening source code modules in multiple windows and to arranging them on the screen, would be an example.

*Environment Switching* is the act of changing the screen contents to an existing environment set-up. An example would be to switch to reading specifications in the middle of programming.

*Task execution* during this phase, actions are performed on or with information contained in windows. Sequentially looking through many job descriptions to find the best paying job would be an example of this phase.

However this can have a more general meaning and can identify the unique phases that a user has to complete to accomplish a given goal. They can therefore be referred
to as the environment set-up phase, environment switching phase and the task execution phase.

They identified four execution types: *Sequential scanning, Comparison, Determine context + scan,* and *Recall context + scan.*

- **Sequential Scanning** - looking sequentially through a number of information sources for certain attribute of the information, such as job salary.

- **Comparison** - comparing a number of information sources based on one or more attributes, such as job description and benefits. It is different from sequential scanning because the users have to "glance" back and forth multiple times until they comprehend the distinction well enough to make a judgement.

- **Determine Context + Scan** - a filtering approach based on an attribute to establish a context for further scanning. For example, once a decision is made to seek jobs in California, this context enables users to limit scanning to jobs in California.

- **Recall Context + Scan** - The context is not determined but rather recalled based on previous interactions with the information source.

This evaluation metric was used to compare user performance with the Elastic Window system and traditional window management techniques for 2 (Low), 6 (Medium), and 12 (High) window situations. The results showed that task environment set-up times, for medium and high complexities for the Elastic Windows, were lower than those of the independent overlapping windowing system. But a low complexity task showed no statistically significant differences. For task environment switching, all results supported the Elastic Windows Interface. The differences were statistically significant except for low-to-medium and low-to-high environment switching. In addition, task execution times for all complexities were
The experiment used a variably transparent tool palette superimposed over different background content: text, wire-frame images, and solid images. The palette contained text icons, line art icons and solid, rendered object icons. Both the effects of varying the transparency levels and the interference produced by different types of context information were studied.

The experiment presented a palate of icons and the subject was asked to identify a particular icon. There were three independent variables: the type of palette icon (text, line art, and solid), the type of background (text pages, wire frame images, and solid images), and the transparency level. The dependent variables were; selection response time and errors. Two error condition were possible: The subject pressed the "can't see" button indicating that the item was not legible, or the subject selected the incorrect palette item.

The results showed that icon type, background type and transparency all affected response time performance. They found that highly transparent (90%) interfaces were unusable. In most cases transparency levels of 0% to 50% (opacity) seem to work about equally well, independent of icon type or background. Wire frame backgrounds were found to perform the worst. Subjects commented that the wire frame images seemed very visually complex which resulted in maximum interference.

These results can be used to design an adaptive window manager whereby windows become transparent when overlapped. The results seem to indicate that when solid images are used, the identification performance is better, implying that windows should remain solid. However, further research is required to determine the affect of depth of transparency. For example, if many other windows overlap a window, to what depth should transparency be used?

3D Window Manager (Leach et al., 1997)

An interesting solution to the screen real-estate problem is a 3D Window Manager (Leach et al., 1997). Their system MaW³ - A 3D Window Manager uses a 3D space in which windows are arranged in a tunnel. The user is positioned in the middle of the
statistically significantly shorter for Elastic Windows, although there were some exceptions for the low complexity situation.

2.5.3.3 Miscellaneous Approaches

Rooms (Henderson and Card, 1986)

Henderson and Card’s system allocates a virtual workspace to an abstract object called a Room and links such rooms by interconnecting doors. The doors can be used to access other applications.

The Rooms system attempts to reduce space contention on the screen by distributing the user’s windows into window locality sets (grouping windows together) in virtual workspaces. But this very fragmentation of the spaces creates a navigation problem: how can users keep track of the windows no longer visible and find their way through the Rooms?

A Room may still contain a number of windows and unless the user limits the number of windows contained within a Room, the problem of cluttered desktop will still exist but will be contained within a Room.

The overall issue of navigation contains several sub-issues. How do users: (1) return to a Room, (2) find other Rooms, (3) find connecting Rooms and (4) find Windows.

Some navigational issues were addressed by the Rooms system. Doors interconnect Rooms. A back door is created when a user enters a room. This back door connects to the previous Room. Pop-up menus for Room names are used to alleviated the problem of disorientation and to help users to remember (or discover) the route to particular workspaces. In addition an overview of the set of Rooms is also provided to help users from being disorientated. No usability testing was carried out.

Transparent Interfaces (Harrison et al., 1995)

Harrison et al. (1995) conducted an extensive evaluation of “transparent user interface tools”. Their experiment explored the issues of focused attention and interference, by varying both visual distinctiveness and levels of transparency.
mouth of the tunnel looking toward the other end. The tunnel, and windows in it, is displayed with a perspective projection. Windows are essentially 2D; that is, their work area is 2D. However, they have 3D frames, decoration and buttons.

Windows may be positioned at arbitrary depths in the tunnel. As the user pushes a window further into the tunnel its size is diminished in the normal inverse-size-to-distance relationship of perspective projection.

In addition to the front-on window display mode there is a hanging mode where the windows are hung on the left or right walls of the tunnel. Additional functionality is available to hang individual windows, or all windows may be hung at once. Hanging all windows allows the user to quickly gain an overall idea of where the windows are in the tunnel.

Two additional features have been implemented. The first is an overview area to the right of the tunnel. The overview area provides a plan or top view of the tunnel. This is in addition to the main “down the tunnel” view. Windows may be selected and moved up and down in the overview area, corresponding to back and forward in the tunnel. Window names are displayed in the overview area to aid identification of specific windows. However, the overview area can easily become crowded as the number of windows is increased. Identifying individual window can become very difficult just from the name alone.

The second feature is a console, positioned at the bottom of the screen. Controls are provided on the console for changing global settings affecting windows. These are:

- Button to hang all windows at once rather than individually
- Button to toggle transparency – windows can be rendered partially see-through, so that windows may be seen through other windows.
- Options to set different rendering modes – solid, wire frame and invisible for the walls to allow user to customise their appearance.
• Mechanisms to control lighting, which provides enhanced visual cues about the position of windows in the tunnel.

The 3D Window manager was designed to address the issue of “window thrashing” (Henderson and Card, 1986). The 3D-window manager addresses it in four main ways:

• Through window hanging. Hanging some or all of the windows allows the user to obtain a global view of the window locations. It also allows better allocation of window real estate.

• Through the scaling of window size in inverse proportion to distance down the tunnel. Windows which are not in use may be pushed back down the tunnel where they will be small but visible.

• By reducing mouse movement required to access windows.

• With transparency. If windows are made transparent, obscured windows can be seen (although selection becomes difficult).

However, extensive evaluation needs to be carried out on the 3D-window manager to quantify whether these approaches solve the window-thrashing problem. The problem is not as simple as reducing mouse movement but more about being able to identify required windows without the need for too many searches and switches between windows.

2.6 The Need to Manipulate Windows

In multiple windowing systems, like overlapping windows, users need to perform a number of windowing operations to accomplish a given task. Kandogan and Shniederman (1997) identified two distinct sequential phases that user's follow to achieve a task. The two phases are environment set-up and task switching phase (see earlier section). It is during these phases that the user manipulates the windowing environment to perform their tasks.
Some reasons for window manipulation are:

- To alleviate the complexity of the task. It may be easier to complete a task when windows are laid out in a certain fashion. Generally, the user has a 'mental model' of what is required to complete the task. The term 'mental model' is used to convey the idea that users have in their mind a structure with elements and relationships that map to the elements and relationships at the interface level (Norman et al., 1986).
- To see more information contained in a window.
- In an overlapping windowing system, to locate relevant information located in one of the windows.

A number of visual searches of the windows occur during task environment set-up and in a task-switching phase.

2.7 Evaluation Methods

A review of evaluation techniques is presented in this section, identifying advantages and disadvantages of each of the methods presented.

Evaluation is an important activity in the design cycle. In recent years, a number of practical methods have been developed, which can be applied to interface evaluation. The term "tools", as in "software evaluation tools", or "usability assessment tools", has often been used to describe such methods. Each of the different approaches to evaluation has its own strengths and weaknesses. They also differ in their cost and in the technical competence required to carry them out.

Evaluation can be either qualitative (how successful did you think the interface was?) or quantitative (what was the percentage speed-up in time to complete the task). Quantitative data deal with either user performance or attitudes that can be recorded in a numerical form. Qualitative data focus on reports and opinions that may be categorised in some way but are not reduced to numerical values. Often, the result of the evaluation of an interface is a set of design improvements.
Chapter 2

There are two fundamental issues in evaluation: Firstly, how do we decide what dimensions to evaluate and secondly, how do we put a value upon the dimension we evaluate.

2.7.1 Common Approaches to Evaluation

There are five common approaches to evaluation (Preece J., 1993; Lansdale and Ormerod, 1994), which we will examine, referring to their main features, advantages and disadvantages.

- **Analytic evaluation** uses formal or semi-formal interface descriptions to predict user performance.
- **Expert evaluation** involves experts in assessing an interface.
- **Observational evaluation** involves observing or monitoring users' behaviour while they are using an interface.
- **Survey evaluation** seeks to elicit users' subject opinions of an interface.
- **Experimental evaluation** uses scientific experimental practice to test hypotheses about the use of an interface.

2.7.1.1 Analytic Evaluation (Formal modelling as an approach to evaluation)

Formal models attempt to define the interactive properties of the interface and cognitive skills required to use them.

Formal modeling methods fall into three overlapping categories:

- They are exploratory tools in research. The attempt to develop formal models of interface performance reveals interesting insights and develops methodology further. To some extent, all the models described below fall into this category.
- They represent theories of cognition and interfaces. At present, it seems optimistic to think that the models developed will qualify as scientific theories.
• They are useful disciplines for the evaluation of interfaces.

A number of approaches to formal modelling have arisen in recent years. These are:

GOMS (Goals, Operators, Methods and Selection rules; Card et al., 1983)

TAG (Task-Action Grammars; Payne and Green, 1986)

CCT (Cognitive Complexity Theory; Kieras and Polson, 1985; Polson 1987)

PUM (Programmable User Models; Young et al., 1989)

Such models have been associated with the following advantages (Olson and Olson, 1990):

• They can limit designs to psychologically plausible alternatives. For example, they can avoid overloading working memory.

• They can be used in deciding between two alternative designs for an interface.

• They can estimate performance times for given tasks.

• They can contribute to the development of training regimes and inform designers as to the most efficient methods.

• They can indicate hot spots of interface use where errors or delays can be expected, hence directing attention towards significant performance issues.

2.7.1.2 Expert Evaluation (Walkthrough approaches)

While informal demonstrations to colleagues or customers can provide useful feedback, more formal expert reviews have proved to be effective (Nielsen and Mack, 1994). Walkthrough approaches are structured methods, whose purpose is to allow evaluators to evaluate interfaces by stepping through their use without the need for formal modelling or usability testing. The expert works through the design for a particular task, step by step, identifying potential problems. For each task that the design is intended to support, the expert considers the following issues:
• What impact will the interaction have upon the user?
• What cognitive processes are required?
• What learning problems may occur?

Expert Evaluation that is guided by general 'rules of thumb' is known as heuristic evaluation (Nielsen and Molich, 1990). Heuristic evaluation consists of the application, by evaluators of varying degrees of expertise, of a set of heuristics to judge the adequacy of a design prototype. Its purpose is to be a stripped down rule base, which is simple to apply. The expert reviewers critique an interface to determine conformance with a short list of design heuristics such as the eight rules of interface design (Shneiderman, 1992)

The dangers with expert reviews are that the experts may not have an adequate understanding of the task domain or user communities.

2.7.1.3 Observational Evaluation

A popular way to gather information about actual use of a system is to observe users interacting with it. Usually a specific task is set by the evaluator and completed by the user, although, if the observation is carried out in the users’ place of work, they may be observed going about their normal duties.

The evaluator watches and records the user's actions using a variety of techniques (see below). Simple observation is not sufficient to determine how well the system meets the user's requirements since it does not always give an insight into the user's decision processes or attitude. Therefore, users are asked to 'think aloud' (describe what they believe is happening, why they take an action and what they are trying to do) to elaborate their actions.

The 'think aloud' process has a number of advantages:

• The process is less constrained and therefore easier to learn to use by the evaluator
• The user is encouraged to criticise the system
• The evaluator can clarify points of confusion at the time they occur and so maximise the effectiveness of the approach for identifying problem areas.

The usefulness of the think aloud and general observation is largely dependent on the effectiveness of the recording method and subsequent analysis. There are a number of methods for recording user actions. These include:

Paper and pencil – This is a primitive method but allows the evaluator to note interpretations and extraneous events as they occur. However, it is hard to get detailed information, as it is limited to the evaluators writing speed. Coding schemes can be developed during preliminary studies to improve the rate of recording substantially.

Audio Recording – This is useful if the user is actively ‘thinking aloud’. However it maybe difficult to correlated information from the audio recording to some other form of protocol such as a hand written script.

Video Recording – This has the advantage that we can see what the subject is doing.

Computer Logging – It is relatively easy to get a system automatically to record user actions at a keystroke level. Software such as Lotus ScreenCam enables Windows screens to be recorded and played back. Direct logging has the advantage of being unobtrusive. Technical problems with it are the sheer volume of the data can become unmanageable and difficult to analyse.

2.7.1.4 Survey Evaluation
The purpose of survey methods is to address users' subjective opinions through the use of either interviews or questionnaires. The advantage of such methods is that they get the user's viewpoint directly and may reveal issues, which have not been considered by the designer. In addition they are relatively simple and easy to administer. However, the information gained is necessarily subjective, and may be a
rationalised account of events rather than a wholly accurate one. Also it may be
difficult to get accurate feedback about alternative designs if the user has not
experienced them, which limits the scope of the information that can be obtained.
However, the methods provide useful supplementary material to other methods.

**Interviews**

Interviewing users about their experience with an interactive system provides a direct
and structured way of gathering information. Interviews need careful planning so
that the line of questioning followed is relevant to the interface being evaluated.
Usually, some form of plan is made before the interview: either the sequence of
general topics to be covered is determined or some form of checklist of topics or
questions are prepared. General questions will usually be asked first about a task and
progress to more leading questions (often in the form ‘why...?’ or ‘what if...?’) to
elaborate aspects of the user’s response.

**Questionnaires**

An alternative method to querying the user is to administer a questionnaire. This is
clearly less flexible than the interview technique, since the questions are fixed in
advance, and it is likely that the questions will be less probing. However, it can be
used to reach a wider subject group, it takes less time to administer, and can be
analysed more rigorously. It can also be administered at various points in the design
process, including during requirements capture, task analysis and evaluation, in order
to get information on the user’s needs, preferences and experience.

Given that the evaluator is not likely to be directly involved in the completion of the
questionnaire, it is vital that it is well designed. The first thing the evaluator must
establish is the purpose of the questionnaire: what information is required? It is also
useful to decide at this stage how the questionnaire responses are to be analysed. For
example, is measurable feedback on particular interface features required, or user’s
impression of using the interface?
There are a number of styles of question, which can be included in the questionnaire. These include:

*General* – These are questions which help to establish the background of the user. They include questions about age, sex, occupation, and so on. They may also include questions on previous experience with computers.

*Open-Ended* – These ask the user to provide unprompted opinion on a question, for example, ‘Can you suggest any addition features that may improve the interface’. They are useful gathering general subjective information but are difficult to analyse in any rigorous way, or to compare. However, they may identify errors or make suggestions that have been missed by the designer.

*Scalar* – These ask the user to judge a specific statement on a numeric scale, usually corresponding to a measure of agreement or disagreement with the statement. For example,

> Highlighting keywords on inactive windows makes a document easier to identify.

*Disagree 1 2 3 4 5 Agree*

The evaluator chooses the granularity of the scale. Coarse scale (say 1 to 3) gives a clear indication of the meaning of the numbers (disagree, neutral and agree). However, it gives no room for varying levels of agreement, and users may therefore be tempted to give neutral responses to statements that they do not feel strongly about but with which they mildly agree or disagree. However, a very fine scale (say 1 to 10) suffers from the opposite problem: too many choices are offered to the user and these choices can be difficult to interpret. One user will undoubtedly interpret the scale differently from another. Scales of 1 to 5 or 1 to 7 have been used effectively (Dix *et. al.*, 1993 pp: 392). They are fine enough to allow users to differentiate adequately but still retain clarity in meaning.

*Multi-Choice* – Here the respondent is offered a choice of responses, and is asked to select only one of these, or as many as apply.
Have you used any version of Microsoft Windows and for how long?

☐ No, have not used Microsoft Windows at all
☐ Less than a year
☐ Between 1 and 3 years
☐ More than 3 years

Which types of Microsoft software have you used (tick all that applies)?

☐ Microsoft Word
☐ Microsoft Excel
☐ Microsoft PowerPoint
☐ Microsoft Access

These are useful in gathering information on the user's previous background.

*Ranked* – These place an ordering on items in a list and are useful to indicate a user's preferences. For example,

*Please rank the presentation styles in order that you found easiest to identify the target document*  
(1 easiest, 2 next, 3...)

☐ Compressed
☐ Highlighted
☐ List
☐ Normal

A questionnaire may consist of any number of these question types. Different question types are useful for different purposes as discussed above. However, in order to reduce the burden of the effort on the respondent, it is best to use closed questions such as scalar, ranked or multi-choice, as much as possible.
2.7.1.5 Experimental Evaluation

One of the most powerful methods of evaluating a design or an aspect of a design is to use a control experiment. This provides empirical evidence to support a particular claim or hypothesis. An important feature of the experimental approach is that the evaluator can manipulate a number of factors associated with interface design and study their effects on various aspects of user performance (see section on usability below for a list of measures of performance).

The planning of the experiment requires the specification of three main elements:

- The purpose of the evaluation as to be expressed in terms of what is being varied, what is being kept constant, and what is being measured.
- The hypotheses have to be stated in a way that can be tested.
- Statistical tests have to be selected to check the reliability of the results.

Two additional factors, which must be considered carefully in experimental design, are the subject chosen, and the variables tested and manipulated.

Subjects

In evaluation experiments subjects should be chosen to match the expected user population as closely as possible. Ideally, actual intended users of the system will be used as subjects for the experiment but this is not always possible. If subjects are not the actual users they should be chosen to be of a similar age and level of education as the intended user group. Their experience with computers in general, and with systems related to that being tested, should be similar as should their experience or knowledge of the task domain.

Another issue relating to the subject is the sample size chosen. This is often determined by external factors such as the availability of subjects or resources. However the sample size must be large enough to be considered to be representative of the population taking into account the design of the experiment and statistical
method chosen for the analysis. Dix et al. (1993) suggest as a rough guide a sample size of at least ten subjects for controlled experiments.

Variables
There are two types of variable, those that are measured and those that are manipulated. The former is known as dependent variables, the latter known as the independent variables.

The dependent variables are those that are measured in the experiment. The dependent variable must be measurable in some way, it must be affected by the independent variable, and as far as possible, unaffected by other factors.

The independent variables are those characteristics of the experiment, which are manipulated to produce different conditions for comparison.

2.7.2 Selecting an Evaluation Method
Preece (1993) identified three categories by which the differences among the five evaluation methods can be summarised: the stage of interface development for which they are suitable, the extent and type of user involvement, and the production of either qualitative or quantitative data.

<table>
<thead>
<tr>
<th>Method</th>
<th>Interface Development</th>
<th>User Involvement</th>
<th>Data and Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytic</td>
<td>Specification</td>
<td>No users, task specified</td>
<td>Quantitative</td>
</tr>
<tr>
<td>Expert</td>
<td>Specification or prototype</td>
<td>Role playing, no task restrictions</td>
<td>Qualitative</td>
</tr>
<tr>
<td>Observational</td>
<td>Simulation or prototype</td>
<td>Real users, no task restrictions</td>
<td>Qualitative / quantitative</td>
</tr>
<tr>
<td>Survey</td>
<td>Simulation or prototype</td>
<td>Real users, no task restrictions</td>
<td>Qualitative / quantitative</td>
</tr>
<tr>
<td>Experimental</td>
<td>Normally full prototype</td>
<td>Real users</td>
<td>Qualitative / quantitative</td>
</tr>
</tbody>
</table>

Table 2-6: Differences among the five evaluation methods
(Preece, 1993)

Advantages and disadvantages offered by Preece (1993) and Jeffries et al. (1992) have been consolidated and summarised in Table 2-7.
### Chapter 2

#### Method

<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytic</td>
<td>Usable early in the design and few resources required.</td>
<td>Extremely complex and requires expertise. Broad assumptions of users' cognitive operations. Limited guidance on how to use methods therefore can be difficult for evaluator.</td>
</tr>
<tr>
<td>Expert</td>
<td>Less complex than analytic method. Strongly diagnostic. Overview of whole interface.</td>
<td>Cannot capture real user behaviour. Problems locating experts. Subjective and can miss important problems</td>
</tr>
<tr>
<td>Observational</td>
<td>Quickly highlights difficulties. Verbal protocols valuable source of information. Can be used for rapid iterative development. Rich qualitative data.</td>
<td>Observation can affect user activity and performance levels. Analysis of data can be time consuming.</td>
</tr>
<tr>
<td>Survey</td>
<td>Address users' opinions and understanding of the interface. Can be made to be diagnostic. Can be applied to users and designers. Questions can be tailored to the individual. Ratings scales lead to quantitative results. Can be used on a large group of users.</td>
<td>User experience important. Low response rate (especially to mailed questionnaires). Possible interviewer or recipient bias. Analysis can be complicated and lengthy. Interviews very time consuming.</td>
</tr>
</tbody>
</table>

**Table 2.7: Advantages and disadvantages of the evaluation methods. Preece (1993) and Jeffries et al. (1992)**

#### 2.7.3 Usability testing

Usability testing is based upon the principle of trialing prototypes, capturing data about the system in use and the user performance, which can then be analysed. The evaluator must decide what needs to be evaluated and its relevance to the usability aspect being analysed. For the outcome of these trials to have validity, it is important that the users tested are representative of the intended user population and that the tasks they are required to carry out are realistic. This can be difficult if the intended user population is small and specialised, or if the tasks concerned, are hard to simulate, as in hazardous situations.
Usability can also be decomposed into the following components (after Shackel, 1986).

Effectiveness. This is a measure of how well the interface, and hence, the system, performs in achieving what the user wants to do. This can be measured in terms of:

- Error rates lower than a target level.
- Task completion time within a set time.
- Usage of system facilities above a minimum target frequency.

For example, a typical usability goal for effectiveness may be:

*95% of the users complete the set tasks within 10 minutes with an error rate of less than 2% of the transactions processed.*

Learnability. This measures how easy it is to learn a system, and can be quantified with measures of:

- Decreased error rates over time from the start of system usage.
- Decrease in task completion time from the start of the system usage.
- Increase in user knowledge about system facilities over time.

A sample usability goal may be:

*95% of the users can learn to use the system commands to complete the set tasks of the effectiveness goal with 20 minutes of tutored training before the test.*

Another possibility is to set learning goals over successive sessions:

*Errors should decrease to less than 2% and task completion times of x minutes should be attained by 95% of users after 5 sessions.*

Memorability. It is important that an interface is memorable so that, when returning to use the system after some time, the user does not have to relearn it. Memorability is closely linked to learnability, and is measured either by
comprehension/memory recall tests or by usage tests after an elapsed time period. Example measures are:

- Correct recall of system facilities, operational procedures or command names.
- Percentage of system facilities or commands recalled after time period $t$.
- Percentage of system commands explained adequately after time period $t$.

A usability goal for this component could be:

95% of users should be able to recall and accurately describe 90% of the system commands 7 days after training.

**Attitude/satisfaction.** Attitude is the subjective part of usability, which attempts to quantify user satisfaction with the system. It is measured by rating on a scale typically 1-7, where 1 = bad and 7 = excellent. Typical measures are:

- User satisfaction exceeds a target rating.
- User-perceived problems are kept below a set level.
- User motivation to use the system exceeds a set baseline level.

A usability goal for user satisfaction could be:

95% of the users rate the overall system quality as being 5 or better on a 7-point scale.

Lansdale and Ormerod (1994) list six types of data, which are commonly collected by evaluators in usability tests.

**Throughput**

Throughput is a measure of productivity. It might include the number of pages proof-read, menus navigated, problems solved or forms dispatched. When two interfaces supporting the same task are being compared, throughput measures provide a reasonable way of differentiating between them.
Execution time

The converse of measuring the number of specified tasks carried out in a given time (throughput) is to measure the time taken to carry out a given operation. The usual assumption is that longer performance times reflect poorer design. They are usually associated with greater difficulty, reduced throughput and user frustration.

Sometimes the differences in the times taken to carry out a task with different interface designs are small, and hard to establish statistically without many observations from a large number of subjects. However, small differences can still mean poor usability especially if users carry out the task for extended periods.

Accuracy

Accuracy usually refers to performance in which the precision, rather than the correctness, of inputs are at stake.

Errors

When user actions are clearly distinguishable as either correct or incorrect, performance can be evaluated in terms of the number of errors made. Errors can be used to infer difficulties deriving from a mismatch between the interface design and the user's understanding of it. Errors can arise from a range of causes sometimes in combination with other sources of data.

Subjective measures

Many usability tests incorporate an assessment of the user’s subjective responses, often derived from questionnaires or rating scales. They might record attitudes to a system, preferences between alternatives, and estimations of how easy things are to do.

There are two problems in the use of subjective measures in evaluation. Firstly, what people say in response to subjective rating methods is highly sensitive to the way in which they are asked. In practice, it is very difficult to design questionnaire and rating methods that are not open to biased...
interpretation. The second problem with subjective measures is that they can be misleading. What the users say they prefer and what is ultimately the best design solution is not always the same thing.

**Video tapes and systems logs**

Many systems allow for the capture of data, which records the user's inputs to the system and timing. A number of laboratories also use video recording of users' behaviour in usability trials (e.g. Good, 1985). In the case of UNIX the availability of an event logging system has also resulted in number of studies aimed at understanding UNIX expertise by analysis of these logs (Bradford et al., 1990).

Although it is easy to capture this data, it is more difficult to analyse than discrete measures of performance. A major problem is to categorize sequences of behaviour by the keystrokes or movements on a video. In addition, analysis is very time consuming.

### 2.8 Summary

The literature reviewed supports the fact there are problems with windowing system. In particular, the users tend to spend more time manipulating the interface rather than working on the task.

The literature also supports the fact that cluttered desktops is one reason why users need to manipulate the interface and clutter is due to highly dense and overlapping information. Therefore in order to reduce clutter, one must reduce the density of the information presented and to reduce overlapping information. For a Windowing system overlapping information could mean overlapping Windows.

The literature on complexity theory showed that it is possible to calculate the layout complexity of an interface for static displays of information. Furthermore, retrieval of information from displays with a low complexity value is quicker than displays with higher complexity value. Thus, lowering the complexity of a display can increase user performance on a retrieval task. The complexity equation relied on the
horizontal and vertical alignment of objects, hence to reduce the complexity of a layout objects must be aligned and grouped. In effect, by aligning objects, implied ordering of the information has taken place and therefore has reduced the clutter level.

The thesis addresses the possibility of automating and assisting the user in the task of managing and efficiently accessing windows. Our approach is based on the premise that, by freeing the user's cognitive resources from the task of managing the window aspects of the interface, more of these resources are available for application domain activities. As the problems and application tasks confronting the users become more complex and information intensive, the potential of this approach for improving overall human system performance is enhanced.
3 HOW ARE WINDOWS USED: AN EMPIRICAL STUDY

3.1 Introduction

A brief history of windowing systems has been presented in Chapter 2, in which some of the characteristics of windowing systems were highlighted together with their benefits and drawbacks. The literature review has revealed that users, using current window systems, spend a considerable amount of time manipulating windows rather than on the tasks to be done.

This chapter investigates the ways in which users actually use windows. It describes an exploratory empirical study, which has been carried out, using a relatively complex task. Seven subjects were video taped whilst performing the task, which took approximately one hour to complete. These video sessions were analysed in order to cast some light on the nature of the difficulties faced by users of windowing system. The chapter concludes by making some generalisations about window usage.

3.2 Approaches to Window Usage

3.2.1 Generic Windows Tasks

Tasks are performed to accomplish a goal. Any goal can be described in terms of a number of sub-goals. These sub-goals can then be described in terms of user-defined tasks, which can then be described or translated into window operations (known as windows tasks) and commands that achieve these sub-goals. For example, a goal can be stated as “generate a sales report”. If no sub-goals are defined then user defined tasks may be stated as follows: (i) get sales data using Excel; (ii) extract market size information from in-house database and (iii) include this information in the sales report using Word. In this section, we discuss windows tasks and generalise them so that they can represent typical user tasks.
As Card et al. (1984) have stated, an analysis of window management strategies can only be done by careful consideration of the tasks for which windows are used. They attempted to categorise tasks in terms of window functions, which they listed as follows:

- Provide more information
- Give access to multiple sources of information
- Combine multiple sources of information
- Allow independent control of multiple programs
- Remind users
- Support multiple representation

An experiment was designed to exercise these functions in user tasks, concentrating on the providing of more information, access to multiple sources of information and combining multiple sources of information.

Kandogan and Shneiderman (1997) identified an evaluation metric for user performance measurement to analyse window management strategies. Their performance measurement is based on an environment set-up task, an environment switching task and task execution times. However these tasks can be used to identify unique phases that a user has to complete to accomplish a given goal; the environment set-up phase, environment switching phase and the task execution phase.

To make the analysis easier, the analysis concentrates upon the task set-up/switching phase as this is where most of the window manipulations occur.
3.3 Experiment: Observing Users Working with Windows

3.3.1 Objective of the Experiment

The objective of the experiment is to gather general information on the ways in which windows are used for particular tasks. In particular, how are windows manipulated in order to carry out certain tasks?

It was also hoped some generalised concepts and styles of how people work in a windowing system could be identified. In particular:

- Do people adopt certain layout and window management strategies and if so what are these strategies?
- What window management support can be provided by the system to aid users whilst performing their tasks?

Other interesting questions are:

- Do users minimise/close windows when they no longer require them? Or do they start minimising/closing windows when the screen begins to become cluttered.
- Do users let their windows overlap or do they try and keep them non-overlapping?
- Do users employ any screen layout strategies in the way they perform a task?
- How often are windows moved? Do users tend to move windows a few times before they get annoyed and decide to minimise?

3.3.2 Experimental Design

3.3.2.1 The Task

The task is centred on a scenario in which the subject is asked to produce a report explaining the sudden increase in sales within a bookstore. Numerous data files and diagrams are available and the task is to collect the data and present it in a report.
Some of the data are located on other files and others have to be calculated based on given information.

A subject is asked to work through the task sheet (see appendix A.1). The task is structured so that, as a subject progresses through the task, more and more windows need to be utilised for them to complete the task. The subject is encouraged to speak clearly into the microphone, commenting on what they are doing, in particular, providing reasons for manipulating windows (re-arranging, minimising, maximising, shrinking, closing applications, and so on).

There are four main tasks (in total 22 sub-tasks); these are either environment set-up/switching activities (TSP) or execution activities (TEP) as defined by Kandogan and Shneiderman (1997) see earlier section.

The four main tasks that need to be completed to produce the report are:

- Task 1: Type some text into the report (sub-task 1-6).

<table>
<thead>
<tr>
<th>Sub-Task Num.</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TSP</td>
</tr>
<tr>
<td>2</td>
<td>TEP</td>
</tr>
<tr>
<td>3</td>
<td>TSP</td>
</tr>
<tr>
<td>4, 5, 6</td>
<td>TEP</td>
</tr>
</tbody>
</table>

*Table 3-1: Phase for task 1*

- Task 2: Draw and include a diagram into the report (sub-task 7-12). The subject will need to combine/integrate information.

<table>
<thead>
<tr>
<th>Sub-Task Num.</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>7, 8</td>
<td>TSP</td>
</tr>
<tr>
<td>9, 10, 11</td>
<td>TEP</td>
</tr>
<tr>
<td>12</td>
<td>TSP + TEP</td>
</tr>
</tbody>
</table>

*Table 3-2: Phase for task 2*
• Task 3: Calculate the monthly sales figures, produce a graph and include into the report (sub-task 13-20). The user will require access to multiple sources of information.

<table>
<thead>
<tr>
<th>Sub-Task Num.</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>TSP</td>
</tr>
<tr>
<td>14</td>
<td>TSP + TEP</td>
</tr>
<tr>
<td>15</td>
<td>TSP + TEP</td>
</tr>
<tr>
<td>16, 17, 18</td>
<td>TEP</td>
</tr>
<tr>
<td>19</td>
<td>TSP + TEP</td>
</tr>
<tr>
<td>20</td>
<td>TEP</td>
</tr>
</tbody>
</table>

*Table 3-3: Phase for task 3*

• Task 4: Calculate the quarterly sales figures, produce a graph and include in the report (sub-task 21-22). The subject will require access to multiple sources of information.

Sub task 21 & 22 simply states "Using the monthly figures... calculate the quarterly sales using the calculator... enter data in a new excel spread sheet... produce graph and copy to report." Operational phases (TSP & TEP) can not be identified. However, subjects will switch between these phases a number of times to accomplish the task.

3.3.2.2 Set-up and Procedure

The windowing system used for the experiment was Microsoft Window 3.1. A seventeen-inch monitor was used and the screen resolution set to 640 by 480. The lower screen resolution made the task more difficult and required the user to manipulate the windows more than if the resolution was higher. A VGA to PAL converter unit was used to record the computer screen activity on videotape for analysis.

Each subject filled in a simple questionnaire (see appendix A.2.) to determine their familiarity with windowing systems and their experience of applications such a MS-Word, MS-Excel and MS-PowerPoint.
3.4 Analysis of Results

Seven subjects carried out the experiment. The total duration of the experiment varied between 40 minutes to 70 minutes. However, the total time taken to complete the experiment is not a valid measure of performance as the time taken is very much dependent on individual user competency and skills, such as typing speed and familiarity with individual applications. In addition, some sub-tasks were left to the individual's discretion to decide when to stop. For example, one of the sub-tasks was to produce a layout of the shop floor in PowerPoint. Subjects could spend a lot of time trying to be very precise or produce a rough outline of the layout with very few labels. Therefore, the primary objective measure that can be used to compare performance between subjects, using different styles or strategies, is to look at the number of window manipulation operations needed to complete a sub-task.

3.4.1 Notation Used to Transcribe the Video Recordings

The videotapes were transcribed using the following notation.

<Window Operator> (Object, [Time Stamp])

Where:

Window Operator is one of a number of predefined windowing operations.
Object is the item (usually a window) that an operation is applied on.
Time Stamp is the time when, at which the user action occurred and is the relative time from the start of the experiment.

Window Operator can be any one of the following:

Window Operator: = Open | Close | Move | Resize | Min | Max | RestoreFromMax | RestoreFromMin | Alt-Tab | Exit | Activate

Task (Brief Task Description, [ST| CT Time Stamp])

Where:

[] = item in square brackets are optional
ST = Start Time (default when not specified)
CT = Completed Time
Chapter 3 How are Windows Used?

{ } = Additional comments are provided within these brackets

An Example

<table>
<thead>
<tr>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open (Program Manager, 00.00)</td>
</tr>
<tr>
<td>Open (Main)</td>
</tr>
<tr>
<td>Open (File Manager)</td>
</tr>
<tr>
<td>Task (Create Directory, ST 00.10)</td>
</tr>
<tr>
<td>Close (File Manager)</td>
</tr>
</tbody>
</table>

The complete transcribed video data for three of the subjects are presented in appendix A.3.

3.4.2 Observations

There were two distinct layout strategies used by the subjects when working on two or more windows concurrently. In the first case, information on one window is needed for another window. In the second case information from two windows is combined into the third window. The two-window situation is exemplified by Task 3 and the three-window situation by Task 4.

One-Window Situation (Task 1 & 2)

Task 1 was to open the file manager, create a directory, open Microsoft Word and type some text. All subjects followed a similar set of operations as exemplified below.

<table>
<thead>
<tr>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open (Program Manager, 00.00)</td>
</tr>
<tr>
<td>Open (Main, 00.16)</td>
</tr>
<tr>
<td>Open (File Manager, 00.19)</td>
</tr>
<tr>
<td>Task (Create Directory, ST 00.38)</td>
</tr>
<tr>
<td>Activate (Program Manager, 1.18)</td>
</tr>
<tr>
<td>Open (MS Office Group, 1.20)</td>
</tr>
<tr>
<td>Open (MSWord, 1.23)</td>
</tr>
<tr>
<td>Task (MSWord - Type Text, ST 1.50)</td>
</tr>
<tr>
<td>Task (MSWord - File Save As, 7.20)</td>
</tr>
</tbody>
</table>

In the above example one might have expected to see the MSWord Window being maximised, as this required full attention of the subject and did not rely on
information from other windows. In the experiment, 2 out of 7 subjects were observed to maximise the MSWord Window during this task. One subject maximised as soon as the application was launched and the other typed a few lines of text then realised that this window would be the only window being used until the text was typed, therefore subject decided to maximise.

The other subjects offered two explanations for not maximising the Word document. Firstly, the MSWord window was of an adequate size and few other distracting windows were open. Secondly the subjects did not want to deviate from the primary task and perform a window management operations.

Task 2 was to draw and include a diagram into the report. Almost all subjects minimised the MSWord application after task 1 before proceeding to task 2 except for two subjects. These two subjects (number 2 & 6) closed the MSWord application - one commented that he normally used to a low powered machine which could not run multiple applications because of lack of memory, so he was used to closing and starting applications to conserve memory. The other subject most probably did not realise that application was still required during later tasks.

Microsoft PowerPoint was used to produce the drawing. Again, only a few subjects maximised this application (3 out of 7), even though this was another task where no data from other windows is required. These three subjects (number 3, 6, & 7) seem to follow the characteristics of a controlled approach (see section 3.5) to window operation. The data from the questionnaire showed that these three subjects were the most experienced users of Windows and the applications used in this experiment.

Two-Window Situation (Task 3)
Task 3 calculated the monthly sales figures displayed in MSWord using the Calculator application. Subjects adopted one of two layout strategies (Tiled or Overlapping). Subjects took, on average, 11 window operation to achieve the tiled layout, whereas subjects on average achieved the desired overlapping layout using 6
window operation. However, these results should be read with caution, as the sample size for the experiment is low.

1. The Tiled Approach - subjects tiled the two applications (see Figure 3-1) so that they could switch between them easily (information contained in both windows was visible). Two out of the seven subjects (number 2 & 4) adopted this strategy. For both of these subjects, analysis of the questionnaire showed that they were infrequent users of Windows and of some of the applications used for the experiment.

![Figure 3-1: Tiled approach to task 3](image)

2. The Overlapping Approach - subjects maximised MSWord and Alt-Tab to the Calculator (one window overlapped the other). In this case the Calculator partially overlapped the MSWord window (see Figure 3-2a). However when users switched to the MSWord application (making it the active application) MSWord totally overlapped the Calculator (see Figure 3-2b). Information contained on the calculator was therefore not visible so subjects had to remember the total calculated, switch to MSWord and entered the total in the appropriate column. The majority of the subjects used this strategy (5 out of 7).
The Task Execution Phase showed that 2 out of the 5 subjects that adopted the overlapping strategy made mistakes in their calculation so had to re-do them. This is illustrated by taking extracts from the transcribed data for one of the subject for whom errors were recorded.

```
•

Task (Calc - Add weekly figures, ST 28.00)
Repeat
  AltTab (MSWord, -)
  Task (MSWord - Type total, -)
  AltTab (Calc, -)
  Task (Calc - Add weekly figures, -)
Until (all months calculated)
(This took 18 iterations when there were only 12 calculation required)
•
```

The subject took 18 iterations for which only 12 iterations should have sufficed. The additional 6 were required as the wrong values were calculated and entered in the wrong columns.

Three-Window Situation (Task 4)
Again, subjects used one of two-layout strategies to complete the task requiring concurrent use of three-windows:
(1) They re-arranged the screen so that all three applications were visible on screen and switched between the three applications. However, a problem with this approach is that each application has only a finite screen usage, hence the document needs to be scrolled to view the rest of the information.

(2) One application was maximised and pressing AltTab keys brought the other two forward. In this case, additional cognitive processing was required, as users had to remember the calculated figure to enter it in another application. Again, if the layout and the sequencing of the Alt-Tabs had been set-up properly, it was possible to alleviate the additional cognitive processing, as the calculated figure was visible.

Three months sales figures form the background window (MSWord) are typed into the calculator (see Figure 3-3) to calculate the quarterly sales figure. The total is then entered in the Excel spreadsheet, but further switching is required to locate and switch to the spreadsheet. This is achieved by using Alt-Tabs to switch between applications until the layout is as shown in Figure 3-4.
The tiled approach was used by 3 subjects (number 2, 4, & 7) whilst the remaining 4 used the overlapping approach. Even the overlapping window approach required some window manipulation as the subject specifically positioned the overlaps. This is exemplified in Figure 3-4, the calculator is positioned in the top left corner, MSExcel is resized and positioned on the bottom left corner and MSWord is maximised.

Observations show that subjects formulated solutions to how best to resolve the problem as the windows were being moved. One subject (number 6) started executing the task and then realised the problem was more complex than anticipated. The subject paused execution of the task and began a task set-up phase. An extract from the transcribed data is shown below. Eggemeier, (1988) suggests that such strategy changes can be indicators of increased workload.
2 out of 4 subjects that used the overlapping strategy made mistakes during the calculation, whereas the subjects that used the tiled approach made no mistakes when calculating the quarterly figures.

In addition, subjects, when using the overlapping strategy, also performed a number of AltTab operations sequencing through all the applications to check which applications were currently running. This was only carried out to remind the subject which application was running.

3.5 Overall Observation

During the experiment two approaches to window usage were observed and have been categorised as follows:
• The controlled approach.

• The simplistic approach.

The controlled approach to window usage is characterised by users who perform window management or housekeeping regularly. That is, they open windows that are required for the current task and close or minimise windows that are not required for the current task. They will also arrange windows to suit the current task. The simplistic approach to window usage is adopted by users who tend to limit the number of applications running concurrently. They minimise task switching and favour completion of all tasks in a certain application/window before moving on to the next task. Kahn and Charnock (1995) also observed subjects opening a new window for each new information request. They became disorientated in their tasks, and began creating windows that contained information that was already displayed in obstructed windows. Interestingly these subjects commented that they should, “close some windows”, but never did. Other subjects displayed the behaviour of what can be termed the controlled approach – always creating new windows, but closing them as soon as they had completed the immediate task.

A simple example is used below to demonstrate the two approaches. See appendix A for actual transcribed data exemplifying these approaches.

The controlled approach

1. Open File Manager and open Excel spreadsheet, marketing report and in house database
2. Close File Manager and arrange screen
3. Copy and paste sales data from Excel spreadsheet to marketing report
4. Extract market size information from in house database
5. Copy and paste market size information to the marketing report
6. Close applications

The Simplistic approach

1. Open File Manager and open Excel spreadsheet
2. Open marketing report, close File Manager
3. Switch to Excel spreadsheet and copy sales data
4. Switch to marketing report, locate relevant place for sales data and paste sales data
5. Close Excel spreadsheet
6. Open File Manager and open in house database
7. Extract the market size information and copy
8. Switch to marketing report, locate relevant place and paste market size information
9. Close applications

The above simplified examples demonstrated some of the similarities and differences between the two different approaches. The simplistic approach to window management will tend to exhibit excessive windows management operations compared with the controlled approaches.

3.6 Summary
Fewer errors were recorded when subjects used the tiling strategy than with the overlapping strategy. A higher percentage of subjects used the overlapping strategy, but no reason for using this strategy was offered. One might conclude that it was easier for subjects to do nothing and let the windows overlap, avoiding any window management operations.

However, if only a few user manipulations of the windows were required, then the subjects may have preferred to use the tiled strategy. Auto tiling, at the user's request, seems to be a feasible solution at first. However, most auto-tile functions, provided in most windowing systems or applications, tend to size all windows equally. During this experiment it was observed that users did not size windows equally but sized each window within some constraints and such that it also satisfied some of the goals. They are constrained by the screen size and also by how much screen real estate they can allocate to each window. Allocation of screen real estate is
based on how much information contained within an application is required for the current task and how many other windows need to be present for the current task.

There was some evidence to support the controlled and simplistic approach to window usage. Furthermore, as the task complexity increased, the number of window manipulations required also increased.

The experiment indicates that overlapping windows are one cause of increased cognitive complexity. Reducing the overlap may reduce this complexity. Users may sometimes avoid the necessary manipulations required to unoverlap windows, as they do not wish to deviate from the primary task and engage in a window organisation task. Together with the observations made by Bly and Rosenberg (1986), Bury et al. (1985) and the observations from this experiment, show that there is potential in an adaptive window management approach.

The notation used to transcribe the video recordings and the method used for the analyses have been valuable and on average it took around two hours to transcribe a 40 minute recording.
Chapter 4

4 THE CONCEPT OF VANISHING WINDOWS

4.1 Theoretical Considerations

From the literature review (chapter 2) clutter was identified as one contributing factor to the problem of 'Window Thrashing'. Chapter 2 and the empirical study reported in chapter 3, showed that during a search task a number of window manipulations are needed.

In addition, Tullis (1981) showed that by structuring the layout of information, users could locate information on a display quicker than using an unstructured narrative display. He found that the structured display produced a lower complexity value (based on the Shannon and Weaver – formula for Information Theory, 1949) than the unstructured narrative display.

The complexity values reduce when items/objects are aligned horizontally and vertically. For windowing systems a tiled layout would result in a lower complexity value than that of an overlapping layout. This is one possible reason why, in the experiment conducted by Lane et al, (1997), subjects completed their task quicker in the tiled windowing interface than that in the independent overlapping windowing interface. However, overlapping windows have other benefits such as offering users a greater flexibility over the layout of the display. Funke et al. (1993) defined clutter as a condition of highly dense or overlapping information, therefore, by reducing the overlapping of information (i.e. windows), some of the clutter on the desktop can also be reduced. A windowing system that encompasses both the tiling and overlapping strategies would therefore prove to be useful.

4.2 The Vanishing Windows Technique: Concept and Issues

The basis of the Vanishing Windows approach is the gradual reduction in screen real estate requirements for an unused window as time proceeds. This reduction strategy
releases real estate for the active window and reduces unwanted clutter. A window that is apparently not needed is gradually reduced in size until it reaches the size of an icon.

The reduction of inactive window size progressively increases the overall visibility of other windows on the desktop (less overlap). However, the visibility of the content of the reducing window will reduce progressively. Therefore techniques will also be needed to maximise the visual cues in vanishing windows as they reduce (See Chapters 5 and 6).

The Vanishing Windows approach poses at least two important questions:

1. How does the system decide that a window is no longer in use?
2. How does the system reduce the window size?

4.2.1 When is a window no longer needed?

Once a user has opened a window, it is difficult to know when it is of no further use (or at least no further use in the near future). There are three possible solutions to this difficulty - adopt a Task-Based approach (decide on window activeness based upon task characteristics), adopt a Task-Independent approach (define a common strategy for all tasks), or adopt a User-Modelling approach (decide upon window activeness according to past user actions and preferences).

The Task-Based approach has some difficulties. Firstly, it will necessarily be task specific, and, because of the generic nature of some applications (for example a word processor might be used for Email), it may be difficult to decide what task is actually being carried out. Secondly, the way in which a task is carried out might well depend upon the type of user running the application (for example a novice or an expert).

However, the Task-Independent approach is the most appropriate at this stage because it does not try to distinguish between tasks. Instead it defines a common strategy for all tasks. It removes the complexity of determining individual methods for deciding when windows are not required for individual tasks. A User Modelling
approach will be appropriate if personalised alternative vanishing strategies are needed. In this approach user habits on particular tasks would be noted, learned, and then used as a basis for window management decisions.

In reality, it is likely that a combination of these approaches would be successful. The User Modelling approach may be adopted in future developments of the system.

In order to present a consistent system to the user, the system will assume that "usefulness" is inversely proportional to the time since last active. The greater the time since last active, the less useful is the window. This implies that inactive widows should be reduced in the size in a linear fashion as time progresses. It is clear that this assumption is not always true, but it is important that a simple strategy is adopted at the outset. It also has the merit of being consistent and understandable.

The Task-Independent solution uses a limited amount of static knowledge, which is domain independent and makes two assumptions to simplify the design.

Assumption 1: A window not interacted with for a certain period is not required in the near future by the user. This period we call the "time-out" period. We know, of course that this assumption is not always valid. An exception to this case is that the outputting of information is not considered as an interaction occurring on a window.

Assumption 2: Users will require a continuous reminder of the reducing importance of unused windows. They will need to know when a window has exceeded its time-out period and is progressively being reducing. They will also need to know when it has effectively been removed from the system.

4.2.2 How does the system reduce the window size?
Once windows, which are no longer required, have been identified, the next objective is to remove them in a manner which provides users with a visible cue that the item is being removed. This cue should not be too intrusive and should be
designed so as to minimise negative effects on the user's concentration in the primary task.

This can be achieved and can satisfy the two assumptions stated above, by allowing the windows to vanish slowly after the initial time-out period. The vanishing effect will clearly be visible but should not intrude too strongly into the users conscious activity. A slowly reducing window will be perceived peripherally. Should the user decide that, at any time during the vanishing cycle, the window needs to be re-activate, a simple selection will prevent further vanishing and restore it to its original size and location. The time-out period will then be reset. However, a user may also hasten the vanishing process by iconising it, if required. Windows are never allowed to vanish completely, their minimum size is an icon.

4.2.3 Vanishing Windows Parameters

Vanishing will not start immediately. There will need to be a Pre-vanish Time-out Period $PreVanTime$ which is defined as the period of time elapsing between the end of the active period of an application(window) and the time at which any vanishing begins to take place on that window.

![Diagram of vanishing parameters](image)

Figure 4-1: Vanishing parameters

A window begins to vanish from an Original Size $OrigSize$, and eventually reaches a Minimum Size $MinWinSize$. We anticipate that $MinWinSize$ will be the same for all windows (the size of a standard icon). The application or the user will originally set
OrigSize. The vanishing effect therefore happens in a Number of Steps \( \text{DecSteps} \) each of which lasts for a Decay Time of \( \text{DecStepTime} \).

The Vanishing Time for a window to reach its \( \text{MinWinSize} \) is \( \text{VDecTime} \) and is given by

\[
\text{VDecTime} = \text{DecStepTime} \times \text{DecSteps}
\]

_Equation 4-1: Vanishing Time_

![Figure 4-2: Parameters for Vanishing Windows](image)

The Vanishing Rate \( \text{VanRate} \) is given by

\[
\text{VanRate} = \frac{1}{\text{VDecTime}}
\]

_Equation 4-2: Vanishing Rate_

The Reduction Per Step \( \text{VStepSize} \) is

\[
\text{VStepSize} = \frac{(\text{OrigSize} - \text{MinWinSize})}{\text{DecSteps}}
\]

_Equation 4-3: Vanishing Step Size_

Thus the parameters to be decided by the designer are \( \text{PreVanTime}, \text{DecSteps}, \text{DecStepTime} \) and \( \text{MinWinSize} \)
Chapter 4

4.2.3.1 Rate of Progress to Iconification

All windows, irrespective of their original size, will iconify after the number of vanishing steps DecSteps. One option is to make DecSteps the same for all windows. It will therefore take the same time for a large window to iconify as for a small window. This is an important design issue. An alternative viewpoint would be to ensure that the iconification time was dependent upon the Original Window Size OrigSize. This would require the VanStepSize parameter to be made dependent on the original window size of each window and windows will shrink or reduce at different rates dependent upon the original size.

The second alternative seems more reasonable. A larger window is likely to be more important than a smaller window, hence it should take longer to vanish. However, this is outside the scope of this thesis and was not evaluated.

4.2.4 Vanishing Strategies

The section above discussed the issues relating to the vanishing process and the implications of varying some of these parameters and suggests the vanishing strategies that can be adopted.

The fundamental strategy is to vanish inactive windows over a period, but there are several methods that can be applied. Each method will have a different impact on the users. Possible strategies are:

- After a time out period, the inactive windows start to shrink in size, giving the impression that the window is vanishing.
- After a time out period the active window begins to grow while the inactive windows begin to shrink in size
- After a time out period, the inactive windows become un-overlapped by moving to the sides and adopt a tiling type layout.

STRATEGY 1: After a time out period inactive windows start to shrink in size, giving the impression of the window vanishing or fading into the distance.
The example in Figure 4-3 shows this strategy in use. The user is using four windows; a word processor (MS-Word), MS-Power Point, and two directory views (MS Explorer). The user is currently interacting with the word processor. Figure 4-3a, b and c shows the screen after a short time. Since MS-Word is the one being interacted with, all other windows begin to vanish. In Figure 4-3d, all inactive windows have completed vanished into icons.

![Figure 4-3: Vanishing Windows by means of window resizing](image)

**STRATEGY 2:** After a time out period, the active window begins to grow while the inactive windows begin to shrink in size

This is a plausible solution but the possible distraction effects on users need to be assessed. How will users feel when the window they are working on actually begins to change size?

**STRATEGY 3:** After a time out period, Inactive Windows become un-overlapped by moving to the sides and possibly adopting a tiling type layout. Figure 4-4 shows this strategy being applied to a windowing system. The main window being used remains unaffected, whilst the other two windows, which are not being utilised, are slowly beginning to un-overlap, until eventually they are distinct and are tiled.
This method is a cross between a tiling system and an overlapping windowing system. It is a hybrid of the two systems and may offer the benefits of each. It maximises screen usage. Further maximisation of screen usage can be achieved if desired by expanding the inactive windows to fill the vacant space similar to most tiling based system. Cohen et al. (1986) demonstrated a similar concept. When the user enlarges a window, all other windows are shrunk by a proportional amount. This was called the RTL/RTL system (Siemens Research and Technology Laboratories Rectangular Tiled Layout).

4.3 Algorithms for Vanishing Strategies
Windows can be moved and shrunk based on a number of parameters and a number of strategies can be applied to aid this process. Parameters such as the location of the overlapping region and distance to neighbouring windows or edges are used in the decision process. Two primary strategies can be adopted to move and shrink windows. These are the co-operative and the self-centred approaches.
Chapter 4

The Co-operative Strategy - this involves the negotiation between windows regarding their movement and size. For example, in a two window overlapping situation, one window overlaps another window. One is the window being overlapped and the other overlapping it. For the overlapping region to be made visible either the window being overlapped or the window overlapping it can be moved or shrunk. Negotiation between the windows can take place to determine the most efficient way to move/shrink so that the overlapped region becomes visible. Each window will have to calculate the consequence of its action and then, between them, they will have to decide to move window with the least consequences.

The Self-centred Strategy - as the name suggests, each window moves/shrinks itself to make itself visible. Each window assesses whether it is being overlapped and the sides being overlapped. This information is used to determine the direction the window has to move/shrink to reveal itself. Additional information regarding distance to neighbouring windows is gathered to improve the decision-making algorithm.

A Hybrid Approach - In this strategy, a number of decisions must be made by the window manager in real time to fill its requirement. This will require some intelligence on the part of the window manager. The window manager will have to gather information regarding its environment; in particular the location, size and the overlaps of each window. Additional information such as time of creation, time since last interaction time, and frequency of interaction for each window can be utilised by the window manager to make its decision (Funke et al. 1993; Miah et al. 1997). As a first step, this seems to be the best initial strategy to adopt.

4.3.1 Top Level Algorithm

Toolbars on inactive windows are of no value to the user and only occupies valuable screen real estate (see Chapter 5 or Miah and Alty, 1999). By removing the toolbars, additional workspace is made available for the application. Also, increased recognition rate for a window can be achieved by transforming the contents of the document by highlighting or concentrating keywords (words that can unique
identify a document) (see Chapter 6 or Miah and Alty, 1998). Therefore, toolbars can be removed from inactive windows and the contents of the inactive window can also be transformed. Similarly, scroll bars can also be removed from inactive windows. However, no scroll bars were removed for the current implementation.

```
For each inactive window
{
    If timedOut
    {
        Switch to adapted/transformed view
        Remove toolbars
        Align to Side
        If Window is overlapping
        {
            Reduce Size by one step
        }
    }
}
```

*Figure 4-5: Top level pseudo code*

This top-level algorithm is performed continuously periodically. However, the heart of the algorithm is to align the window to a side. This seems to be straightforward but how do we determine which side to align it to?

4.3.2 Vanishing Operations

A number of high level vanishing concepts can be defined and applied in a number different ways. The way in which these operations are constructed defines a vanishing method or tactic. These operations are Shrink, Migrate, Shunt and Evade.

**Shrink** - reduces the window size.

**Migrate** - This simply moves the window away from the overlapping side towards an edge of the desktop window.

**Shunt** - This is an extension to the *migrate* operation. Migration can cause a window to hit another adjacent window. If this happens these windows should move together in the immediate migrate operation.

**Evade** - This is an alternative to the *Shunt* operation. Rather than propelling adjacent windows together, this operation tries to avoid any windows in its path.
when moving to an edge of the desktop. This will side-step the obstructing window to reach an edge.

Figure 4-6: State transition diagram for Vanishing Operations

Figure 4-6 shows the state transition diagram for the implemented strategy. Transitions between these operations occur when windows overlap, hit an edge of the desktop window, when there is no overlaps or when a window hit an adjacent window.

These operations need to assess their situation and make a decision regarding their course of action. The ultimate goal of a window is to become totally visible without any overlaps. This goal can be achieved using a combination of migrate and shrink operations. A window has two options; it can shrink first then migrate or migrate first then shrink. Some of the drawbacks of shrinking first are listed below:

- Unoccupied desktop space will never be utilised by the window
The window will be unnecessarily shrunk (window display area reduced) when it could have simply moved away, whilst retaining it size and still achieving its objective of becoming totally visible with no overlaps.

Therefore, it is better to design an algorithm to migrate first and then shrink if still required.

All of these operations at some stage need to make decisions regarding the direction they need to move or shrink towards to meet their objective. This decision can be based on a number of factors:

- Which sides are partially overlapped by other windows
- If there are no overlapping windows on a side then what is the distance to the nearest neighbour.

High level rules governing the window behaviour can be put into action. For example, a rule may state that the window moves away from the overlapping side. A number of tactics can then be employed to achieve this objective. These tactics are deployed using a combination of vanishing operations (Migrate, Shrink, Shunt, and Evade).

Tactics are executed by primitive operations. Primitive operations are simple movements of a window in the migrate operation or a reduction of the window size in the shrink operation. These primitive operations require a direction to move or shrink. The direction is governed by a set of rules and these are formulated in the next section.

### 4.3.3 Operations Employed in the Vanishing Windows System

#### 4.3.3.1 Migrate: Formulation of Rules for Direction of Migration

One of the first operations a window does when it detects that it is being overlapped is to migrate. It can migrate in four possible directions, Top, Bottom, Left or Right or composites of these four directions, Top-Left, Top-Right, Bottom-Left & Bottom-Right.
Chapter 4

The migration strategy currently employed is to move away from overlapping sides. If two or more choices are available then the side is selected which is farthest from its nearest neighbour.

So, if sides of a window are labelled Top (T), Bottom (B), Left (L) and Right (R) then under the following conditions the possible manoeuvres are as indicated by Table 4-1. Zero (0) indicates no overlap on a side and one (1) indicates that there is an overlap on a side.

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>R</th>
<th>B</th>
<th>Migrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Don’t Move</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>L</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>TL</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>L OR R</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>BL</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>L</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>TR</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>T OR B</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>BR</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COMXY</td>
</tr>
</tbody>
</table>

*Table 4-1: Maneouers available for the Migrate operation*

If a single side is overlapped, then the direction of migration is simply to the opposite side. If two adjacent sides are overlapped then migration is towards the opposite adjacent sides. However, if two opposite sides are overlapped then the direction of the migration operation can be one of two choices. These choices depend on whether the opposing sides are Left and Right or Top and Bottom. If it is Left and Right then the possible migration is towards the Top or Bottom. The
algorithm decides between the two by selecting the side that has the greatest distance to a neighbour on that side.

```
Select Case (number of sides overlapping)
Case 1:
{ Migrate towards opposite side }
Case 2:
{ If overlapping sides are adjacent
  { Migrate towards opposite adjacent sides }
  Elseif overlapping sides are opposite each other
  { Migrate towards side with no overlap & side with greatest distance to neighbour }
}
Case 3:
{ Migrate towards side with no overlap }
Case 4:
{ If all four sides not overlapped by a single window
  { Compress in X and Y direction }
  Elseif overlapping window is not maximised then
  { Migrate towards a side with greatest distance to neighbour }
  Elseif
  { Do nothing }
}
```

4.3.3.2 Shrink – Formulation of Rules for Direction of Reduction

Different methods are employed to shrink a window along either or both of the x and y axes. These are detailed below.

- **SHRINK ALONG A SINGLE AXIS** – Shrink to the Right (STTR), Shrink to the Left (STTL), Shrink to the Top (STTT), Shrink to the Bottom (STTB).
• SHRINK ALONG BOTH AXES SIMULTANEOUSLY - Compress$^1$ along x-axis (COMX), Compress along y-axis (COMY), Compress along x and y-axis (COMXY), Shrink to the Bottom Left (STTBL), Shrink to the Bottom Right (STTBR), Shrink to the Top Left (STTTL), Shrink to the Top Right (STTTR).

Deciding on which of these methods to use depends on the number of sides being overlapped and on which sides are the overlaps.

If an overlap occurs on a single side then selecting the method to use is straightforward. That is, shrink to the opposite side of the overlap. However, if overlapping occurs on more than one side then a much more complex algorithm is required. Simple rules for manoeuvres are as follows:

$$\text{Manoeuvre} = \begin{cases} \text{STOS};(\text{olaps} = 1) \\
\text{STOC};(\text{olaps} = 2) & \text{(oconfig} = \text{adj}) \\
\text{CAFO};(\text{olaps} = 2) & \text{(oconfig} = \text{opp}) \end{cases}$$

Where

STOS = Shrink To Opposite Side  \hspace{1cm} \text{olaps} = \text{Overlaps}
STOS = Shrink To Opposite Side  \hspace{1cm} \text{adj} = \text{adjacent}
STOC = Shrink To Opposite Corner \hspace{1cm} \text{opp} = \text{opposite}
CAFO = Compress Away from \hspace{1cm} \text{oconfig} = \text{Overlap}
Overlaps \hspace{1cm} \text{Configuration}

When overlapping occurs on three or more sides, we simply select either two adjacent sides or two opposite sides and apply the manoeuvres as indicated above.

However, further complications are added by overlaps occurring at opposite corners. Consider the following case:

Two windows at a diagonal overlap a window (see Figure 4-7). The aim or the objective of a window is to become totally visible with no overlaps. The window has

---

$^1$ Compress is when two opposite side reduce towards each other, whereas shrink is when a single side moves towards the opposite side.
one available operation it can perform but has two tactics that it can employ using this one operation to meet its objective.

The two tactics are either to compress in the direction of the $x$-axis (COMX) or the $y$-axis (COMY). If the window is compressed in the $x$-axis then the resulting window shape will be tall and narrow indicated by the dashed lines. Alternatively, if the window is compressed in the $y$-axis the resulting window shape will be short and wide, indicated by the dotted lines.

At this point, a tactical decision needs to be made. Since both tactics achieves the overall goal of removing the overlap, a secondary goal is required to select between competing tactics. The secondary goal is then to retain maximum window visibility size.

Visible window size after applying COMX

\[ \text{VisibleArea} = h(w - (\text{owl} + \text{ow2})) \]

Visible window size after applying COMY
VisibleArea = \( w(h - (oh1 + oh2)) \)

Since \((ow1 + ow2)\) and \((oh1 + oh2)\) represents the total horizontal and vertical overlaps respectively. The lower of the two values results to a greater window size. Therefore, selection between the two tactics is based on the following equation:

\[
SelectedTactic = \begin{cases} 
    \text{COMX}; (ow1 + ow2) < (oh1 + oh2) \\
    \text{COMY}; (oh1 + oh2) < (ow1 + ow2)
\end{cases}
\]

4.3.3.3 Shunt - Formulation of Rules

Although the shunt operation has not been implemented explicitly, a window begins to shunt as a consequence of a migrate operation. Therefore, the shunt operation has been implicitly implemented. This implicit implementation occurs as a result of a window migrating on top of another window, which in turn causes it to migrate giving the impression of a shunt operation.

For a proper implementation of the shunt operation, when a window hits an adjacent window, these windows should be treated as a single window and any further operations (shrink or migrate) should effect both windows as if they were a single window. When these windows are treated as a single window then the rules developed for shrink and migrate apply to this composite window.

4.3.3.4 Evade - Formulation of Rules

As mentioned earlier, evade is an alternative operation to the shunt operation. Decisions as to which direction a window should evade is very similar to the ones developed for the migrate operation.

Consider the example shown in Figure 4-8. Window \( m \) is being forced to migrate towards the left. However, window \( ow \) is obstructing window \( m \) from migrating. For Window \( m \) to go round the obstruction (evade), it must decide whether to move round the top of the obstruction or round the bottom of the obstruction. To make this decision, distances from the obstructing window to the top and to the bottom of the window need to be calculated. However, if there are windows to the top or
bottom of the obstructing window, then it is the distance to these adjacent windows that are calculated (as shown in the diagram). The window decides to evade in the direction where the distance between the obstructing window and the adjacent window is greater. In the example, this is towards the bottom, since $2y$ is greater than $y$. If $h$, the height of the migrating/evading window, is less than $2y$, then the window can safely pass in-between the obstructing window $OW$ and the bottom adjacent window $BAW$. However, if $h$ is greater than $2y$ then a subsequent shrink operation is required.

![Diagram of window interactions](image)

*Figure 4-8: Some possibilities for the direction of the Evade operation*

Once algorithms for individual vanishing operations have been developed and tested, further strategies can be developed to intelligently switch between operations.
4.3.4 Intelligent switching between Shrink and Migrate Operations

Earlier, it was suggested that a migration followed by a shrinking operation provides a good adaptation strategy. An alternative to this approach is to use an intelligent switch between the two operations (Figure 4-9b). Rather than a serial approach where the window migrates away from the overlapping sides and then starts shrinking if it is still being overlapped (Figure 4-9a). The intelligent switch changes the operations back and forth between migrate and shrink.

4.4 Metrics for Evaluating Vanishing Algorithms

A number of metrics for evaluating vanishing algorithms is outlined below. These can be used to compare and contrast between strategies and be used as a benchmark for how well a certain strategy performs in meeting its objective. The objective is to maximise screen real estate usage, to minimise clutter and retain maximum visual cues.

4.4.1 Distance From Original Position and Size

People use positional cues to remind themselves of where objects are located. For example, 'the book is on my desk, on the left hand side of the telephone'. Similarly objects on the computer's desktop have positional cues which are used by users to locate relevant objects. Each window on the desktop has a position and when a user
begins to work, a mental model of the position of each window on the desktop is formed. This model is accessed to locate an individual window. When users move windows around on the desktop, the mental model is updated to reflect the changes. Nevertheless, the vanishing windows system should minimise the distance moved from the original window position and minimise the change in the size of the window.

4.4.2 Make maximum use of the Screen Real Estate

When the adaptation has reached a final point, utilised screen space can be calculated and used as a guide to how successful the algorithm was at maximising the screen real estate. However, determination of the final point can be difficult, as the user will generally switch between windows when proceeding through a task.

4.5 Some Associated Problems of Vanishing Windows

In a commercial version of Vanishing Windows, some auxiliary problems need to be addressed. These are:

The current system assumes that a window is in use when it is the active window (i.e. the window has mouse and keyboard focus). However, 'use' might also include outputs. The user might want the window to remain present as long as output is being directed towards it. For instance, a user may carry out a search on a web page, and, whilst this search is being conducted, the user may continue with other work and still monitor the progress the search. The user should be able to specify additional constraints to prevent vanishing occurring on individual windows.

The system performance needs to be improved so that no delays are observed by users when they switch between applications. The algorithms developed will need to be rationalised for operational speed.

Implementation of additional features such as user defined constraints on Windows need to be incorporated into the system. For example, the facility to prevent specific windows from vanishing. In addition user configurable settings for the rate at which vanishing occurs and possibly the vanishing strategy to employ should be provided.
5 VISUAL RECOGNITION OF WINDOWS: EFFECTS OF SIZE VARIATION AND PRESENTATION STYLES

5.1 Overview
This chapter reports on an experiment carried out to determine the effects of window size and presentation styles on user's ability to recognise a document contained in a window. The experiment investigates recognition performance of users using two presentation techniques (Scaled and Cropping). This is an important experiment since the Vanishing Windows system reduces window sizes of inactive windows. Therefore, the effect of window size on recognition is investigated.

Different sizes of windows containing a document were presented to the user in one of two presentation styles (scaled and cropped). The results were analysed both in terms of accuracy of identification and response time for correct and incorrect identification.

5.2 Introduction
Window systems provide multiple views on computer applications. These views can either present different aspects of the same task, aspects of different tasks, or a mixture of the two. Only one window is currently active and frequently obscures the non-active windows. To make a window active, the user needs to:

1. Decide the new task
2. Locate the appropriate inactive window
3. Activate it
Deciding on the new task and activating it are straightforward tasks, but on a crowded screen, locating the appropriate inactive window may be time consuming and error prone.

In choosing the required window, the search task is concerned both with recall (what am I looking for?) and with recognition (is this is?). In the windows case, covered windows are recalled for its contents and recognition performed on uncovered windows to check if this is the target document. Since users usually know what they are looking for (either the form e.g. “Excel window”, “Word window” or content of the window e.g. “letter about expenses”) the task is mainly recognition. In accomplishing this recognition task, the user needs visual cues from the different windows (either in form or content) They are therefore seeking a collection of objects in a defined target domain.

In a multiple window situation, a user often has to identify the content of a window when only presented with a partial or scaled view. Visual cues to the identity of the contents of the window are partially hidden, and users have to make judgements about the identity of the window (or document) based on the limited cues available. Below are examples of partial and scaled views.
Visual Recognition of Windows

A common problem for the user is to identify the correct or scaled view presented. Visual cues to the identical window are hidden. There are have to make judgments about documents based on the limited cues available. It means by partial and scaled views.

The object of the experiment is to determine the users ability to distinguish and identify a document. Contents of the window can be presented in two different partial and scaled views.

Even windows, which have not been used for a considerable length of time, may eventually be reactivated. Additionally, a user might see a window in the process of vanishing and decide it need to be reactivated. However, if the target window has reduced considerably in size (or even iconised) it might not be easy for the user to uniquely identify it from the set of reduced windows. Even if the window is not iconised, the window title might no longer be wholly in view and only a fraction of the contents might be visible.

The question therefore arises as to how to manage the visual aspects of the content of the window as it reduces. Two immediate techniques suggest themselves:

Reduce the whole window appearance in proportion to the reduction state. This will maintain the overall view of the window but rapidly reduce actual readability, as shown in Figure 5-2b. Since the layout or shape of the document is maintained, the
ability to search between several documents based on their overall appearance is potentially exploitable (Lansdale, 1991).

Reduce the window size but keep the content the same size. This is like cropping, one simply sees less of the window contents, as shown in Figure 5-2c.

![Figure 5-2: Reducing window size - (a) original window (b) reducing whilst scaling content of window and (c) reducing whilst cropping content of window](image)

The former approach provides better cues as to the overall function of the window (for example an EXCEL work sheet would be visible, if not readable). The second approach provides readable cues (but which progressively become partial and not distinctive). The reduction technique is also more useful when two windows of a different type are reducing (e.g. EXCEL and WORD) because the distinctive patterns will remain visible for a considerable part of the reduction period.

The two techniques enable windows to be reduced by either “cropping” or “scaling”. Which technique provides better visual cues so that a user can identify and reactive windows when they are vanishing?
Although performance for correct identification of target window will vary between the two strategies, as screen utilisation increases beyond a certain point, performance between the two strategies is likely to be the same (see Figure 5-3).

![Graph showing performance vs. screen utilisation]

**Figure 5.3:** A simple model of reduction strategies

### 5.3 Experiment 1: Scaled Versus Cropping

An experiment was constructed to examine the effects of reducing window size under the different presentation styles (cropping and scaling), on a user's ability to identify a target document. The title bar is one method by which the document can be identified. However, the title bar can only accommodate a limited number of characters. Often, this may not be enough information to identify the document or the user may just simply forget the title of the document. For this experiment and all subsequent experiments the title bar contained the same name for all documents. Hence, users could not use it to identify the document.

Software was written to present one of four possible documents, in various window sizes, and in one of the two possible presentation styles. The four documents were research papers on different subject areas. There was a paper on Mobile Computing, a paper on Information Technology and the Construction Industry, a report on Video Conferencing and a paper on the Application of Fuzzy Logic to Radar
Receivers. The window size was allowed to vary between 0-100% of the screen size, where 100% of the screen size was 600 by 440 pixels. The screen was normal laptop size. (25cm x 19cm).

The left/top co-ordinates of the window were fixed at location (0,50) pixels on the screen. The width was allowed to vary between 50 to 600 pixels and the length between 50 to 400. Screen size was therefore constrained between 50 x 50 pixels and 600 x 400 pixels. The desktop screen size for the experiment was 600 x 440, that is 264,000 pixels\(^2\) and this was taken to represent 100% utilisation of the desktop area. The screen resolution was set to 800 x 600 pixels.

The 10 subjects who took part were computing research students. They were all familiar with technical computer documents.

**Sample size**

Two hundred sample screens were presented to each subject. These samples had the following distribution:

<table>
<thead>
<tr>
<th></th>
<th>0-9%</th>
<th>10-19%</th>
<th>20-29%</th>
<th>30-39%</th>
<th>40-49%</th>
<th>50-59%</th>
<th>60-69%</th>
<th>70-79%</th>
<th>80-89%</th>
<th>90-100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cropped</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Scaled</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

*Table 5.1: Sample Screen Distribution*

There were 10 sample screens in each of the screen utilisation groupings and for each presentation styles.

Subjects received a randomised sequence of screens, but all subjects received the same random sequence of screens so that comparisons between the users could be made. The sequence were randomly generated and coded as a lookup table.

**5.3.1 Experimental Procedure**

Subjects were given a paper copy of the target document (Mobile Computing) and asked to familiarise themselves with it. A 15-minute time limit was placed on this
process. During the experiment, subjects were allowed to refer back to this paper document if required. However, subjects were made aware of the fact they were being timed, hence referring back would impose a penalty.

The software presented two hundred sample screens (of varying sizes) chosen at random from the four documents (including the target document) and the subject was simply asked to respond by click on one of three buttons - “Yes” this is the target document, or “No” this is not the target document, or “Don’t Know”. The response time between presentation and answer was recorded. All answers were then analysed to determine if they were correct, incorrect or “don’t know”.

Half the above presentations were Cropped. The other half were Scaled. These were mixed at random.

5.4 Results of Analysis
The performance for both reduction strategies was analysed using two criteria. The first criterion for analysis was the accuracy of identification of the target document. The second criterion was the response time for correct identification of the target document.

Also examined was the relationship between window size and number of correct identifications; and window size against response time for correct identification of the target document. An absolute value for window size was not used. Instead, the percentage of screen utilisation was used.

5.5 Target Document Identification Accuracy
Higher accuracy of document identification was obtained when reducing the size of the window using the cropping technique as opposed to the scaling technique (see Table 5-2). This suggests that it is easier to identify the window using the cropping reduction technique compared to that of the scaling technique.
The overall accuracy is increased from 68.2% to 70.9% in favour of the cropping strategy. However, this is not a very significant increase in performance. More detailed analysis is required to determine how accuracy is effected at different screen utilisation percentages for both styles of reduction.

Graph 5-1 and Graph 5-2 shows how the accuracy of identification varies as the screen utilisation varies for scaled and cropped presentation respectively. The screen utilisation was later grouped into bin sizes of 10 for further analysis. This is shown in Graph 5-3 and Graph 5-4.
Further analysis showed that the accuracy varied as window size varied (see Graph 5-3 and Graph 5-4). Both reduction strategies tended to follow a general trend. That is they tend to have lower accuracy at the lower end of the screen size. This is an expected observation. The critical point is around the 30% screen utilisation.

Graph 5-3: Accuracy of identification as window size varies for the crop reduction strategy
Chapter 5 Visual Recognition of Windows

Graph 5-4: Accuracy of identification as window size varies for the scaled reduction strategy

A plot of both reduction strategies for number of correct answers against screen utilisation percentage confirms that for screen utilisation of <30%, the crop reduction strategy shows a higher performance than that of the scaled strategies. In addition, as expected above 30% utilisation both strategies seem to level off (see Graph 5-5).
A comparison was carried out between the means of number of correct identification for the cropped, and scaled, reduction strategies for <30% screen utilisation using Student's t-test. A significant (t(9) = 4.34, p < 0.001, 1-tail) difference was found between the two groups.

5.5.1 Removal of Redundant Information

One of the fundamental arguments in favour of the Vanishing Windows approach is that valuable screen real estate is wasted by inactive windows. In addition, they add clutter to the screen. Searching for the desired window is one of the activities that a user has to perform in a multiple windowing environment. As the number of windows on the desktop increases, this search task becomes increasingly difficult and time consuming. Overlapping windows can obscure a window, hiding relevant information. Hence, users have to move these windows to reveal the information. Once moved, the visible window may not be the target window and the whole
process must be repeated until the desired or target window containing the required information is found.

If the window is reduced or obscured, then less of the workspace is providing valuable visual cues to enable the user to make a decision. In addition, if the window is reduced to such a size where only the toolbars are displayed (and no workspace) it becomes very difficult for the user to decide whether this window contains the required information or not. Thus, valuable workspace is also wasted in windows by redundant toolbars. This problem can be minimised using adaptive toolbars. Miah et al. (1997) demonstrated this technique for reducing wasted workspace in an application by limiting the number toolbars displayed and adaptively displaying only the required toolbars based on user needs.

Although adaptive toolbars do increase valuable workspace area for active windows they do nothing for inactive windows. For inactive windows the toolbars serve no useful purpose so the Vanishing Windows system can take advantage of this fact and remove the toolbars of inactive windows so that space usage can be maximised. When the application is reactivated, the toolbars can be put back in the application so that the expanded window appears the same as before vanishing. With Toolbar removal, the Vanishing Windows system can reduce the size of inactive windows without further loss of visual cues from the application.
Chapter 5 Visual Recognition of Windows

Figure 5-4: Toolbars wasting valuable workspace – (a) window without toolbars and (b) same window with toolbars.

As can be seen from Figure 5-4, simply removing the toolbars and the menu bar from the window reveals more of the content of the document.

Microsoft windows application such as MS-Word, MS-Excel, MS-PowerPoint and others all have a fixed toolbar width. The length of the toolbar reduces as the length of an application window is reduced. The items on the toolbar are cropped as this reduction takes place. However, the menu bar behaves differently. As the window size is reduced the items on the menu bar wrap around to the next line. This, in effect, increases the height of the menu bar. Hence, the visible workspace within an application is reduced.

The wrapping of the menu bar occurs when the width of the window is less than the total width of the items on the menu bar. Figure 5-5 shows the overall usable area varying as the window size varies. The “usable area” is defined as the total window area minus the area taken by the menu bar and toolbars. If a window is placed on the diagram as shown, then the point at which the right edge of the window crosses the menu bar line can be taken as a point, and a rectangle drawn. The area of this rectangle is the area occupied by the menu bar.
Usable area for a window containing 1, 2, 3 or $N$ toolbars can also be calculated by simply using the appropriate toolbar line on Figure 5-5.

For a Window:

$$U_A = (W_L W_H) - (MB_A + N_{TB} TB_A)$$

*Equation 5-1*

Where:

- $U_A =$ Usable Area
- $W_L =$ Window Length
- $W_H =$ Window Height
- $MB_A =$ Menu Bar Area
- $MB_{T\text{max}} =$ Menu bar Max Length
- $N_{TB} =$ Number of Toolbars
- $TB_A =$ Toolbar Area
- $TB_W =$ Toolbar Width
- $MB_{T\text{max}} =$ Menu bar Max height
- $W_{L\text{min}} =$ Min Window Length
Chapter 5

Visual Recognition of Windows

\[ TB_A = TB_w W_L \]

Equation 5-2

Using the model described in Figure 5-5 the menu bar area (MB_A) is as follows:

\[ MB_A = \begin{cases} 
W_H W_L; & (W_L \leq W_{Lmin}) \text{ and } (W_H \leq MB_H_{max}) \\
MB_{Hmax} W_L; & (W_L \leq W_{Lmin}) \text{ and } (W_H > MB_H_{max}) \\
\left( \frac{50}{231} W_L + 70 \right) W_L; & W_{Lmin} < W_L \leq MB_L_{max} \\
TB_w W_L; & W_L > MB_L_{max} 
\end{cases} \]

This can be used to recalculate the screen utilisation percentage in terms of the application usable area.

5.5.2 Recalibrated Results

As described in the previous section the toolbar uses valuable screen space. This space can be bested utilised to display contents of the document. Rather than repeating the experiment without the toolbars, the results obtained above can be recalibrated to compensate for the amount of space used by the toolbars. This can be achieved by recalculating the useable area using Equation 5-1.

The results were recalculated using the data from the experiment outlined in section 5.3 and using Equation 5-1. This calculates the screen utilisation based on usable screen area rather than based on the window size. This, in effect, removes the menu and toolbars and generates a set of results based only on the usable area.
Graph 5.6 shows the percentage of correct identifications of the target document as the window size varies. The screen utilisation is divided into 9 groups each group increases utilisation by 10%. The graph shows a plot for both cropped reduction with toolbar (with TB) and without toolbar (no toolbar – NTB). As we would expect, higher correct identification is achieved for the NTB (No Toolbar) case. This is also the case for the scaled reduction strategy as shown in Graph 5.7.

If 80% accuracy is taken as a cut off point, then for the cropped case, an 80% accuracy level is reached when screen size is 40% of the total screen size with no toolbars. However, with the toolbars, this figure was achieved at screen utilisation percentage of 45%. Similarly, for the scaled case, 80% accuracy level was reached at 35% screen utilisation for the no toolbar case and 40% for the with toolbar case. Although, it seems that the scaled case reaches the 80% accuracy at a lower screen
utilisation, this is offset by the fact that the cropped style scores higher accuracy for the 20% screen utilisation than its counterpart in the scaled case.

Valuable application screen is therefore made available by removing the toolbar. This meant that more visual cues can be displayed and hence an increase in the identification percentage is observed. As expected both strategies show better performance with the toolbars removed.

5.6 Response Time for Correct Identification of Target Document

Given enough time and patience, subjects can identify a document with very high accuracy level. However, when working with documents and switching amongst them, subjects do not wish to waste too much time searching for the documents. They would like to be able to locate the required document quickly and efficiently. Hence, a measurement of response time for correct identification is also relevant, though emphasis on response time should not be at the expense of accuracy.
Response time for correct identification can also be used to judge the ease by which subjects can identify a document. A quicker response time will mean that it is more easily identified.

Table 5-3 presents the basic descriptive statistics for the average response times for identification of the target document. Statistics are shown for all three cases of identification (Correct, Incorrect, and Unidentified-DK) and for both presentation styles (Cropped and Scaled).

<table>
<thead>
<tr>
<th></th>
<th>Cropped</th>
<th></th>
<th></th>
<th>Scaled</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C</td>
<td>IC</td>
<td>DK</td>
<td>C</td>
<td>IC</td>
<td>DK</td>
</tr>
<tr>
<td>Mean</td>
<td>4.68</td>
<td>7.32</td>
<td>3.24</td>
<td>4.64</td>
<td>7.48</td>
<td>3.30</td>
</tr>
<tr>
<td>Median</td>
<td>4.89</td>
<td>6.04</td>
<td>3.23</td>
<td>4.77</td>
<td>5.61</td>
<td>3.16</td>
</tr>
<tr>
<td>Std Deviation</td>
<td>1.88</td>
<td>4.98</td>
<td>0.86</td>
<td>1.70</td>
<td>7.91</td>
<td>0.85</td>
</tr>
<tr>
<td>Sample Variance</td>
<td>3.54</td>
<td>24.77</td>
<td>0.75</td>
<td>2.89</td>
<td>62.64</td>
<td>0.73</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>-1.33</td>
<td>-1.65</td>
<td>-0.40</td>
<td>-1.33</td>
<td>4.15</td>
<td>1.56</td>
</tr>
<tr>
<td>Skewness</td>
<td>0.30</td>
<td>0.08</td>
<td>-0.38</td>
<td>0.23</td>
<td>1.94</td>
<td>0.15</td>
</tr>
<tr>
<td>Minimum</td>
<td>2.47</td>
<td>0.00</td>
<td>1.65</td>
<td>2.53</td>
<td>0.00</td>
<td>1.68</td>
</tr>
<tr>
<td>Maximum</td>
<td>7.55</td>
<td>14.00</td>
<td>4.36</td>
<td>7.43</td>
<td>27.00</td>
<td>4.89</td>
</tr>
<tr>
<td>Range</td>
<td>5.08</td>
<td>14.00</td>
<td>2.71</td>
<td>4.90</td>
<td>27.00</td>
<td>3.21</td>
</tr>
</tbody>
</table>

Table 5-3: Descriptive statistics for response times for cropped and scaled presentation styles

The following observations can be made from the descriptive statistics:

1. The means for all cases of identification and presentation style do not differ significantly. However, detailed analysis is required as the window size and the contents of the window may affect response times.

2. Both the standard deviation and the sample variance for the “don’t know” (DK) case is lower and much more closer to zero than the correct and incorrect identification cases for both presentation styles. This indicates that all subjects responded with “don’t know” within a close proximity of time.
3. The Kurtosis and the Skewness values provide more information about the shape of the distribution. Kurtosis is the term used to identify the degree of steepness or shallowness of a distribution. The majority of the data presented exhibit a shallow distribution except for the Scaled-Incorrect case (Kurtosis value of 4.15). Skewness values are similar for all cases and again, except for the Scaled-Incorrect case, which shows the distribution to be slightly more skewed (Skewness value of 1.94) – positive skew.

![Graph 5-8: Average response times for correct, incorrect and unidentified (DK) results for both presentation styles](image)

The distinguishing feature that can be seen from Graph 5-8 is the higher response time for incorrect answers for both presentation styles. This indicates that subjects tend to take longer to reach a decision, yet when they finally do reach a decision, they often get it wrong. Closer examination of the standard deviation and sample variance for incorrect answers shows higher values for the Cropped case than for the Scaled case. This indicates that the subjects response times were more varied and the variation is greater in the Cropped case. This variation is confirmed by a larger range value.
One reason for this variation is that the readability of words diminishes as the window size decreases. Therefore, subjects tend to spend more time scanning the window content to see if they can identify words to assist in the decision. Since words are more readable in the Cropped case, subjects usually spend the time and effort necessary to make a positive identification. However, for the Scaled case, subjects tend to try to make a positive identification and, after a while make, a hasty decision.

5.6.1 Screen Utilisation and Response Times

The screen utilisation percentage is the relative size of the target window compared to that of the desktop size. Response times have been grouped in bin sizes of 10. Response time is likely to vary with screen utilisation percentage. Table 5-4: shows how screen utilisation percentage effects response time. Again, descriptive statistics are presented to get a feel for the data before detailed analysis is performed.

<table>
<thead>
<tr>
<th>Screen Utilisation percentage</th>
<th>0-9</th>
<th>10-19</th>
<th>20-29</th>
<th>30-39</th>
<th>40-49</th>
<th>50-59</th>
<th>60-69</th>
<th>70-79</th>
<th>80-89</th>
<th>90-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>4.80</td>
<td>5.70</td>
<td>4.00</td>
<td>3.90</td>
<td>3.80</td>
<td>4.70</td>
<td>5.40</td>
<td>5.70</td>
<td>4.50</td>
<td>4.70</td>
</tr>
<tr>
<td>Median</td>
<td>1.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.50</td>
<td>4.00</td>
<td>6.00</td>
<td>4.00</td>
<td>4.50</td>
</tr>
<tr>
<td>Mode</td>
<td>0.00</td>
<td>3.00</td>
<td>2.00</td>
<td>5.00</td>
<td>4.00</td>
<td>2.00</td>
<td>4.00</td>
<td>9.00</td>
<td>2.00</td>
<td>4.00</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>7.66</td>
<td>4.72</td>
<td>1.70</td>
<td>1.66</td>
<td>1.81</td>
<td>2.06</td>
<td>3.72</td>
<td>2.63</td>
<td>2.22</td>
<td>1.64</td>
</tr>
<tr>
<td>Sample Variance</td>
<td>58.62</td>
<td>22.23</td>
<td>2.89</td>
<td>2.77</td>
<td>3.29</td>
<td>4.23</td>
<td>13.82</td>
<td>6.90</td>
<td>4.94</td>
<td>2.68</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>2.95</td>
<td>0.40</td>
<td>-1.78</td>
<td>-0.45</td>
<td>2.62</td>
<td>-1.08</td>
<td>5.61</td>
<td>-1.71</td>
<td>-0.83</td>
<td>-0.65</td>
</tr>
<tr>
<td>Skewness</td>
<td>1.80</td>
<td>1.18</td>
<td>0.00</td>
<td>0.38</td>
<td>1.35</td>
<td>0.12</td>
<td>2.20</td>
<td>-0.05</td>
<td>0.64</td>
<td>0.03</td>
</tr>
<tr>
<td>Range</td>
<td>23.00</td>
<td>14.00</td>
<td>4.00</td>
<td>5.00</td>
<td>6.00</td>
<td>6.00</td>
<td>13.00</td>
<td>7.00</td>
<td>6.00</td>
<td>5.00</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.00</td>
<td>0.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
</tr>
<tr>
<td>Maximum</td>
<td>23.00</td>
<td>14.00</td>
<td>6.00</td>
<td>7.00</td>
<td>8.00</td>
<td>8.00</td>
<td>15.00</td>
<td>9.00</td>
<td>8.00</td>
<td>7.00</td>
</tr>
</tbody>
</table>

Table 5-4: Descriptive Statistics for Response Time – Cropped Correct Identification

Table 5-4: shows higher variance in response time for lower screen utilisation percentages (utilisation percentage < 20%). The range value also supports this
observation. However, there are no strikingly obvious differences in the variance for the scaled case (see Table 5-5:).

<table>
<thead>
<tr>
<th>Screen Utilisation percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-9</td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>Median</td>
</tr>
<tr>
<td>Mode</td>
</tr>
<tr>
<td>Standard Deviation</td>
</tr>
<tr>
<td>Sample Variance</td>
</tr>
<tr>
<td>Kurtosis</td>
</tr>
<tr>
<td>Skewness</td>
</tr>
<tr>
<td>Range</td>
</tr>
<tr>
<td>Minimum</td>
</tr>
<tr>
<td>Maximum</td>
</tr>
</tbody>
</table>

Table 5-5: Descriptive Statistics for Response Time – Scaled Correct Identification

Graph 5-9: Screen utilisation percentage against response time for both cropped and scaled presentation styles for correct identification
At lower screen utilisation the scaled case shows a lower response time and tends to stabilise at screen utilisation above 30%. However, the cropped case reveals a much more stable response time across all screen utilisation (see Graph 5-9).

Response times for the unidentified (DK) case does not provide much useful information. This is dependent on the data presented and on the individual subjects. Some subjects many take some time to determine that they are unable to identify the document, whilst others may take little time and do not try as hard as the other subjects in the identification process. However, we would have expected to see lower response time at low screen utilisation percentages for the Scaled case. When Scaled, and the screen utilisation is low, subjects are unable to read any of the words hence they should in theory just give up and say "can not read the contents - unable to identify the document". As the screen size increased we would have expected them to make more effort to identify the document and should have taken longer before they reached a "don't know" decision. For the Cropped case, the response should have been relatively constant across the whole range of screen utilisation.
Chapter 5
Visual Recognition of Windows

Graph 5.11: Response time against screen utilisation percentage for both cropped and scaled presentation styles for incorrect identification case

The response time across screen utilisation percentage is very difficult to interpret. Response time is greatly dependent on the information presented. Screen size limits the amount of information presented and presentation styles also affect the way the information is presented. They influence the recognition performance. Whilst conducting this experiment subjects commented that they identified the document based on keywords, i.e. whether the document was talking about the right subject matter, and used the format of the document to re-enforce their decisions. Where subjects were unable to identify the document based on keywords, they appeared to use the format type to assist identification (i.e. font, character/line spacing, etc.). This was a rather speculative approach as they could not be entirely sure of a correct identification, unless the format was so distinct that it could clearly be identified.

In this experiment, the subjects were only asked to familiarise themselves with the document they were asked to identify. What would the result show if the subjects were given all the documents that were likely to be presented to them and asked to identify one of them? It is likely that they would show better performance both in terms of a shorter duration for identification and greater accuracy of identification.
The shorter duration is likely to be seen for presentations of document for which the subjects answer is "no, this is not the document asked to be identified".

5.7 Further Strategies for Improving Visual Cueing

We have found that the size of a window affects identification of a target document. The identification task becomes difficult at sizes below 30% of screen utilisation. If we want to further reduce the window size without compromising identification accuracy, we need new strategies for improving visual cueing. Three useful types of information can aid in the identification of a document. These are:

a) Diagrams and pictures
b) Document format (font, size, spacing, layout and so on)
c) Document contents in particular keywords that characterise the document

Diagrams and pictures tend to provide the best visual cues for identification. Subjects more easily remember diagrams contained in a document than words. Document format is usually difficult to recognise unless it is very distinctive. In addition, subjects identifying a document solely based on the format of a document can not be entirely sure that it is the target document. They still require additional information to make positive identification. Keywords can be used to identify a document with reasonable confidence, so long as unique keywords can be found (using the term 'keyword' to mean any word that uniquely identifies the document).

The number of unique keywords will decrease as the number of documents increases, since words that were previously unique to a document, may no longer be unique when two or more documents are compared. This is demonstrated in Figure 5-6 - it shows three different documents (1- a paper on Mobile Worker, 2- a paper on IT and Construction and 3- a report on Video Conferencing). There will be several keywords for each of the papers. Document 1 has keywords like PDA, Mobile,
Network and Computing. However, when document 1 and 3 are the possible choices then the keywords like Mobile, Network and Computing must be excluded as these are contained in both documents. They do not distinctly identify a specific document.

![Diagram showing visual cues provided by a document]

During the experiment subjects were asked to think aloud whilst making their decision about whether the document presented was the target document or not. Subjects quickly began to identify keywords belonging to the target document. They used these keywords to determine whether the document was the required document or not. In addition, they also began to identify keywords that were not part of the target document.

Concentrating the keywords might be used to improve the performance of users in identifying documents. This would enable us to further reduce the window size and yet still retain acceptable identification accuracy.
5.8 Summary of Results

The findings outlined in this chapter showed that higher recognition rates were achieved by subjects when using the cropped presentation style. As expected, lower correct identification of the document was observed at low screen utilisation percentage (i.e. window size as a percentage of the desktop size). That is, as the screen utilisation percentage increased so did the number of correct identifications. In addition, an accuracy level of about 60% was reached for 30% screen utilisation. Beyond this point, the accuracy seems to be very close for both presentation strategies.

A comparison was carried out between the means of number of correct identification for the cropped and scaled reduction strategies for < 30% screen utilisation using Student’s t-test. A significant (t(9)=4.34, p<0.001, 1-tail) difference was found between the two groups.

One early conclusion that can be made is that toolbars on applications which are inactive does not add value in the identification process. Rather, they can waste valuable workspace on inactive windows. This workspace could be best utilised by presenting more of the contents of the document. By removing the toolbars from inactive windows, the screen size is reduced further without compromising the accuracy of identification.

5.9 What Next?

One of the main questions that remain to be answered is “are there other techniques that can be employed to further reduce the screen sizes and still retain high levels of identification accuracy?”

The answer to this question is “yes” and techniques have been suggested by subjects commenting that they used keywords to identify the document. The next chapter explores these ideas and techniques. One of the hypotheses to be tested in the next chapter is that better performance can be achieved by identifying unique keywords
for a document. Pages containing several of these unique keywords should be easier to identify than those containing none or a few keywords. Thus, if partial views contain more of these keywords then even in a crowded or clutter screen it should be easier to identify the desired window. As a screen becomes crowded, the individual windows can adapt themselves to present maximum visual cues, even if the windows contain partial or obscured views.

A machine-readable dictionary is to be used to automatically identify keywords based on frequency of a particular word used in normal English language. The frequency of each word in a document is compared with the frequency in the dictionary for this word. If this frequency is higher in the document than that in the dictionary then this word is a keyword for this particular document.
6 ENHANCING VISUAL CUES BY TRANSFORMING INACTIVE WINDOW CONTENT

6.1 Overview
This chapter presents a technique for automatically extracting keywords (words that can be used to uniquely identify a document). Once these keywords have been extracted they can be used to adapt the document by presenting these keywords using a number of different techniques. Four such techniques are described in this chapter and an experiment carried out to determine which one of the four presentations of the keywords yields the highest recognition rate.

In addition, a subjective evaluation of the four presentation techniques was carried out to determine user preference for a particular style. Recognition becomes difficult at small window sizes. Therefore, the initial experiment in this chapter was further refined to examine details of recognition at smaller window sizes (less than 30% of the screen size).

The final experiment in this chapter investigates the effects of multiple documents in the search space on identification accuracy and response time for correct identification.

6.2 Introduction
During experiment 1 (see Chapter 5), subjects were asked to think aloud whilst making their decisions about whether the document presented was the target document or not. Subjects commented that the appearance of the text was often important in identifying the document. For example, they tended to remember the headings of sections, and the use of bold or italic fonts. They also observed that subjects rapidly identified keywords belonging to particular documents. They then
used these keywords to determine whether the observed document was the required
document or not. In addition, they also used key words that were not part of the
target document to reject candidate documents.

It was this observation which led to the suggestion that concentrating keywords in a
reduced window might improve the performance of users in identifying documents.
This would then allow the window size to be further reduced and yet still retain
acceptable identification accuracy.

A machine-readable dictionary was used to automatically identify key words (we
will now refer to these as keywords) in a document based on the frequency of use in
normal English. The frequency of each word in a document was compared with the
frequency in the dictionary for this word. If this frequency was higher in the
document than that in the dictionary then this word was a candidate to be labelled as
a keyword for this particular document.

6.3 A Technique for Automatic Extraction of Keywords

Keywords can be automatically identified in a document by comparing their use
with a dictionary that contains the frequency of words occurring in normal written
English language. One such dictionary is the MRC Psycholinguistic Database:
Machine Usable Dictionary. This database dictionary differs from other machine
usable dictionaries in that it includes not only syntactic information but also
psychological data for each entry (see Amsler, 1984 for a review of other machine-
readable dictionaries). It also differs from most conventional dictionaries in that it
does not currently attempt to provide any semantic information. The dictionary was
designed to be of use to:

- Psycholinguists in selecting stimulus materials for testing
- For use by researchers in Artificial Intelligence as a source of information
  required for natural language processing and cognitive simulation
• For use by computer scientists who wish to use the word lists and syntactic information in the design of text processors.

The dictionary was assembled by merging a number of smaller databases:

• The tape dictionary of Dolby, Resnikoff and MacMurray (1963) which was created by taking all the left justified bold faced words from the Shorter Oxford English Dictionary together with parts of speech given by that dictionary. In addition, words were taken from the Cornell University tape of 20,000 commonly used words, and the parts of speech for all these words found in the third edition of Webster's New International Dictionary.

• The Edinburgh Associative Thesaurus (Kiss, Armstrong, Milroy and Piper, 1973)

• The Colorado Norms (Toglia and Battig, 1978)

• The Pavio Norms (Pavio, Yuille and Madigan, 1968)

• The Gilhooly-Logie norms (Gilhooly and Logie, 1980)

• The Kucera-Francis written Frequency count (Kucera and Francis, 1967)

• The Thorndike-Lorge written frequency count (Thorndike and Lorge, 1944)

• The phonetic transcriptions from Daniel Jones Pronouncing Dictionary of English Language, 12th Edition (see Guierre, 1966)

• The frequency count for spoken English from the London Lund Corpus of English Conversation (Svartvik and Quirk, 1980; Brown, 1984)

The dictionary file occupies 11 Mbytes as a sequential Unix file. Each line of the file contains the field for one word. The longest entry is 130 characters. The last four properties are held in variable length fields separated by a |character. The structure of each entry is as follows:
The composition of the dictionary file is summarised in Table 6-1, which specifies the linguistic properties described in an entry. The first column indicates the columns/field in the file containing the data. The second column indicates the name of the data field. The third column specifies the identity of the linguistic property.

<table>
<thead>
<tr>
<th>Column</th>
<th>Name</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>NLET</td>
<td>Number of letters in the word</td>
</tr>
<tr>
<td>3-4</td>
<td>NPHON</td>
<td>Number of Phonemes in the word</td>
</tr>
<tr>
<td>5</td>
<td>NSYL</td>
<td>Number of syllables in the word</td>
</tr>
<tr>
<td>6-10</td>
<td>K-F-FREQ</td>
<td>Kucera and Francis written frequency</td>
</tr>
<tr>
<td>11-12</td>
<td>K-F_NCATS</td>
<td>Kucera and Francis number of categories</td>
</tr>
<tr>
<td>13-15</td>
<td>K-F-NSAMP</td>
<td>Kucera and Francis number of samples</td>
</tr>
<tr>
<td>16-21</td>
<td>T-L-FREQ</td>
<td>Thorndike-Lorge Frequency</td>
</tr>
<tr>
<td>22-25</td>
<td>BROWN-FREQ</td>
<td>Brown verbal frequency</td>
</tr>
<tr>
<td>26-28</td>
<td>FAM</td>
<td>Familiarity</td>
</tr>
<tr>
<td>29-31</td>
<td>CONC</td>
<td>Concreteness</td>
</tr>
<tr>
<td>32-34</td>
<td>IMAG</td>
<td>Imagery</td>
</tr>
<tr>
<td>35-37</td>
<td>MEANC</td>
<td>Mean Colorado Meaningfulness</td>
</tr>
<tr>
<td>38-40</td>
<td>MEANP</td>
<td>Mean Pavio Meaningfulness</td>
</tr>
<tr>
<td>41-43</td>
<td>AOA</td>
<td>Age of Acquisition</td>
</tr>
<tr>
<td>44</td>
<td>TQ2</td>
<td>Type</td>
</tr>
<tr>
<td>45</td>
<td>WTYPE</td>
<td>Part of Speech</td>
</tr>
<tr>
<td>46</td>
<td>PDWTYPE</td>
<td>PD Part of Speech</td>
</tr>
<tr>
<td>47</td>
<td>ALPHSYL</td>
<td>Alphasyllable</td>
</tr>
<tr>
<td>48</td>
<td>STATUS</td>
<td>Status</td>
</tr>
<tr>
<td>49</td>
<td>VAR</td>
<td>Variant Phoneme</td>
</tr>
<tr>
<td>50</td>
<td>CAP</td>
<td>Written Capitalised</td>
</tr>
<tr>
<td>51</td>
<td>IRREG</td>
<td>Irregular Plural</td>
</tr>
<tr>
<td>52</td>
<td>WORD</td>
<td>The actual word</td>
</tr>
<tr>
<td>53</td>
<td>PHON</td>
<td>Phonetic Transcription</td>
</tr>
<tr>
<td>54</td>
<td>DPHON</td>
<td>Edited Phonetic Transcription</td>
</tr>
<tr>
<td>55</td>
<td>STRESS</td>
<td>Stress Pattern</td>
</tr>
</tbody>
</table>

Table 6-1: Composition of the dictionary file
Chapter 6  
Enhanced Visual Cues

The primary data of interest is the Kucera-Francis written frequency count. For each word contained in a document the Kucera-Francis written frequency count (Expected Frequency) can be compared to the frequency of the word contained within the document (Observed Frequency).

6.3.1 The Process of Generating Keywords for a Document Automatically

Firstly, the original document is converted to a text-only document and then all the words in the document are extracted, and a count of the number of occurrence of each word made. This list is then sorted alphabetically. Using this sorted list, the expected number of occurrence of each word from the corpus (the Kucera-Francis written frequency count) is extracted. For both the expected and observed number of occurrences, the frequencies are calculated. For the expected frequency, this is the number of occurrence of the word within the corpus divided by a million (since the corpus contains a million words). For the observed frequency of each word in the document, this is simply the number the times the word occurs in the document divided by the total number of words in the document.

The difference between the expected and observed frequency is calculated based on the following formula:

\[
\text{Difference} = \frac{F_o - F_e}{F_e}
\]

*Equation 6-1: Difference between observed and expected frequencies for a word*

Where  
\( F_o = \) Observed Frequency  
\( F_e = \) Expected Frequency

The higher the difference the higher the probability that this word is a potential keyword for the document. Taking a few sample words, we can check to see if this holds true. For words like “the”, “is”, “and”, “then”, “where” and so on, the difference in expected and observed frequency should be low. However for our
target document words like "user", "graphical", and "mobile" should show a higher difference between the expected and observed frequencies.

<table>
<thead>
<tr>
<th>Word</th>
<th>Document 1</th>
<th></th>
<th>Document 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$F_O$</td>
<td>$F_E$</td>
<td>$\text{Diff}/F_E$</td>
<td>$F_O$</td>
</tr>
<tr>
<td>the</td>
<td>0.07686</td>
<td>0.06997</td>
<td>0.09845</td>
<td>0.05074</td>
</tr>
<tr>
<td>is</td>
<td>0.01471</td>
<td>0.01009</td>
<td>0.45736</td>
<td>0.02097</td>
</tr>
<tr>
<td>and</td>
<td>0.01676</td>
<td>0.02885</td>
<td>-0.4190</td>
<td>0.02097</td>
</tr>
<tr>
<td>then</td>
<td>0.00224</td>
<td>0.00137</td>
<td>0.63294</td>
<td>0.00451</td>
</tr>
<tr>
<td>where</td>
<td>0.00081</td>
<td>0.00093</td>
<td>-0.1282</td>
<td>0.00180</td>
</tr>
<tr>
<td>user</td>
<td>0.01042</td>
<td>0.000004</td>
<td>2605.29</td>
<td>NID</td>
</tr>
<tr>
<td>graphical</td>
<td>0.00143</td>
<td>0.000001</td>
<td>1429.90</td>
<td>NID</td>
</tr>
<tr>
<td>mobile</td>
<td>0.00858</td>
<td>0.00004</td>
<td>194.123</td>
<td>0.00067</td>
</tr>
<tr>
<td>scan</td>
<td>NID</td>
<td>NID</td>
<td>NID</td>
<td>NID</td>
</tr>
<tr>
<td>fuzzy</td>
<td>NID</td>
<td>NID</td>
<td>NID</td>
<td>0.005187</td>
</tr>
<tr>
<td>radar</td>
<td>NID</td>
<td>NID</td>
<td>NID</td>
<td>0.012404</td>
</tr>
<tr>
<td>computer</td>
<td>0.001839</td>
<td>0.000013</td>
<td>140.5183</td>
<td>0.000902</td>
</tr>
</tbody>
</table>

Table 6-2: Sample words and their associated frequencies

NID = Not In Document

It is clear from the sample of words, shown in Table 6-2, that the observed and expected frequencies of common words such as the, is, and, then, where are very close to each other. This is true for both documents, as we would expect. These words are common words used in the construction of sentences and are of no value in the identification task. The remaining words in the table show a substantial difference between the expected and observed frequencies, and are potential keywords. However, two words mobile and computer appear in both documents and the difference in frequency is greater in one document than the other document. If the two words are considered as being keywords for both documents and if both documents are compared, then these keywords will have to be excluded, as they will not uniquely identify an individual document. This indicates that only words with much higher value should be identified as keywords in a document.
6.4 Techniques for Displaying Keywords

As the inactive windows reduces in size, real estate becomes more scarce, and the number of observable keywords will decrease as the size decreases. To increase identification, we need to increase the ratio of visible keywords to visible non-keywords. This ratio can be increased using a number of techniques, three of which are discussed below. These are:

RETAIN ONLY KEYWORDS. All non-keywords are removed from the document, leaving only keywords. This results in a keyword to non-keyword ratio of infinity, i.e. only keywords remain.

These keywords can be displayed in one of two possible presentation styles.

1. LIST – Each of the keywords contained within a document is extracted and listed in a table. However, with this technique, all visual cues provided by the format (line spacing, font, and so on) of the document are lost. In addition, the relationships between words (the sequence of words) are also lost. These can be referred to as "positional cues". For example, mobile may be followed by computing, or wireless followed by network. All positional cues are lost.

![Figure 6-1: Presenting keywords in the form of a LIST](image)

2. COMPRESSED – This technique again removes all non-keywords from the document, but instead of removing the non-keywords, these are replaced by spaces. This retains some positional cues. Keywords followed by other keywords
appear together. However, the technique tends to leave large spaces between keywords and, as the window, size decreases, it is possible for the visible window portion to contain only spaces. A better technique will be to compress all spaces and replace them with three dots.

![Image of compressed spaces](image)

**Figure 6-2: Presenting keywords - COMPRESSED SPACES**

An alternative to the two techniques outlined above is to highlight the keywords in context and retain all other words. We call this “highlighted”.

3. **HIGHLIGHTED** - All keywords are highlighted within the document. This retains all formats and positional cues provided by the document. However, by highlighting the keywords, subjects attention is drawn to the highlighted words, as they tend to stand out of the document. This relates to the two levels of visual processing - entire and focused (Treisman, 1986). Certain aspects of visual processing seem to be accomplished simultaneously (that is, for the entire visual field at once) and automatically (that is, without attention being focused on any one part of the visual field). Other aspects of visual processing seem to depend on focused attention and are done serially, or one at a time.
The proverbial needle in a haystack is hard to find because it shares properties of length, thickness and orientation with the hay in which it is hidden. A red poppy in a haystack is a much easier target; its unique colour and shape are detected automatically.

Treisman (1986) reports that “if a target differs from the distracters in some simple property, such as colour or shape, the target is detected about equally fast in an array of 30 items and an array of three items. Such targets pop out of the display, so that the time it takes to find them is independent of the number of distracters.”

### 6.5 A Simple Method for Implementation

For each document, a secondary view is created, called the keyword view. The user view switches between the original document and the keyword view as the window is activated and deactivated.
This technique has a problem. For example, comparing information contained in two or more windows can be difficult. Since only one window can be the active window with its original content intact (i.e. no adaptation/transformation of the window content) and the other windows being compared with, will have adapted to the keyword view. Therefore users will only be able to compare the active window containing the original view to the inactive windows containing a keyword view.

However, this problem can be overcome by providing an adaptation override facility that will prevent the system from automatically manipulating the contents of a particular window.

6.6 Experiment 2: Adapting the Contents of the Document
An experimental study is required to determine which one of the four possible adaptation techniques yields greater recognition performance in terms of both accuracy and speed. The four possible techniques are highlighting keywords,
compressing keywords, listing keywords in a table format, or leaving as it is, i.e. normal.

The effect of window size and adapted presentation on recognition is to be studied. The experimental method should vary the screen size and presentations style. The time for the user to identify the document and the result of the identification is recorded.

The dependent variables for the experiment are therefore response time and result of identification. The identification result can be one of three outcomes; correctly identified; incorrectly identified; or don't know (the subject could not make an identification given the information presented).

Correct identification can result from two conditions. Firstly, if the document presented is the target document and the subject responds by saying “Yes – it is the document” this results to a correct identification. Secondly, if the document presented is not the target document and the subject responds by saying “No – this is not the target document” then this also results to a correct identification. All other combination results to an incorrect identification, except for the “don’t know” cases.

The independent variables are:

- Adapted keyword presentation techniques - four such techniques are being evaluated; highlighting keywords, compressing keywords, listing keywords and presenting the document as is without any adaptation.

- Screen utilisation percentage – this is the size of the window as a percentage of the total screen size.

6.6.1 Sample Size and Distribution

Two hundred trial screens were presented to each subject according to the distribution shown in Table 6-3. Trials were presented to subjects in a random sequence, hence minimising the effect of learning. Some learning of the document
cannot be avoided but can be minimised by presenting the trials randomly. Thus, the
effects of learning are distributed across all presentation techniques and screen
utilisation percentages.

<table>
<thead>
<tr>
<th>Presentation Technique</th>
<th>Screen Utilisation Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0-9</td>
</tr>
<tr>
<td>Compressed</td>
<td>5</td>
</tr>
<tr>
<td>Highlighted</td>
<td>5</td>
</tr>
<tr>
<td>List</td>
<td>5</td>
</tr>
<tr>
<td>Normal</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 6-3: Distribution of trials

6.7 Results of Analysis – Accuracy of Identification
Eighteen researchers and students from the department volunteered for the experiment. The subjects were required to have a computing or engineering background, as the document in question was from the computing field. Each subject was tested in all four conditions of the experiment. The descriptive statistics for the data are presented in appendix C.2.

6.7.1 Tests for Normality and Homogeneity of Variances
To perform an ANOVA (Analysis of Variance) two assumptions regarding the data must hold true.

- Each group is an independent random sample from a normal population. (test for normality)
- In the population, the variances of the groups are equal. (test for homogeneity of variance)

6.7.1.1 Test for Homogeneity of Variances
Numerous tests are available for evaluating the assumption that all groups come from populations with equal variances. Many of these tests, however, are heavily dependent on the data being from normal populations. Analysis of Variance
procedures, on the other hand, are reasonable robust against departures from normality (Norusis, 1993a).

The Levene test is a homogeneity of variance test that is less dependent on the assumption of normality than most tests and thus is particularly useful with analysis of variance. It is obtained by computing, for each case, the absolute difference from its cell mean and performing a one way analysis of variance on these differences.

A Levene test is used to test the null hypothesis that the groups come from populations with the same variance. If the observed significance level is small (<0.05), we can reject the null hypothesis that all variances are equal.

<table>
<thead>
<tr>
<th>Correct</th>
<th>Levene Statistic</th>
<th>df1</th>
<th>df2</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Based on Mean</td>
<td>.086</td>
<td>3</td>
<td>716</td>
<td>.968</td>
</tr>
<tr>
<td>Based on Median</td>
<td>.238</td>
<td>3</td>
<td>716</td>
<td>.870</td>
</tr>
<tr>
<td>Based on Median and with adjusted df</td>
<td>.238</td>
<td>3</td>
<td>691.959</td>
<td>.870</td>
</tr>
<tr>
<td>Based on trimmed mean</td>
<td>.275</td>
<td>3</td>
<td>716</td>
<td>.844</td>
</tr>
</tbody>
</table>

Table 6-4: Test of homogeneity of variance for number of correct identification across all screen sizes for individual presentation styles

It can be seen from Table 6-4 that the significance of the Levene statistic based on mean is 0.968 and is considered large. Therefore the null hypothesis is rejected. This means that we do not have sufficient evidence to suspect that the variances are unequal.

6.7.1.2 Test for Normality

Since the normal distribution is very important for correct statistical inference, it is necessary to examine the assumption that the data come from a normal distribution. One way to do this is with a normal probability plot. In a normal probability plot, each observed value is paired with its expected value from the normal distribution. The expected value from the normal distribution is based on the number of cases in the sample and rank order of the case in the sample. If the sample is from a normal distribution, the points can be expected to lie more or less on a straight line.
Although normal probability plots provide a visual basis for checking normality, it is often desirable to compute the statistical test of the hypothesis that the data are from a normal distribution. Two commonly used tests are the Shapiro-Wilks test and the Lilliefors test.

<table>
<thead>
<tr>
<th>Correct</th>
<th>comp</th>
<th>Statistic</th>
<th>df</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.292</td>
<td>180</td>
<td>0.000</td>
</tr>
<tr>
<td>high</td>
<td></td>
<td>0.341</td>
<td>180</td>
<td>0.000</td>
</tr>
<tr>
<td>list</td>
<td></td>
<td>0.333</td>
<td>180</td>
<td>0.000</td>
</tr>
<tr>
<td>norm</td>
<td></td>
<td>0.288</td>
<td>180</td>
<td>0.000</td>
</tr>
</tbody>
</table>

*Table 6-5: Tests of normality for number of correct identification*

If the observed significance level is small, the null hypothesis that the data come from a normal distribution can be rejected.

However, if the sample size is large, almost any goodness of fit test (like the Lilliefors) will result in rejection of the null hypothesis. It is almost impossible to find data that are exactly normally distributed. For most statistical tests, it is sufficient that the data are approximately normally distributed (Norusis, 1993a).

Although Lilliefors test of normality (Table 6-5) shows, that data is not from a normal distribution, the histogram shown below for each of the presentation styles show a normal distribution. Based on the histogram we can assume that the data is from a reasonably normal distribution.
Chapter 6  
Enhanced Visual Cues

Histogram  
For PSTYLE = comp

Histogram  
For PSTYLE = high

Histogram  
For PSTYLE = list

Histogram  
For PSTYLE = norm

<table>
<thead>
<tr>
<th>Angle</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>10</td>
</tr>
<tr>
<td>1.0</td>
<td>20</td>
</tr>
<tr>
<td>2.0</td>
<td>30</td>
</tr>
<tr>
<td>3.0</td>
<td>40</td>
</tr>
<tr>
<td>4.0</td>
<td>50</td>
</tr>
<tr>
<td>5.0</td>
<td>60</td>
</tr>
</tbody>
</table>

Correct

Histogram Details:
- Std. Dev = 1.18
- Mean = 4.2
- N = 180.00

Histogram Details:
- Std. Dev = 1.22
- Mean = 3.8
- N = 180.00

Histogram Details:
- Std. Dev = 1.24
- Mean = 3.8
- N = 180.00

Histogram Details:
- Std. Dev = 1.22
- Mean = 3.8
- N = 180.00
Shown above is the normal probability plot, a straight line through the plot indicates that the data is from a normal distribution. If the plotted data generally follows a normal distribution, the line is likely to be straight.

6.7.2 Target Document Identification Accuracy

The overall result show that the compressed presentation technique yields the greatest recognition rate (see Table 6-6). On average 82.8% of the samples were correctly identified. The remaining three presentation techniques are very close together in terms of the correct identification percentage. However, the incorrect and don’t know percentages vary between these three styles. The normal presentation shows the highest incorrect responses.
Therefore it can be concluded that improved identification performance can be achieved by adapting the presentation styles. Table 6-6 shows that compressed presentation does better than the other three presentation but is this statistically significant?

A repeat measure ANOVA shows that the mean number of correct identifications is significantly different between the different presentation styles; $F(3,537)=9.69$, $p<0.001$.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>SS</th>
<th>DF</th>
<th>MS</th>
<th>F</th>
<th>Sig. of F</th>
</tr>
</thead>
<tbody>
<tr>
<td>WITHIN+RESIDUAL</td>
<td>279.86</td>
<td>537</td>
<td>0.52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSTYLE</td>
<td>15.14</td>
<td>3</td>
<td>5.05</td>
<td>9.69</td>
<td>.000</td>
</tr>
</tbody>
</table>

The ANOVA shown in Table 6-7 only indicates that the overall results are significantly different from each other. Hence, further statistical tests are required to see if individual presentations are significantly different from each other. This comparison need only be made with the normal presentation since this is what we are trying to improve on. A two tailed t-test can be used to measure if the means between the samples are significantly different from each other and the direction of the difference.

The hypothesis states that there is no difference in the number of correct identifications between Compressed & Normal, Highlighted & Normal and List & Normal.
Table 6-8: Paired two-tailed t-test for identification accuracy

Table 6-8 shows paired two tail t-tests comparing the means of compressed, highlighted, and list against the normal presentations. For the Comp-Norm pair there is sufficient evidence ($t(179) = 4.445$, $p < 0.001$, 2-tail) to reject the null that there is no difference between the means. However, there is no statistical evidence to reject the null hypothesis for the High-Norm ($t(179) = -1.145$, $p = 0.254$, 2-tail) and List-Norm ($t(179) = -0.302$, $p = 0.763$, 2-tail) cases.

6.7.2.1 Detailed Analyse of Screen Utilisation and Identification Accuracy

Graph 6-1 shows how the identification accuracy varies as the screen utilisation percentage increases for the four presentation techniques. Also shown on the graph is the standard deviation marked by the vertical bar.
A two-way analysis of variance is required to assess the influence of two independent variables (Screen utilisation groupings and Presentation styles). Since there are 10 screen utilisation groups and 4 presentation styles, a 10 x 4 Repeated Measures ANOVA is performed.

The analysis of variance summary table (see Table 6-9) shows that:

- Screen utilisation group has a significant influence on the number of correct identifications.
- The presentation styles have a significant influence on the number of correct identifications.
- In addition, the interaction between screen utilisation and presentation styles has a significant influence on the number of correct identifications.
Chapter 6 Enhanced Visual Cues

Table 6-9: 10x4 Repeated Measures ANOVA table for number of correct identifications

A repeated measure ANOVA for each individual screen utilisation group is shown in Table 6-10. The most surprising result is that no significance difference was observed for screen utilisation groupings of 10-19% and 20-29%. In addition, the repeated measures ANOVA for above 50% utilisation shows highly significant differences in the sample means. The hypothesis being tested is that there is no difference between the average number of correct identifications.

Table 6-10: Repeated Measures ANOVA for identification accuracy for all screen utilisation groups

ANOVA only identifies whether there are statistically significant differences between the samples but does not identify which samples are different from each other. Further tests can be employed to identify the individual differences. For the set of data outlined
in this experiment, a paired sample t-test can be used to compare the means of two samples for significant differences, as shown in Table 6-11. All samples are compared to the normal presentation styles. The highlights shown results that are significant at $p < 0.05\%$. 
<table>
<thead>
<tr>
<th>Paired Differences</th>
<th>Mean</th>
<th>Std. Dev</th>
<th>Std. Error Mean</th>
<th>95% Confidence Interval of the Difference</th>
<th>t</th>
<th>df</th>
<th>Sig. (2-tailed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-9%</td>
<td>Comp-Norm</td>
<td>.44</td>
<td>1.10</td>
<td>-.26</td>
<td>.48</td>
<td>1.17</td>
<td>.99</td>
</tr>
<tr>
<td></td>
<td>High-Norm</td>
<td>-.39</td>
<td>1.14</td>
<td>.27</td>
<td>.18</td>
<td>1.44</td>
<td>.18</td>
</tr>
<tr>
<td></td>
<td>List-Norm</td>
<td>-.17</td>
<td>1.29</td>
<td>.31</td>
<td>.48</td>
<td>1.19</td>
<td>.19</td>
</tr>
<tr>
<td>10-19%</td>
<td>Comp-Norm</td>
<td>.44</td>
<td>1.58</td>
<td>.37</td>
<td>-.34</td>
<td>1.23</td>
<td>1.193</td>
</tr>
<tr>
<td></td>
<td>High-Norm</td>
<td>.61</td>
<td>1.75</td>
<td>.41</td>
<td>-.26</td>
<td>1.48</td>
<td>1.479</td>
</tr>
<tr>
<td></td>
<td>List-Norm</td>
<td>.44</td>
<td>1.62</td>
<td>.38</td>
<td>-.36</td>
<td>1.25</td>
<td>1.166</td>
</tr>
<tr>
<td>20-29%</td>
<td>Comp-Norm</td>
<td>.22</td>
<td>.88</td>
<td>.21</td>
<td>-.21</td>
<td>.66</td>
<td>1.074</td>
</tr>
<tr>
<td></td>
<td>High-Norm</td>
<td>.11</td>
<td>.58</td>
<td>.14</td>
<td>-.18</td>
<td>.40</td>
<td>.809</td>
</tr>
<tr>
<td></td>
<td>List-Norm</td>
<td>-.17</td>
<td>.86</td>
<td>.20</td>
<td>-.59</td>
<td>.26</td>
<td>.825</td>
</tr>
<tr>
<td>30-39%</td>
<td>Comp-Norm</td>
<td>-.67</td>
<td>.84</td>
<td>.20</td>
<td>-.108</td>
<td>-.25</td>
<td>-3.367</td>
</tr>
<tr>
<td></td>
<td>High-Norm</td>
<td>-.50</td>
<td>.79</td>
<td>.19</td>
<td>-.89</td>
<td>-.11</td>
<td>2.699</td>
</tr>
<tr>
<td></td>
<td>List-Norm</td>
<td>-.83</td>
<td>1.10</td>
<td>.26</td>
<td>-.138</td>
<td>-.29</td>
<td>-3.220</td>
</tr>
<tr>
<td>40-49%</td>
<td>Comp-Norm</td>
<td>.28</td>
<td>.83</td>
<td>.19</td>
<td>-.13</td>
<td>.69</td>
<td>1.426</td>
</tr>
<tr>
<td></td>
<td>High-Norm</td>
<td>.11</td>
<td>.68</td>
<td>.16</td>
<td>-.23</td>
<td>.45</td>
<td>.697</td>
</tr>
<tr>
<td></td>
<td>List-Norm</td>
<td>-.33</td>
<td>.77</td>
<td>.18</td>
<td>-.71</td>
<td>4.81E-02</td>
<td>-1.844</td>
</tr>
<tr>
<td>50-59%</td>
<td>Comp-Norm</td>
<td>.78</td>
<td>.81</td>
<td>.19</td>
<td>.38</td>
<td>1.18</td>
<td>4.082</td>
</tr>
<tr>
<td></td>
<td>High-Norm</td>
<td>.72</td>
<td>.89</td>
<td>.21</td>
<td>.28</td>
<td>1.17</td>
<td>3.424</td>
</tr>
<tr>
<td></td>
<td>List-Norm</td>
<td>.28</td>
<td>.67</td>
<td>.16</td>
<td>-.50E-02</td>
<td>.61</td>
<td>1.761</td>
</tr>
<tr>
<td>60-69%</td>
<td>Comp-Norm</td>
<td>.11</td>
<td>.58</td>
<td>.14</td>
<td>-.18</td>
<td>.40</td>
<td>.809</td>
</tr>
<tr>
<td></td>
<td>High-Norm</td>
<td>-.67</td>
<td>.69</td>
<td>.16</td>
<td>-.101</td>
<td>-.33</td>
<td>-4.123</td>
</tr>
<tr>
<td></td>
<td>List-Norm</td>
<td>.22</td>
<td>.65</td>
<td>.15</td>
<td>-.94E-02</td>
<td>.54</td>
<td>1.458</td>
</tr>
<tr>
<td>70-79%</td>
<td>Comp-Norm</td>
<td>.22</td>
<td>.55</td>
<td>.13</td>
<td>-.50E-02</td>
<td>.49</td>
<td>1.719</td>
</tr>
<tr>
<td></td>
<td>High-Norm</td>
<td>-.89</td>
<td>1.28</td>
<td>.30</td>
<td>-.152</td>
<td>-.25</td>
<td>-2.950</td>
</tr>
<tr>
<td></td>
<td>List-Norm</td>
<td>.00</td>
<td>.69</td>
<td>.16</td>
<td>-.34</td>
<td>.34</td>
<td>.000</td>
</tr>
<tr>
<td>80-89%</td>
<td>Comp-Norm</td>
<td>1.11</td>
<td>.58</td>
<td>.14</td>
<td>.82</td>
<td>1.40</td>
<td>8.086</td>
</tr>
<tr>
<td></td>
<td>High-Norm</td>
<td>.44</td>
<td>.62</td>
<td>.15</td>
<td>.14</td>
<td>.75</td>
<td>3.063</td>
</tr>
<tr>
<td></td>
<td>List-Norm</td>
<td>5.56E-02</td>
<td>.42</td>
<td>9.81E-02</td>
<td>.15</td>
<td>.26</td>
<td>5.566</td>
</tr>
<tr>
<td>90-100%</td>
<td>Comp-Norm</td>
<td>.28</td>
<td>.67</td>
<td>.16</td>
<td>-.50E-02</td>
<td>.61</td>
<td>1.761</td>
</tr>
<tr>
<td></td>
<td>High-Norm</td>
<td>-.50</td>
<td>.86</td>
<td>.20</td>
<td>-.93</td>
<td>7.36E-02</td>
<td>-2.474</td>
</tr>
<tr>
<td></td>
<td>List-Norm</td>
<td>.28</td>
<td>.75</td>
<td>.18</td>
<td>-.96E-02</td>
<td>.65</td>
<td>1.567</td>
</tr>
</tbody>
</table>

Table 6-11: Paired two-tail t-test for identification accuracy for all screen utilisation groups
6.7.3 Did the Data Set Effect the Result?

The results could be affected by the actual data selected for the documents. Although these were selected randomly, there is a slight chance that a particular trial screen maybe easier to identify than another screen because of the data contained within it. To minimise this effect and to test if the data set affected the results, two sets of data were used on subjects. Ten subjects used data set 1 and eight subjects used data set 2.

A two-way analysis of variance is required to assess the influence of two independent variables (Presentation styles and the data set). Since there are 4 presentation styles and there are 2 sets of data, a 4 x 2 Repeated Measures ANOVA is performed.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>SS</th>
<th>DF</th>
<th>MS</th>
<th>F</th>
<th>Sig. of F</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main Effects:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Set</td>
<td>6.006</td>
<td>1</td>
<td>6.006</td>
<td>4.745</td>
<td>0.032</td>
</tr>
<tr>
<td>Presentation Styles</td>
<td>14.319</td>
<td>3</td>
<td>4.773</td>
<td>9.258</td>
<td>0.000</td>
</tr>
<tr>
<td><strong>Error (Data Set)</strong></td>
<td>99.994</td>
<td>79</td>
<td>1.266</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Error (Presentation Styles)</strong></td>
<td>122.181</td>
<td>237</td>
<td>0.516</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Interaction:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Data Set * Presentation Styles)</td>
<td>4.469</td>
<td>3</td>
<td>1.490</td>
<td>3.110</td>
<td>0.027</td>
</tr>
<tr>
<td><strong>Error (Data Set * Presentation Styles)</strong></td>
<td>113.531</td>
<td>237</td>
<td>0.479</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6-12: 4x2 Repeated Measures ANOVA table for number of correct identifications for presentation styles and data set

The analysis of variance summary table (see Table 6-12) shows that the presentation styles has a significant influence on the number of correct identification at p < 0.001 level of significance. In contrast, the data set has a significant influence on the number of correct identification at p < 0.05 level of significance. Similarly, the interaction between data set and presentation styles has a significant influence on the number of correct identification at p < 0.05 level of significance.

This suggests that the identification of a target document is highly influenced by the presentation style and that the data set also influenced the identification of target document. However, the influence of the data set on the recognition of a target document was lower than that of the influence of presentation style.
6.7.4 Subjective Evaluation

6.7.4.1 Ranking

Subjects were asked to rank the four presentation styles (compressed, highlighted, list and normal) in the order in which they found it easiest to identify the target document.

<table>
<thead>
<tr>
<th>Style</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highlighted</td>
<td>1 - easiest</td>
</tr>
<tr>
<td>List</td>
<td>2</td>
</tr>
<tr>
<td>Compressed</td>
<td>3</td>
</tr>
<tr>
<td>Normal</td>
<td>4 - hardest</td>
</tr>
</tbody>
</table>

*Table 6-13: User ranking of presentation styles*

In total, 16 out of the 18 subjects ranked the presentation styles. An overall ranking made by the subject was determined by calculating the mode for each presentation style (see Table 6-13). Most of the subjects ranked *highlighted* as the easiest presentation style for identifying the document, and the *normal* presentation style as being the most difficult. However, the earlier objective evaluation conducted indicated that subjects performed better using the compressed presentation style. This shows that what subjects “like” and how they perform does not always agree. The preferences were more biased towards the highlighted condition because users were able to obtain a global view of the document and, if they required an overview of the document, they could quickly scan the highlighted words. They could, themselves, filter between the two views as required.

6.7.4.2 Subjects Comments on Ranking Order

Most subjects stated that the list presentation style provided less layout cues. The highlighted and normal presentations provided good overviews of the document and its layout. However, with the highlighted presentation style, it was easier to quickly scan through the keywords than for normal presentation. The compressed presentation style had mixed comments, some liked it but others did not. All saw that it had potential.
One subject commented that perhaps the compressed presentation style could have some keywords highlighted. For example, keywords that are deemed strong keywords should be highlighted within the compressed presentation style.

Another subject commented on the fact that all highlighting had the same priority. Perhaps there should be different priorities for different keywords.

Most subjects agreed that the normal presentation style was the worst presentation style for identifying a document. Using the normal presentation style, it took longer to read through the document and identify words that were familiar.

6.8 Results of Analysis - Response Times

Descriptive statistics for response times for correct identification of target document are presented in appendix C.3.

6.8.1 Test for Homogeneity of Variances

A Levene test is used to test the null hypothesis that the groups come from populations with the same variance. If the observed significance level is small (<0.05), we can reject the null hypothesis that all variances are equal.

<table>
<thead>
<tr>
<th></th>
<th>Levene Statistic</th>
<th>df1</th>
<th>df2</th>
<th>Sig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>Based on Mean</td>
<td>1.843</td>
<td>3</td>
<td>697</td>
</tr>
<tr>
<td>Correct</td>
<td>Based on Mean</td>
<td>1.843</td>
<td>3</td>
<td>697</td>
</tr>
<tr>
<td>Correct</td>
<td>Based on Median</td>
<td>1.473</td>
<td>3</td>
<td>697</td>
</tr>
<tr>
<td>Correct</td>
<td>Based on Median</td>
<td>1.473</td>
<td>3</td>
<td>697</td>
</tr>
<tr>
<td>Based on Median and with adjusted df</td>
<td>1.473</td>
<td>3</td>
<td>639.716</td>
<td>.221</td>
</tr>
<tr>
<td>Based on Median and with adjusted df</td>
<td>1.473</td>
<td>3</td>
<td>639.716</td>
<td>.221</td>
</tr>
<tr>
<td>Based on trimmed mean</td>
<td>1.617</td>
<td>3</td>
<td>697</td>
<td>.184</td>
</tr>
<tr>
<td>Based on trimmed mean</td>
<td>1.617</td>
<td>3</td>
<td>697</td>
<td>.184</td>
</tr>
</tbody>
</table>

Table 6-14: Test of homogeneity of variance for response time across all screen sizes for individual presentation styles

It can be seen from Table 6-14 that the significance of the Levene statistic based on mean is 0.138 and is considered large. We can not therefore reject the null hypothesis. This means that we do not have sufficient evidence to suspect that the variances are unequal.
6.8.2 Test for Normality

Table 6-15 shows the Lilliefors test for normality. If the observed significance level is small (<0.05) then the null hypothesis that the data is from a normal distribution can be rejected. For the response time data, the null hypothesis is rejected.

<table>
<thead>
<tr>
<th></th>
<th>Kolmogorov-Smirnov*</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Statistic</td>
<td>df</td>
<td>Sig.</td>
</tr>
<tr>
<td>Correct Compressed</td>
<td>0.131</td>
<td>169</td>
<td>0.000</td>
</tr>
<tr>
<td>Correct Compressed</td>
<td>0.131</td>
<td>169</td>
<td>0.000</td>
</tr>
<tr>
<td>Highlighted</td>
<td>0.141</td>
<td>169</td>
<td>0.000</td>
</tr>
<tr>
<td>Highlighted</td>
<td>0.141</td>
<td>169</td>
<td>0.000</td>
</tr>
<tr>
<td>List</td>
<td>0.092</td>
<td>169</td>
<td>0.001</td>
</tr>
<tr>
<td>List</td>
<td>0.092</td>
<td>169</td>
<td>0.001</td>
</tr>
<tr>
<td>Normal</td>
<td>0.126</td>
<td>169</td>
<td>0.000</td>
</tr>
<tr>
<td>Normal</td>
<td>0.126</td>
<td>169</td>
<td>0.000</td>
</tr>
</tbody>
</table>

a. Lilliefors Significance Correction

Table 6-15: Tests of normality for response time for correct identification

However, if the sample size is large, almost any goodness of fit test (like the Lilliefors) will result in rejection of the null hypothesis. It is almost impossible to find data that are exactly normally distributed. For most statistical tests, it is sufficient that the data are approximately normally distributed (Norusis, 1993a).

Although Lilliefors test of normality (Table 6-15) showed, that data is not from a normal distribution, the histogram shown below for each of the presentation styles show a normal distribution.
Figure 6-5: Response Ttime histograms with normal curve for Compressed, Highlighted, List and Normal presentation styles

Shown below is the normal probability plot, if the plotted data generally follow a normal distribution, the line is likely to be straight.
Chapter 6  
Enhanced Visual Cues

6.8.3  Response Time – Overall Analysis

The average response time for Correct, Incorrect and Don’t Know conditions are shown in Table 6-16 for all screen utilisation groupings. The highlighted presentation showed the best performance in terms of response time (on average it took 3.78 seconds for a correct result).

<table>
<thead>
<tr>
<th></th>
<th>Correct (s)</th>
<th>Incorrect (s)</th>
<th>Don’t Know (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compressed</td>
<td>4.14</td>
<td>0.55</td>
<td>0.31</td>
</tr>
<tr>
<td>Highlighted</td>
<td>3.78</td>
<td>1.09</td>
<td>0.14</td>
</tr>
<tr>
<td>List</td>
<td>3.82</td>
<td>0.91</td>
<td>0.27</td>
</tr>
<tr>
<td>Normal</td>
<td>3.81</td>
<td>0.82</td>
<td>0.37</td>
</tr>
</tbody>
</table>

Table 6-16: Average response time for correct, incorrect and don’t know answers for the four presentation techniques

Figure 6-6: Normal probability plots for response time for correct identification
Table 6-16 shows that the highlighted presentation style does better than the other three presentations in terms of response time but is this statistically significant?

A repeat measure ANOVA confirms that the mean response times for correct identification is significantly different between the different presentation styles; $F(3,504)=4.45$, $p<0.005$.

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>SS</th>
<th>DF</th>
<th>MS</th>
<th>F</th>
<th>Sig. of F</th>
</tr>
</thead>
<tbody>
<tr>
<td>WITHIN+RESIDUAL</td>
<td>2894.49</td>
<td>504</td>
<td>5.74</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSTYLE</td>
<td>76.60</td>
<td>3</td>
<td>25.53</td>
<td>4.45</td>
<td>.004</td>
</tr>
</tbody>
</table>

*Table 6-17: ANOVA table for response times for correct identification for presentation styles*

The ANOVA tables above only show that there is sufficient evidence that the response times are different between the presentation styles. However, to determine if there is a statistical difference between one of the adapted presentation styles (Compressed, Highlighted, and List) and the Normal presentation style a two tailed t-test between pairs of styles can be performed (as shown in Table 6-18).

<table>
<thead>
<tr>
<th>Paired Differences</th>
<th>95% Confidence Interval of the Difference</th>
<th>t</th>
<th>df</th>
<th>Sig. (2-tailed)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std. Error Mean</td>
<td>Lower</td>
<td>Upper</td>
</tr>
<tr>
<td>Comp - Norm</td>
<td>-0.6819</td>
<td>3.26</td>
<td>0.25</td>
<td>-2.27</td>
</tr>
<tr>
<td>High - Norm</td>
<td>-7.82E-02</td>
<td>3.80</td>
<td>0.29</td>
<td>-0.65</td>
</tr>
<tr>
<td>List - Norm</td>
<td>-0.7289</td>
<td>3.10</td>
<td>0.24</td>
<td>-1.20</td>
</tr>
</tbody>
</table>

*Table 6-18: Paired two-tail t-test for response times*

The mean response time for the compressed presentation styles compared to the normal was statistically significantly different ($t(172) = -2.76$, $p<0.01$, 2-tail), however the difference was in favour of the normal presentation style.

A similar observation was made for the list – normal presentation styles ($t(171) = -3.09$, $p<0.01$, 2-tail test). For the highlighted – normal case no statistically significant difference between the mean response time was observed ($t(172) = -0.27$, $p=0.787$, 2-tail).
6.8.4  **Response Time – Detailed Analysis**

Graph 6-2 shows the average response time for each of the presentation styles for each screen utilisation grouping. Also shown on the graph is the standard deviation marked by the vertical bars.

**Average (Mean) Response Time for Correct Identification**

![Graph 6-2: Average response time for correct identification against screen utilisation percentage](image)

Graph 6-2: Average response time for correct identification against screen utilisation percentage

Again, a two-way analysis of variance is required to assess the influence of two independent variables (Screen utilisation groupings and Presentation styles). The analysis of variance summary table (see Table 6-19) shows that screen utilisation grouping has a significant influence on the response time at the 5% level ($p < 0.05$). However, interaction between screen utilisation and presentation styles had no significant influences on the response time.
Chapter 6

Enhanced Visual Cues

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>SS</th>
<th>DF</th>
<th>MS</th>
<th>F</th>
<th>Sig. of F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Effects:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Screen Utilisation</td>
<td>107.564</td>
<td>9</td>
<td>11.952</td>
<td>2.636</td>
<td>0.013</td>
</tr>
<tr>
<td>Presentation Styles</td>
<td>21.035</td>
<td>3</td>
<td>7.012</td>
<td>0.997</td>
<td>0.417</td>
</tr>
<tr>
<td>Error (Screen Utilisation)</td>
<td>244.812</td>
<td>54</td>
<td>4.534</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error (Presentation Styles)</td>
<td>126.642</td>
<td>18</td>
<td>7.036</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Screen Utilisation * Presentation Styles)</td>
<td>177.447</td>
<td>27</td>
<td>6.572</td>
<td>1.071</td>
<td>0.380</td>
</tr>
<tr>
<td>Error (Screen Utilisation * Presentation Styles)</td>
<td>993.841</td>
<td>162</td>
<td>6.135</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6-19: 10x4 Repeated Measures ANOVA table for response time for correct Identifications

A repeated measure ANOVA for each screen utilisation group was performed to determine whether there were any significant differences between the presentation styles within each screen utilisation groups (see Table 6-20). The analysis shows that a statistically significant difference between the styles was observed for the 30-39% and 70-79% screen utilisation groups at \( p < 0.05 \) level of significance.

<table>
<thead>
<tr>
<th>Source PStyle</th>
<th>df</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F-test</th>
<th>P Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>00-09%</td>
<td>3</td>
<td>44.41</td>
<td>14.81</td>
<td>.675</td>
<td>0.579</td>
</tr>
<tr>
<td>10-19%</td>
<td>3</td>
<td>44.41</td>
<td>14.81</td>
<td>.675</td>
<td>0.579</td>
</tr>
<tr>
<td>20-29%</td>
<td>3</td>
<td>17.20</td>
<td>5.74</td>
<td>.790</td>
<td>0.505</td>
</tr>
<tr>
<td>30-39%</td>
<td>3</td>
<td>40.88</td>
<td>13.63</td>
<td>4.650</td>
<td>0.006</td>
</tr>
<tr>
<td>40-49%</td>
<td>3</td>
<td>4.74</td>
<td>1.58</td>
<td>.281</td>
<td>0.839</td>
</tr>
<tr>
<td>50-59%</td>
<td>3</td>
<td>42.73</td>
<td>14.25</td>
<td>2.501</td>
<td>0.070</td>
</tr>
<tr>
<td>60-69%</td>
<td>3</td>
<td>31.08</td>
<td>10.36</td>
<td>2.217</td>
<td>0.097</td>
</tr>
<tr>
<td>70-79%</td>
<td>3</td>
<td>46.27</td>
<td>15.42</td>
<td>2.993</td>
<td>0.039</td>
</tr>
<tr>
<td>80-89%</td>
<td>3</td>
<td>7.44</td>
<td>2.48</td>
<td>0.466</td>
<td>0.708</td>
</tr>
<tr>
<td>90-100%</td>
<td>3</td>
<td>34.43</td>
<td>11.48</td>
<td>2.68</td>
<td>0.057</td>
</tr>
</tbody>
</table>

Table 6-20: Repeated Measures ANOVA for response time for all screen utilisation groups

The result does support the fact that no significant difference between the presentation styles at higher screen utilisation group is likely to be observed. This is because as the screen utilisation increases the identification accuracy and response time should stabilise. However, it was expected that some statistically significant
differences for screen utilisation of less than 30% would have been observed. This was not the case.

The result of the t-test (see Table 6-21) indicates that there is no statistically significant difference between the means to indicate that one presentation style performs better than the other in terms of response time for each screen utilisation group. However, the overall result (i.e. across all screen utilisation) does show a significant result (see section 6.8.3). Further investigation is required to determine the effects of adapted presentation at 30% and lower screen utilisation.
<table>
<thead>
<tr>
<th>Comp - Norm</th>
<th>High - Norm</th>
<th>List - Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Std. Dev</td>
<td>Mean</td>
</tr>
<tr>
<td>0-9%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1.4109</td>
<td>3.7234</td>
<td>1.1227</td>
</tr>
<tr>
<td>.8964</td>
<td>8.3331</td>
<td>2.5125</td>
</tr>
<tr>
<td>-.1630</td>
<td>3.2737</td>
<td>1.0352</td>
</tr>
<tr>
<td>10-19%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-.0461</td>
<td>4.8125</td>
<td>1.1343</td>
</tr>
<tr>
<td>1.2794</td>
<td>4.5416</td>
<td>1.0725</td>
</tr>
<tr>
<td>-.9500</td>
<td>3.8948</td>
<td>.9180</td>
</tr>
<tr>
<td>20-29%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-.2344</td>
<td>2.9663</td>
<td>.6992</td>
</tr>
<tr>
<td>8.7778E-02</td>
<td>2.5877</td>
<td>.6099</td>
</tr>
<tr>
<td>.4483</td>
<td>3.3946</td>
<td>.8001</td>
</tr>
<tr>
<td>30-39%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-.2889</td>
<td>2.1881</td>
<td>.5157</td>
</tr>
<tr>
<td>1.2089</td>
<td>3.2344</td>
<td>.7624</td>
</tr>
<tr>
<td>-.8528</td>
<td>2.1886</td>
<td>.5159</td>
</tr>
<tr>
<td>40-49%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-.2844</td>
<td>3.6495</td>
<td>.8602</td>
</tr>
<tr>
<td>-.7094</td>
<td>2.6615</td>
<td>.6273</td>
</tr>
<tr>
<td>-.4350</td>
<td>3.0785</td>
<td>.7256</td>
</tr>
<tr>
<td>50-59%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1.0394</td>
<td>2.7630</td>
<td>.6513</td>
</tr>
<tr>
<td>2.206</td>
<td>3.8279</td>
<td>.9022</td>
</tr>
<tr>
<td>-.6722</td>
<td>2.7772</td>
<td>.6546</td>
</tr>
<tr>
<td>60-69%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-.4167</td>
<td>3.1849</td>
<td>.7507</td>
</tr>
<tr>
<td>-.1356</td>
<td>2.2112</td>
<td>.5212</td>
</tr>
<tr>
<td>.3028</td>
<td>2.8112</td>
<td>.6626</td>
</tr>
<tr>
<td>70-79%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1.3233</td>
<td>3.5895</td>
<td>.8460</td>
</tr>
<tr>
<td>.3289</td>
<td>3.8616</td>
<td>.9102</td>
</tr>
<tr>
<td>-.5094</td>
<td>2.6807</td>
<td>.6319</td>
</tr>
<tr>
<td>80-89%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.8000</td>
<td>2.3654</td>
<td>.5575</td>
</tr>
<tr>
<td>.6211</td>
<td>2.8520</td>
<td>.6722</td>
</tr>
<tr>
<td>.1894</td>
<td>3.5832</td>
<td>.8446</td>
</tr>
<tr>
<td>90-100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-.3583</td>
<td>2.8661</td>
<td>.6756</td>
</tr>
<tr>
<td>-.4667</td>
<td>3.2973</td>
<td>.7772</td>
</tr>
<tr>
<td>-.8250</td>
<td>2.7689</td>
<td>.6526</td>
</tr>
</tbody>
</table>

Table 6-21: Paired two-tail t-test for response time for all screen utilisation groups
6.9 Experiment 2B: Screen Utilisation of 0-30% - Step Size of 3
The results of Experiment 2 showed that above 30% screen utilisation, on average all presentation accuracy measures of identification were high and were beginning to converge to approximately 80% correct identification. Further detailed results are required at the lower screen utilisation as this is where we wish to maximise recognition rates.

6.9.1 Experimental Design and Conditions
The experimental design is similar to that described in section 6.6 but in this experiment the screen utilisation groups are between 0-30% with a step size of 3. Again, 200 sample screens were presented to each subject.

<table>
<thead>
<tr>
<th>Presentation Technique</th>
<th>Screen Utilisation Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Compressed</td>
<td>5</td>
</tr>
<tr>
<td>Highlighted</td>
<td>5</td>
</tr>
<tr>
<td>List</td>
<td>5</td>
</tr>
<tr>
<td>Normal</td>
<td>5</td>
</tr>
</tbody>
</table>

*Table 6-22: Distribution of trials for experiment 2B*

A total of six subjects took part in this experiment and the average number of correct identifications for each presentation style for each screen utilisation percentage (in steps of three) are plotted as shown in Graph 6-3.
Since there are 5 samples in each presentation group, then five correct answers result in 100% identification accuracy. If 60% accuracy is used as a standard measure to compare amongst the four different presentation styles then the compressed presentation style reaches this level at a screen utilisation percentage of 6 percent. In contrast, the three remaining styles achieve this level of accuracy at a screen utilisation percentage of 9 percent (see Graph 6-3). Therefore, higher identification can be achieved using the compressed presentation styles at lower screen utilisation percentages.

6.10 Experiment 2C: Identification of Target within a Multiple Document Search Space

This experiment was primarily designed to investigate the effect of increasing the number for windows contained in the search space on recognition rate.
6.10.1 Experimental Design

Experiment 2C was a continuation of experiment 2A. Subjects were asked to identify the same target document as in experiment 2A. However, the search space contained multiple documents from which the target document had to be identified. There were 2, 3, or 4 documents within the search space, one of which is always the target document. These documents were tiled so as not to overlap.

When two windows are displayed on the screen each window, is allowed to vary between 0 to 50% of the screen size, so that no overlap occurs. For 3 and 4 windows, the size is allowed to vary between 0-20% of the screen size. The experiment consisted of 180 trials in total. For each trial 1 of 3 complexity levels (i.e. 2, 3, or 4 windows) was selected automatically. There are 4 presentation styles (see Table 6-23). Five trials were presented for each presentation style and screen utilisation group.

The system recorded the length of time for identification and whether the identification was correct.

<table>
<thead>
<tr>
<th>Complexity (No. of Win)</th>
<th>Presentation Styles</th>
<th>Screen Utilisation Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0-9</td>
</tr>
<tr>
<td>2</td>
<td>Compressed</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Highlighted</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>List</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Compressed</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Highlighted</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>List</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Compressed</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Highlighted</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>List</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>5</td>
</tr>
</tbody>
</table>

*Table 6-23: Distribution of trial samples for experiment 2C*
6.10.2 Descriptive Statistics

6.10.2.1 Test for Homogeneity of Variances - The Levene Test

A Levene test is used to test the null hypothesis that the groups come from population with the same variance. If the observed significance level is small (<0.05), we can reject the null hypothesis that all variances are equal.

<table>
<thead>
<tr>
<th></th>
<th>Levene Statistic</th>
<th>df1</th>
<th>df2</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Based on Mean</td>
<td>2.522</td>
<td>2</td>
<td>240</td>
<td>.082</td>
</tr>
<tr>
<td>Based on Median</td>
<td>1.681</td>
<td>2</td>
<td>240</td>
<td>.188</td>
</tr>
<tr>
<td>Based on Median and with adjusted df</td>
<td>1.681</td>
<td>2</td>
<td>203.054</td>
<td>.189</td>
</tr>
<tr>
<td>Based on trimmed mean</td>
<td>2.242</td>
<td>2</td>
<td>240</td>
<td>.108</td>
</tr>
</tbody>
</table>

*Table 6-24: Test of homogeneity of variance for response time*

It can be seen from Table 6-24 that the significance of the Levene statistic based on mean is 0.082 and is considered large (i.e. >0.05). Therefore the null hypothesis is rejected. This means that we do not have sufficient evidence to suspect that the variances are unequal.

6.10.2.2 Test for Normality

Lilliefors test for normality will not always show that the data is from a normal distribution. It is almost impossible to find data that are exactly normally distributed. For most statistical tests, it is sufficient that the data are approximately normally distributed (Norusis, 1993a).

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Statistic</th>
<th>df1</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compressed 2</td>
<td>0.205</td>
<td>81</td>
<td>0.000</td>
</tr>
<tr>
<td>Compressed 2</td>
<td>0.205</td>
<td>81</td>
<td>0.000</td>
</tr>
<tr>
<td>Compressed 3</td>
<td>0.262</td>
<td>79</td>
<td>0.000</td>
</tr>
<tr>
<td>Compressed 3</td>
<td>0.262</td>
<td>79</td>
<td>0.000</td>
</tr>
<tr>
<td>Compressed 4</td>
<td>0.154</td>
<td>83</td>
<td>0.000</td>
</tr>
<tr>
<td>Highlighted 2</td>
<td>0.193</td>
<td>81</td>
<td>0.000</td>
</tr>
<tr>
<td>Highlighted 2</td>
<td>0.193</td>
<td>81</td>
<td>0.000</td>
</tr>
</tbody>
</table>
Although Lilliefors test for normality (Table 6-25) shows that the data is not from a normal distribution, the histograms shown in Figure 6-7, Figure 6-8, and Figure 6-9 shows a normal distribution.

**Figure 6-7: Histogram for the four presentation styles at complexity level 2**
Figure 6-8: Histogram for the four presentation styles at complexity level 3
Sixteen subjects participated in this experiment. These subjects also took part in experiment 2A.

The hypothesis for this experiment is that as the complexity of the search space increases so will the response times for correct identification. The increase in complexity in the search space is achieved by adding more windows containing other documents from which the target document must be identified.

Boxplots are used to show the distribution of the data. The following convention is used for these boxplots (see Figure 6-10).
Graph 6-4 shows the distribution of response times for correct identification of the target document. This is grouped by presentation style and for the 3 levels of complexities (2, 3, and 4). The primary observation that can be made from this graph is that as the complexity level increases so does the response time for correct identification. This is valid for all four-presentation styles (Compressed, Highlighted, List and Normal). In addition, higher variability for response time can be seen for complexity level 4.
Graph 6-5 shows the same data as Graph 6-4 but in this instance the plots are grouped by complexity level. From this graph observations regarding the effect response time has on presentation styles and complexity can be determined. For example, all complexity levels (2, 3, and 4), the variability of response times for correct identification are in the following sequence of low to high; compressed, highlighted, list, and normal. This support the findings outlined in section 6.7.2 that subjects achieve greater recognition accuracy and response time using the compressed presentation followed by the highlighted. This observation holds valid for all three-complexity levels.
The box plots above show the general spread of the data, the median value. The mean response time is shown in Graph 6-6. The mean response times between the complexity levels are different but are they statistical significantly different?
A one way ANOVA should show whether the differences are statistically significant. This is shown in Table 6-26.

![Bar graph showing mean response time for correct identification for the four presentation styles at the 3 complexity levels.]

**Chart 6-6: Mean response time for correct identification for the four presentation styles at the 3 complexity levels**

Table 6-26 shows that the differences in the mean response times between the complexity levels are statistically significant for the Compressed and Highlighted
presentations; $F(244,2) = 5.061$, $p < 0.05$ and $F(277,2) = 11.485$, $p < 0.01$ respectively. However, for the List and Normal presentation styles the differences in the mean response time between the complexity levels are not statistically significant.

A standard ANOVA only shows whether there is a difference between the groups but does not identify which groups are different from each other. A post hoc test is required to identify the individual groups. A post hoc test will perform multiple comparisons to determine whether the difference in response time between two complexity levels are statistically significant. The post hoc test carried out is the Bonferroni test.

<table>
<thead>
<tr>
<th>Dependent Variable</th>
<th>Complexity</th>
<th>Complexity</th>
<th>Mean Difference (I-J)</th>
<th>Std. Error</th>
<th>Sig.</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compressed</td>
<td>2</td>
<td>3</td>
<td>-0.94</td>
<td>0.627</td>
<td>0.406</td>
<td>-2.45</td>
<td>0.57</td>
</tr>
<tr>
<td>Compressed</td>
<td>2</td>
<td>3</td>
<td>-0.94</td>
<td>0.627</td>
<td>0.406</td>
<td>-2.45</td>
<td>0.57</td>
</tr>
<tr>
<td>Compressed</td>
<td>2</td>
<td>4</td>
<td>-1.95</td>
<td>0.627</td>
<td>0.406</td>
<td>-2.45</td>
<td>0.57</td>
</tr>
<tr>
<td>Compressed</td>
<td>2</td>
<td>4</td>
<td>-1.95</td>
<td>0.627</td>
<td>0.406</td>
<td>-2.45</td>
<td>0.57</td>
</tr>
<tr>
<td>Compressed</td>
<td>2</td>
<td>3</td>
<td>-0.94</td>
<td>0.627</td>
<td>0.406</td>
<td>-2.45</td>
<td>0.57</td>
</tr>
<tr>
<td>Compressed</td>
<td>2</td>
<td>3</td>
<td>-0.94</td>
<td>0.627</td>
<td>0.406</td>
<td>-2.45</td>
<td>0.57</td>
</tr>
<tr>
<td>Compressed</td>
<td>2</td>
<td>4</td>
<td>-1.95</td>
<td>0.627</td>
<td>0.406</td>
<td>-2.45</td>
<td>0.57</td>
</tr>
<tr>
<td>Compressed</td>
<td>2</td>
<td>4</td>
<td>-1.95</td>
<td>0.627</td>
<td>0.406</td>
<td>-2.45</td>
<td>0.57</td>
</tr>
<tr>
<td>Highlighted</td>
<td>2</td>
<td>3</td>
<td>9.87E-02</td>
<td>8.10</td>
<td>1.000</td>
<td>-1.85</td>
<td>2.05</td>
</tr>
<tr>
<td>Highlighted</td>
<td>2</td>
<td>3</td>
<td>9.87E-02</td>
<td>8.10</td>
<td>1.000</td>
<td>-1.85</td>
<td>2.05</td>
</tr>
<tr>
<td>Highlighted</td>
<td>2</td>
<td>4</td>
<td>3.42</td>
<td>8.12</td>
<td>0.000</td>
<td>5.38</td>
<td>-1.47</td>
</tr>
<tr>
<td>Highlighted</td>
<td>2</td>
<td>4</td>
<td>3.42</td>
<td>8.12</td>
<td>0.000</td>
<td>5.38</td>
<td>-1.47</td>
</tr>
<tr>
<td>Highlighted</td>
<td>2</td>
<td>3</td>
<td>-9.87E-02</td>
<td>8.10</td>
<td>1.000</td>
<td>-2.05</td>
<td>1.85</td>
</tr>
<tr>
<td>Highlighted</td>
<td>2</td>
<td>3</td>
<td>-9.87E-02</td>
<td>8.10</td>
<td>1.000</td>
<td>-2.05</td>
<td>1.85</td>
</tr>
<tr>
<td>Highlighted</td>
<td>2</td>
<td>4</td>
<td>-3.52</td>
<td>8.47</td>
<td>0.000</td>
<td>5.56</td>
<td>-1.48</td>
</tr>
<tr>
<td>Highlighted</td>
<td>2</td>
<td>4</td>
<td>-3.52</td>
<td>8.47</td>
<td>0.000</td>
<td>5.56</td>
<td>-1.48</td>
</tr>
<tr>
<td>List</td>
<td>2</td>
<td>3</td>
<td>-0.35</td>
<td>0.766</td>
<td>1.000</td>
<td>-2.19</td>
<td>1.50</td>
</tr>
<tr>
<td>List</td>
<td>2</td>
<td>3</td>
<td>-0.35</td>
<td>0.766</td>
<td>1.000</td>
<td>-2.19</td>
<td>1.50</td>
</tr>
</tbody>
</table>
The mean difference is significant at the .05 level.

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>-1.31</th>
<th>.756</th>
<th>.253</th>
<th>-3.13</th>
<th>.51</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 2</td>
<td>.35</td>
<td>.766</td>
<td>1.000</td>
<td>-1.50</td>
<td>2.19</td>
<td></td>
</tr>
<tr>
<td>3 2</td>
<td>.35</td>
<td>.766</td>
<td>1.000</td>
<td>-1.50</td>
<td>2.19</td>
<td></td>
</tr>
<tr>
<td>4 2</td>
<td>-96</td>
<td>.780</td>
<td>.653</td>
<td>-2.84</td>
<td>.92</td>
<td></td>
</tr>
<tr>
<td>4 2</td>
<td>-96</td>
<td>.780</td>
<td>.653</td>
<td>-2.84</td>
<td>.92</td>
<td></td>
</tr>
<tr>
<td>4 2</td>
<td>1.31</td>
<td>.756</td>
<td>.253</td>
<td>-5.1</td>
<td>3.13</td>
<td></td>
</tr>
<tr>
<td>4 2</td>
<td>1.31</td>
<td>.756</td>
<td>.253</td>
<td>-5.1</td>
<td>3.13</td>
<td></td>
</tr>
<tr>
<td>3 3</td>
<td>.96</td>
<td>.780</td>
<td>.653</td>
<td>-9.2</td>
<td>2.84</td>
<td></td>
</tr>
<tr>
<td>3 3</td>
<td>.96</td>
<td>.780</td>
<td>.653</td>
<td>-9.2</td>
<td>2.84</td>
<td></td>
</tr>
<tr>
<td>Normal</td>
<td>2 3</td>
<td>.59</td>
<td>.779</td>
<td>1.000</td>
<td>-1.29</td>
<td>2.46</td>
</tr>
<tr>
<td>Normal</td>
<td>2 3</td>
<td>.59</td>
<td>.779</td>
<td>1.000</td>
<td>-1.29</td>
<td>2.46</td>
</tr>
<tr>
<td>4 2</td>
<td>-1.20</td>
<td>.796</td>
<td>.401</td>
<td>-3.11</td>
<td>.72</td>
<td></td>
</tr>
<tr>
<td>4 2</td>
<td>-1.20</td>
<td>.796</td>
<td>.401</td>
<td>-3.11</td>
<td>.72</td>
<td></td>
</tr>
<tr>
<td>3 2</td>
<td>-59</td>
<td>.779</td>
<td>1.000</td>
<td>-2.46</td>
<td>1.29</td>
<td></td>
</tr>
<tr>
<td>3 2</td>
<td>-59</td>
<td>.779</td>
<td>1.000</td>
<td>-2.46</td>
<td>1.29</td>
<td></td>
</tr>
<tr>
<td>4 2</td>
<td>-1.78</td>
<td>.806</td>
<td>.083</td>
<td>-3.72</td>
<td>.16</td>
<td></td>
</tr>
<tr>
<td>4 2</td>
<td>-1.78</td>
<td>.806</td>
<td>.083</td>
<td>-3.72</td>
<td>.16</td>
<td></td>
</tr>
<tr>
<td>4 2</td>
<td>1.20</td>
<td>.796</td>
<td>.401</td>
<td>-7.2</td>
<td>3.11</td>
<td></td>
</tr>
<tr>
<td>4 2</td>
<td>1.20</td>
<td>.796</td>
<td>.401</td>
<td>-7.2</td>
<td>3.11</td>
<td></td>
</tr>
<tr>
<td>3 3</td>
<td>1.78</td>
<td>.806</td>
<td>.083</td>
<td>-16</td>
<td>3.72</td>
<td></td>
</tr>
<tr>
<td>3 3</td>
<td>1.78</td>
<td>.806</td>
<td>.083</td>
<td>-16</td>
<td>3.72</td>
<td></td>
</tr>
</tbody>
</table>

The Bonferroni multiple comparison tests indicated that statistically significant differences between the mean response times for correct identification between complexity level 2-4 for both the compressed and highlighted presentation was observed. In addition, complexity level 3-4 was also found to be statistically significant (see Table 6-27).

The results presented in this section do indicate that complexity level, that is, the number of windows in the search space does affect the response time for correct identification. Furthermore, the response time increases as the complexity level increases.
6.11 Summary

This Chapter presented a technique for automatic generation of keywords from a given document and applied this technique on a number of documents. The extracted keywords were used in a further experiment to determine if an adapted presentation based on keywords would yield greater recognition rate of a target document.

Three methods for presenting keywords were developed. They were compressed, highlighted and list. A fourth presentation style called the normal, that of the original document was also included in the experiment for comparison purposes. The results of the first experiment in this set found that the compressed presentation technique yielded the greatest recognition rate. The remaining three presentation techniques were very close together in terms of their correct identification percentages. However, the incorrect and don't know percentages varied between these three styles. The normal presentation style shows the highest levels of incorrect responses.

Some subjective comments were also collected from subjects. They were asked to rank the four presentation styles (compressed, highlighted, list and normal) in the order they found easiest to identify the target document. Subjects ranked the highlighted presentation style to be the easiest for identifying a document followed by list, compressed and then normal.

The most interesting region found in the analysis was the lower end of the screen utilisation (that at less than 30%). This is the region where one would like to increase the recognition rates, so that one could reduce the screen real estate occupied by inactive windows without compromising the recognition rate.

The result showed that above 30% screen utilisation subjects tended to score on average 80% correct identification rate. Between 0-9% the compressed strategy showed higher recognition and this was found to be statistically significant. For the
10-19% screen utilisation groups all adapted presentation style (compressed, highlighted, list) showed higher recognition rates than the normal. However, there was no statistical evidence to indicate that the distribution of the sample means was different.

Therefore, a further experiment was conducted only considering screen utilisation between 0-30%. The results showed that an accuracy of 60% is reached by the compressed presentation at 6% screen utilisation, in contrast to the other presentation styles which reached this level at a screen utilisation percentage of 9%. Again, the results indicated that subjects achieved higher recognition rates using the compressed presentation, closely followed by the highlighted presentation.

The third and final experiment in this set was designed to test whether increasing the number of windows in the search space affected response times for correct identification. The result showed that the number of windows in the search space did effect the response time for correct identification. It increased as the number of windows in the search space increased.

The conclusion reached from this experiment is that the vanishing windows system should adapt/transform the contents of the window by compressing the keywords. However, taking into consideration the subjects preference for the highlighted presentation, it was decided to use the highlighted presentation instead of the compressed presentation style. Furthermore, subjects preferred the highlighted presentation as it retained positional cues of the document.
Chapter 7

7 VANISHING WINDOWS: AN EMPIRICAL STUDY

7.1 Introduction
Chapter 4 presented the Vanishing Windows Concept. Using the results from Chapters 5 and 6, a full Vanishing Windows system was implemented. This chapter presents the results of an empirical study conducted using this system. The test compares user performance for a given task on a Non-Vanishing Windows-95 like System with the Vanishing Windows system developed in the thesis.

Subjective comments from users were collected in the form of a questionnaire. The function of the questionnaire was to ascertain how users viewed the usefulness of the main features of the Vanishing Windows system (such as highlighting of keywords, removal of toolbars on inactive windows and the Vanishing Windows concept itself). In addition, the questionnaire was used to gather information on user attitudes towards auto-adaptation since adapting user presentations dynamically, as users work, could be unsettling.

7.2 Evaluating the Vanishing Windows System
The Vanishing Windows System is to be evaluated using an experimental evaluation technique (see Chapter 2 - evaluation methods). The basic hypothesis being tested is “does Vanishing Windows system improve use performance over a traditional windowing system?”

User performance can be measured in terms of the throughput; execution time to complete a task or in terms of accuracy (see chapter 2 - evaluation methods).

Execution time as a measure can vary widely between users and can be problematic to analyse because of the number of factors that can effect execution time between
users. For example, some users may be able to comprehend the task more easily than other users. In complex tasks, involving a number of sub-tasks and sub goals, the execution time can vary depending on the sequence and operations the user employs to achieve the target goal. Another difficulty of using execution time as a performance measure is because the systems being compared may have different system overheads.

A claimed benefit of the Vanishing Windows System is easier identification of a desired application or task amongst a set of windows, achieving rapid switching between applications, without time consuming window management operations. Therefore, one good measure for performance is the number of windows a user switches between before the desired window is reached.

Another issue is learning. If the same task is used for both systems so that direct comparisons can be made, subjects may learn to become more adept at the task, which could then affect the result. This can be avoided by counter balancing of subjects. Half of the subjects’ work is carried out under condition A first, and half work under condition B first (see Table 7-1). This is a variation from the randomised block design and is known as a Latin square design (Ott and Mendenhall, 1994).
Chapter 7  Vanishing Windows: An Empirical Study

<table>
<thead>
<tr>
<th>Vanishing Windows</th>
<th>Non-Vanishing Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject1 (first)</td>
<td>Subject1 (second)</td>
</tr>
<tr>
<td>Subject2 (second)</td>
<td>Subject2 (first)</td>
</tr>
<tr>
<td>Subject3 (first)</td>
<td>Subject3 (second)</td>
</tr>
<tr>
<td>Subject4 (second)</td>
<td>Subject4 (first)</td>
</tr>
<tr>
<td>Subject5 (first)</td>
<td>Subject5 (second)</td>
</tr>
<tr>
<td>Subject6 (second)</td>
<td>Subject6 (first)</td>
</tr>
<tr>
<td>Subject7 (first)</td>
<td>Subject7 (second)</td>
</tr>
<tr>
<td>Subject8 (second)</td>
<td>Subject8 (first)</td>
</tr>
</tbody>
</table>

*Table 7-1: Task Ordering*

7.3 Experimental Design

7.3.1 The Hypothesis

The hypothesis predicts that performance times will be faster, for the Vanishing Windows interface and that there will be lower user management operations in the Vanishing Windows. The Vanishing Windows interface is expected to simplify the users cognitive workload by relieving them of searching difficulties that we expect may occur in a non-vanishing windows interface. The user's cognitive workload is simplified in two ways, firstly, by adapting the inactive window presentation to highlight keywords that uniquely identify the document and secondly, by a gradually transformation to a tiled layout.

The secondary hypothesis is that, as the task complexity increases (i.e. the number of windows in the search space), the greater the difference between performance measures for the two interface will be observed.

7.3.2 The Task

A task was developed which required subjects to perform several searches to locate a required document contained in a window on the desktop. The search task was performed using both the Vanishing Windows System and a Non-Vanishing Windows system.
A number of windows containing documents are presented to the subject. The
documents are taken from a number of papers or reports from the computing field.
These documents form the search space for the task. All are presented in windows
simultaneously. One document is the target document. Questions regarding the
target document's content are asked. The subject is required to locate the relevant
document and record an answer about the content. An example question is as
follows (see Appendix D.2 for the full set of questions):

How many references are listed in the video conferencing document?

Does the word 'Intelligent' appear in section 4 paragraph 2 of the paper titled 'Mobile
Worker: Access to Information on the Move'?

In the paper titled 'A Web-Based Framework for Distributed Expert Systems' what is the
title of figure 4?

The task complexity is defined by the number of windows displayed on the screen.
Three complexity levels were tested: Low (4 windows), Medium (6 windows), and
High (8 windows).

Each subject answers five questions in each complexity level using both the
Vanishing Windows and Non-Vanishing Windows systems. The order of the
complexity level is varied across subjects to reduce any errors that may be introduced
by a particular sequence.
The schedules for 12 subjects are as follows:

<table>
<thead>
<tr>
<th>Subject</th>
<th>First</th>
<th>Second</th>
<th>Task Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>VWin1</td>
<td>Win2</td>
<td>(L) (M) (H)</td>
</tr>
<tr>
<td>S2</td>
<td>VWin2</td>
<td>Win1</td>
<td>(M) (H) (L)</td>
</tr>
<tr>
<td>S3</td>
<td>VWin1</td>
<td>Win2</td>
<td>(H) (L) (M)</td>
</tr>
<tr>
<td>S4</td>
<td>VWin2</td>
<td>Win1</td>
<td>(L) (H) (M)</td>
</tr>
<tr>
<td>S5</td>
<td>VWin1</td>
<td>Win2</td>
<td>(H) (M) (L)</td>
</tr>
<tr>
<td>S6</td>
<td>VWin2</td>
<td>Win1</td>
<td>(M) (L) (H)</td>
</tr>
<tr>
<td>S7</td>
<td>Win1</td>
<td>VWin2</td>
<td>(L) (M) (H)</td>
</tr>
<tr>
<td>S8</td>
<td>Win2</td>
<td>VWin1</td>
<td>(M) (H) (L)</td>
</tr>
<tr>
<td>S9</td>
<td>Win1</td>
<td>VWin2</td>
<td>(H) (L) (M)</td>
</tr>
<tr>
<td>S10</td>
<td>Win2</td>
<td>VWin1</td>
<td>(L) (H) (M)</td>
</tr>
<tr>
<td>S11</td>
<td>Win1</td>
<td>VWin2</td>
<td>(H) (M) (L)</td>
</tr>
<tr>
<td>S12</td>
<td>Win2</td>
<td>VWin1</td>
<td>(M) (L) (H)</td>
</tr>
</tbody>
</table>

*Table 7-2: Schedule for experiment 3*

VWin1 stands for Vanishing Windows interface with the first set of questions and Win2 stands for standard windows interface with the second set of questions. L, M, and H stand for low, medium, and high complexity task corresponding to 4, 6 and 8 window situations.

For example, consider S2

\[ \text{VWin2} \rightarrow \text{Win1(M) (H) (L)} \]

According to this schedule subject 2 will use the Vanishing Windows interface to answer question set 2, and then use the standard Windows interface to answer questions from set 1. Subject 2, will work with tasks in the following sequence of complexity: medium, high and low complexity.

Each subject answered 5 questions from each of the complexity levels, using both windowing systems. Two sets of documents and questions were used. Each set consisted of 8 documents and 15 questions (5 for each complexity level). These sets were assigned to each subject based on the schedule shown in Table 7-2.
7.3.3 Measurements

A question is referred to as a sub-task. Therefore, a subject completes five sub-tasks per complexity level and the time to complete a sub-task is recorded as a sub-task completion time. The sub-task completion times can be divided into two sequential individual times. These are the Task Setup Time and the Task Execution Time. In this experiment, the task setup time is defined as the time required to identify and locate the required window to complete the sub-task. The task execution time is time required to answer the question.

The task execution time is very much dependent on a number of variables such the subject skill, ability and comprehension of the question. Therefore, it was considered a better approach to use the task setup time to compare performance when working with the two windows interface (vanishing and non-vanishing). In addition, the claimed benefits for Vanishing Windows is the reduction of search time, so this measure would seem more appropriate.

If subjects select the wrong window, this is considered as an error, and the subject continued to locate the required document. This simply increased the subjects time to locate the required window.

7.4 Experimental Procedure

Prior to the experiment each subject was given a 5 minute training session with the Vanishing Windows system, and 20 minutes to familiarise themselves with a set of eight documents. Since the subjects did not write the documents, they will be less familiar with them. But this still does accord with many typical document handling situations. The subject then continued with the experiment using one of the windowing systems and complexity sequences assigned to them. Upon completion, they are given another set of eight documents to familiarise themselves with, and then tested on the other windowing system. A short questionnaire (see Appendix D.3) was completed at the end of the experiment to ascertain user background,
experience level and whether Vanishing Windows was helpful and users views on auto-adaptation.

The experiment was carried out on a laptop. Screen resolution was 600x480 pixels and the screen size was 25cm x 19cm.

7.5 Results

Twelve subjects participated in the experiment. They were all volunteers from the department of Computer Studies and were mainly research students, masters student and programmers.

7.5.1 An Analysis of Vanishing Windows Performance

7.5.1.1 Search Time

A similar spread/variability of the data for search times is observed for subjects using both the vanishing and non-vanishing windows as indicated by the length of the boxes in Graph 7-1 (see chapter 6, figure 6-10 for notation used for boxplots). However, the extreme values were greater in the non-vanishing windows case. Furthermore, the data shows that in all cases, as the complexity increased, the search time also increased lending support to our view of complexity.
Chapter 7

Vanishing Windows: An Empirical Study

Graph 7-1: Boxplots grouped by windowing system - search time to locate target window for three complexity levels

Table 7-3 shows the basic descriptive statistics for the search time (time to locate required window) in the three complex environments, where complexity of the environment is defined by the number of windows displayed.

<table>
<thead>
<tr>
<th></th>
<th>VWIn (LOW)</th>
<th>VWIn (MED)</th>
<th>VWIn (HIGH)</th>
<th>Win (LOW)</th>
<th>Win (MED)</th>
<th>Win (HIGH)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num of Sample</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Minimum</td>
<td>86</td>
<td>108</td>
<td>152</td>
<td>77</td>
<td>119</td>
<td>141</td>
</tr>
<tr>
<td>Maximum</td>
<td>550</td>
<td>364</td>
<td>506</td>
<td>310</td>
<td>460</td>
<td>716</td>
</tr>
<tr>
<td>Mean</td>
<td>212.33</td>
<td>231.25</td>
<td>294.08</td>
<td>179.83</td>
<td>221.58</td>
<td>342.92</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>127.07</td>
<td>83.33</td>
<td>117.61</td>
<td>72.90</td>
<td>101.90</td>
<td>176.15</td>
</tr>
<tr>
<td>Variance</td>
<td>16146.242</td>
<td>6943.659</td>
<td>13832.265</td>
<td>5314.333</td>
<td>10383.356</td>
<td>31030.447</td>
</tr>
<tr>
<td>Skewness</td>
<td>1.815</td>
<td>0.111</td>
<td>0.605</td>
<td>0.300</td>
<td>1.348</td>
<td>1.199</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>4.282</td>
<td>-1.169</td>
<td>-0.766</td>
<td>-1.081</td>
<td>1.568</td>
<td>0.708</td>
</tr>
</tbody>
</table>

*Table 7-3: Descriptive statistics for search time*
The following observations can be made from the descriptive statistics:

- The Standard Deviation and Variances for the Vanishing Windows case are lower for medium and high complexities than for the Windows case. However, the reverse is true for the low complexity case. This indicates that subjects took a similar duration of time to perform the search task when using the Vanishing Windows system. In addition this time was lower than its counterpart in the Windows system.

- The skewness value indicates that there were a greater number of lower search time values when subjects used the Vanishing Windows system in the low complexity environment (positively skewed). However, this was reversed in the medium and high complexity cases. Ideally, positive skew value for the Vanishing Windows case would indicate that on the skew test Vanishing Windows is showing better performance.

Graph 7-2 shows the average search times (for 5 searches) to located the required window for different complexity levels, for the two windowing system (Vanishing and Non-Vanishing Windows). For the high complexity environment (i.e. 8 windows), subjects using the Vanishing Windows technique, on average, took less time to locate the required window. However, for the low and medium complexity levels, subjects on average took less time using the Non-Vanishing Windows System.

A Paired Sample t-test was carried out to determine if the difference in the means was statistically significant, between the Vanishing and Non-Vanishing systems.
Vanishing Windows: An Empirical Study

Graph 7-2: Average search times for Low, Medium, and High complexity environment for both windowing systems

The results of the paired 2-tailed t-test shown in Table 7-4, indicate that the difference in the average search time between each pair is not statistically significant.

<table>
<thead>
<tr>
<th>Pair</th>
<th>df</th>
<th>t</th>
<th>Sig. (2-tailed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VWIN LOW - WIN LOW</td>
<td>11</td>
<td>1.206</td>
<td>0.253</td>
</tr>
<tr>
<td>VWIN MED - WIN MED</td>
<td>11</td>
<td>0.362</td>
<td>0.724</td>
</tr>
<tr>
<td>VWIN HIGH - WIN HIGH</td>
<td>11</td>
<td>-1.130</td>
<td>0.282</td>
</tr>
</tbody>
</table>

Table 7-4: Paired 2-tailed t-test for search times

However, the search time did not take into consideration the system performance differences between the two windowing system. The Vanishing Windows system had a slower system response time, which affected the search time. As users switched between windows, they had to wait approximately 1 second for the system to process and adapt the windows. Therefore, the search times shown in Graph 7-2 are higher than what they should have been compared to Non-Vanishing Windows. The number of windows operation carried out during the search task was also analysed. This was expected to be a more reliable measure for performance.
7.5.1.2 Number of Window Operations

The distribution for the number of window operations shows less of a variation for the vanishing windows case than its counter part in the non-vanishing case. In terms of the variability the data presented in Graph 7-3 (see Chapter 6, Figure 6-10 for notation used for boxplots), this indicates that users performed their task within a lower margin of variation. Again, like the search time, the extreme values for non-vanishing case are greater than the vanishing case.

Table 7-5 present the descriptive statistics for the number of windows operation carried out during the search task for both windows systems and all three complexity levels. Both standard deviation and variances are lower for the vanishing windows case for medium and high complexity, indicating that subjects performed the search task in less time and the search completion times were within a lower margin than of the non-vanishing windows system.
Chapter 7  
Vanishing Windows: An Empirical Study

<table>
<thead>
<tr>
<th></th>
<th>V\text{Win} (LOW)</th>
<th>V\text{Win} (MED)</th>
<th>V\text{Win} (HIGH)</th>
<th>W\text{in} (LOW)</th>
<th>W\text{in} (MED)</th>
<th>W\text{in} (HIGH)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num of Sample</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Minimum</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>Maximum</td>
<td>32</td>
<td>33</td>
<td>25</td>
<td>31</td>
<td>44</td>
<td>78</td>
</tr>
<tr>
<td>Mean</td>
<td>11.83</td>
<td>12.17</td>
<td>15.83</td>
<td>13.83</td>
<td>17.42</td>
<td>29.92</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>9.23</td>
<td>7.33</td>
<td>6.29</td>
<td>8.50</td>
<td>11.28</td>
<td>20.46</td>
</tr>
<tr>
<td>Variance</td>
<td>85.242</td>
<td>53.788</td>
<td>39.606</td>
<td>72.333</td>
<td>127.174</td>
<td>418.629</td>
</tr>
<tr>
<td>Skewness</td>
<td>1.564</td>
<td>2.345</td>
<td>0.239</td>
<td>1.109</td>
<td>1.498</td>
<td>1.788</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>1.253</td>
<td>6.466</td>
<td>-1.208</td>
<td>-0.085</td>
<td>1.683</td>
<td>2.311</td>
</tr>
</tbody>
</table>

*Table 7-5: Descriptive statistics for number of windows operations*

The average number of window operations a subject used to locate the desired window for each complexity level for the two windowing systems is shown in Graph 7-4. For all three-complexity levels, subjects used fewer windowing operations to locate the required window. As expected, the differences between the Vanishing and Non-Vanishing systems increased as complexity increases. Since, in the medium complexity environment, the search space consisted of 6 windows, and for the high complexity environment it was 8 windows, this had a direct bearing on Short Term Memory Effects (Miller, 1956). As the number of Windows increases beyond 5, users will begin to experience difficulty in short-term memory and may forget where documents are located within the search space.
Again paired samples t-tests were conducted to check if the results were statistically significant. The results of the t-tests are presented in Table 7-6 and show a high statistical significance in all three cases. The differences between the mean number of window operations required to locate the target window when using the Vanishing Windows system were lower for all complexity levels.

<table>
<thead>
<tr>
<th>Pair</th>
<th>df</th>
<th>t</th>
<th>Sig. (2-tailed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VWIN LOW - WIN LOW</td>
<td>11</td>
<td>-1.414</td>
<td>0.000</td>
</tr>
<tr>
<td>VWIN MED - WIN MED</td>
<td>11</td>
<td>-2.454</td>
<td>0.004</td>
</tr>
<tr>
<td>VWIN HIGH - WIN HIGH</td>
<td>11</td>
<td>-3.031</td>
<td>0.003</td>
</tr>
</tbody>
</table>

Table 7-6: Paired 2-tailed t-test for number of windows operation required to locate required window

7.5.2 Analysis of Questionnaire

For each of the questions asked, the frequency of each response and the percentage of the subjects that made that response are tabulated. In addition a Chi-square goodness-of-fit was performed. This goodness-of-fit test compares the observed and expected
frequencies in each category to test whether all categories contain the same proportion of values.

The questionnaire asked five key questions. The first three questions probed the usefulness of features of the system and the remaining two were concerned about user attitudes to auto system adaptation since adapting user presentations dynamically as users work might be very unsettling.

7.5.2.1 Usefulness of Highlighted Keywords

Subjects were asked: "Did you find the highlighted keywords on Inactive Windows helpful in identifying the document?"

<table>
<thead>
<tr>
<th></th>
<th>Number of Subjects</th>
<th>Percentage of Subjects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Helpful</td>
<td>3</td>
<td>25%</td>
</tr>
<tr>
<td>Helpful</td>
<td>6</td>
<td>50%</td>
</tr>
<tr>
<td>No Difference</td>
<td>3</td>
<td>25%</td>
</tr>
<tr>
<td>Unhelpful</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Annoying</td>
<td>0</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 7-7: Usefulness of Highlighted keyword

As indicated by Table 7-7, 50% of the subjects found that highlighting keywords on inactive windows helped them in identifying the required document. The Chi-square test ($\chi^2=15.0, df=5, p<0.05$) show there is evidence for a significant departure from equal choices.

7.5.2.2 Usefulness of Toolbars Being Removed

Subjects were asked: "Did you find the removal of toolbars on Inactive Windows helpful in identifying the document you were looking for?"
The response of subjects varied between two choices. They either found the removal of toolbars helpful or it made no difference to them. Either way, the data presented show a positive response. Again the Chi-square \( \chi^2=33.0, \text{df}=5, p<0.01 \) test performed show there is evidence for a significant departure from equal choices.

### 7.5.2.3 Usefulness of Vanishing Windows

Question asked: *Did you find Vanishing Windows...*

<table>
<thead>
<tr>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Helpful</td>
<td>0</td>
</tr>
<tr>
<td>Helpful</td>
<td>9</td>
</tr>
<tr>
<td>No Difference</td>
<td>3</td>
</tr>
<tr>
<td>Unhelpful</td>
<td>0</td>
</tr>
<tr>
<td>Annoying</td>
<td>0</td>
</tr>
</tbody>
</table>

*Table 7-8: Usefulness of toolbars being removed*

This question was used to find out whether Vanishing Windows helped users in their task (search task). As can be seen from Table 7-9, 33% of the subjects found Vanishing Windows very helpful and 42% found it just helpful. Fewer subjects found it unhelpful (17%). On average, it can be concluded that subjects did find the Vanishing Windows more helpful than unhelpful. Again the Chi-square \( \chi^2=11.0, \text{df}=5, p=0.05 \) test indicates a significant departure from equal choices.

<table>
<thead>
<tr>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Helpful</td>
<td>4</td>
</tr>
<tr>
<td>Helpful</td>
<td>5</td>
</tr>
<tr>
<td>No Difference</td>
<td>1</td>
</tr>
<tr>
<td>Unhelpful</td>
<td>2</td>
</tr>
<tr>
<td>Annoying</td>
<td>0</td>
</tr>
</tbody>
</table>

*Table 7-9: Usefulness of Vanishing Windows*
7.5.2.4 Loss of Control

Question asked: Did you feel a loss of control when working with Vanishing Windows?

By loss of control the author meant whether subjects were concerned that the windows were changing both in terms of their size and the contents of the document. Did they feel that the documents were going to be lost?

<table>
<thead>
<tr>
<th></th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Most of the Time</td>
<td>1</td>
<td>8%</td>
</tr>
<tr>
<td>No Opinion</td>
<td>1</td>
<td>8%</td>
</tr>
<tr>
<td>Some of the Time</td>
<td>5</td>
<td>42%</td>
</tr>
<tr>
<td>Never</td>
<td>5</td>
<td>42%</td>
</tr>
</tbody>
</table>

*Table 7-10: Responses for loss of control*

This is an important question that most (ROOMS, Henderson and Card, 1986; CUBRICON, Funke et al., 1993; AWM, Stille et al., 1997; Maw3-3D Window Manager, Leach et al. 1997) of the research papers reviewed either failed to answer, or failed to test. The data presented in Table 7-10, showed that 42% of the subjects reported that they never felt a loss control when using the Vanishing Windows Systems. Another 42% said they sometimes felt a loss of control. However, this is also a positive result. Therefore, it can be concluded that on average subjects were in control when using the Vanishing Windows System. Only one person reported as to a feeling of loss of control when using the Vanishing Windows system. This person during the experiment also indicated a preference for command line interfaces as opposed to direct manipulation interfaces, therefore the choice may have been biased against windowing system. Again the Chi-square ($\chi^2=14.0$, df=5, $p<0.05$) statistic indicates a significant departure from equal choices.

7.5.2.5 Disruption of Automatic Movement (Auto-Adaptation)

Another important research question for adaptive interfaces is whether to perform auto-adaptation or not. Two reasons put forward (Kühme, 1993) against auto-
adaptation are distraction/disruption of current work activity and automatic changes may not always be obvious or expected by the user.

Question asked: Did you find the automatic movement of Windows (i.e. Vanishing) disruptive?

<table>
<thead>
<tr>
<th></th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always</td>
<td>1</td>
<td>8%</td>
</tr>
<tr>
<td>Most of the Times</td>
<td>1</td>
<td>8%</td>
</tr>
<tr>
<td>No Opinion</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Some of the Times</td>
<td>5</td>
<td>42%</td>
</tr>
<tr>
<td>Never</td>
<td>5</td>
<td>42%</td>
</tr>
</tbody>
</table>

*Table 7-11: Disruption caused by auto-adaptation*

The distribution of responses for disruption caused by auto-adaptation is similar to the question of 'loss of control'. Table 7-11 indicates that majority of the subjects were not distracted/disrupted by inactive window size changing. Again it was the one subject (who favoured command line interface) that felt disruption when using Vanishing Windows. The Chi-square ($\chi^2=14.0$, df=5, $p<0.05$) test show a statistically significant departure from equal choices.

7.5.2.6 Preference of Windowing System

Question asked: Which version of Window would you choose to work with?

<table>
<thead>
<tr>
<th></th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vanishing</td>
<td>7</td>
<td>58%</td>
</tr>
<tr>
<td>No Preference</td>
<td>3</td>
<td>25%</td>
</tr>
<tr>
<td>Non-Vanishing</td>
<td>2</td>
<td>17%</td>
</tr>
</tbody>
</table>

*Table 7-12: Windowing system preference*

Although the data presented in Table 7-12 show that a higher percentage of subjected preferred to work with Vanishing Windows, the Chi-square ($\chi^2=3.5$, df=2, $p>0.05$) test did not show a statistically significant departure from equal choices.
Some of the subjects’ comments as to their preference for a particular windowing system are listed below.

- “I preferred Vanishing Windows because it was much easier to find a particular window in the examples where there was more than 4 windows.”
- “It makes the document identification a lot easier and manages the screen size effectively”
- “… Vanishing Windows is good for a lot of windows opened at once.”
- “Easier to find a document when the screen was cluttered with lots of windows.”
- “Find it hard to keep track of where the window’s position when returning to it. However highlighted keywords were a help.”
- “Windows (Normal Non-Vanishing) also has annoyances, so I prefer a more configurable windowing system. I tend to use fvwm under X11R6 with a focus follows mouse policy so I can ‘hunt’ through overlapping windows really quickly” – User had no preference for either Vanishing or Non-Vanishing Windows system.

7.5.3 Subjective Comments

Some of the more general comments by the subjects are listed below:

- “With Fixed Windows (i.e. Non-Vanishing), it’s easy to remember where the windows are, however this only applies for low number of windows.”
- “I found that when Vanishing Windows moved the windows to stop them from overlapping, I lost track of which window was which.”
• "Make better use of 'white spaces' in the search condition" – This subject seems to indicate that windows should also expand to make use of 'white spaces'.

• "Sometimes the Vanishing Window caught my attention as it changed size, and was sometimes distracting. This was because of the novelty of it."

• "I didn't feel a loss of control, because the selection of the window restored that window to its original size and position."

• "Easier to identify the window with some of the highlighted keywords."

7.6 Summary
This experiment has compared Vanishing Windows with a Non-Vanishing Windows system in terms user performance times to locate a target window, and the number of windows operation needed to located a target window. Statistically significant results were found in favour of the Vanishing Windows system in terms of the number of windows operation needed to locate a target window. In addition, as the complexity (number of windows in the search space) increased so did the performance differences between the systems.

Lower search times to locate a window in the Vanishing Windows system were observed for the high complexity case. However, this result was not statistically significant. For the low (4 Windows) and medium (6 Windows) complexity cases the Non-Vanishing Windows results showed better performance in terms of search times. Again, no statistical significance was found in favour of the Non-Vanishing Windows.

Because of the additional processing time required by the Vanishing Windows system, system response times were slower than for the Non-Vanishing Windows system case. Therefore, we can speculate that search times for the Vanishing
Windows would have been lower, if the system performance of the Vanishing Windows to the Non-Vanishing Windows were comparable.

Overall, subjects did not feel any loss of control when using Vanishing Windows, nor were they distracted from their tasks by the automatic movement/resizing of inactive windows.
Chapter 8

8 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH

8.1 Summary of Thesis
The objective of the research in this thesis was to develop the Vanishing Windows concept and explore its potential for allowing for more intelligent management of screen real estate. The aim of the system was to increase user performance by reducing the number of window manipulations required to complete a task.

To realise its aims, the Vanishing Windows approach progressively reduces the window size. This is achieved through a number of operations that were utilised in the final prototype. These operations were:

1. *Migrate* – Move windows away from overlapping sides.
2. *Shrink* – Reduce window size from the overlapping sides
3. *Shunt* – When a window collides with another window during a *Migrate* operation, move both windows away from overlapping sides as a single object.
4. *Evade* – This is an alternative to the *Shunt* operation. In the *Evade* operation, when a window collides with another window during a *Migrate* operation, the migrating window side-steps the window that it has collided with.

As a consequence of the shrinking process, valuable visual cues were lost from the window contents. Therefore, techniques had to be developed to compensate for the loss of these visual cues. Experiments were conducted to evaluate the usefulness of two approaches. The first experiment tested whether subjects found it easier to identify a document when reducing the contents by Scaling or by Cropping. The second experiment developed an automatic technique for generating keywords (words that can be used to uniquely identify a document) in the window and a
number of methods for presenting those keywords. These presentation methods were tested to see which presentation style gave higher recognition rates when windows were presented in different sizes. The results of these two experiments were integrated into the design of the final system and a third experiment conducted to assess whether significant user performance could be gained when using the Vanishing Windows system compared with a traditional Non-Vanishing Windows system such as Windows 95.

8.1.1 Cropped Verses Scaling

The results of the cropped verses scaling experiment indicated that better performance, in terms of accuracy of identification, was achieved by the cropped presentation style. As expected, as the screen utilisation percentage (i.e. window size as a percentage of the desktop size) increased so did the accuracy of identification. Above 80% screen utilisation the accuracy of identification between both presentation styles began to converge. This is an expected result, since at higher screen sizes the scaling factor used on the scaled presentation produces an output similar to the cropped presentation. Similarly, in high screen sizes in the cropped presentation style, no cropping of information takes place hence the display produced is similar to it scaled counterpart.

A statistically significant difference between the mean number of correct identifications between the two presentation styles was found for screen utilisation percentages of \(< 30\%\), in favour of the cropped presentation style.

The conclusion reached from this experiment was that the Vanishing Windows system should crop its window contents as it reduced in size. In addition, the experiment also demonstrated that, by removing the toolbars, additional screen real estate could be freed for the application, and users would be able to take advantage of the information presented in this additional space. Therefore another design implication for the Vanishing Windows system was to remove toolbars on inactive
windows, as they are of no value to the user when windows are inactive. When Windows becomes active, the toolbar can be placed back on the application window.

8.1.2 Transforming Window Contents

The objective of the second experiment was to develop techniques to further reduce window size whilst minimising the loss of visual cues. This enabled the Vanishing Windows system to further reduce window size and still maintain acceptable recognition rates.

When conducting the first experiment, users had commented that they identified the document using keywords (together with other attributes such as layout, fonts and pictures). At low screen sizes, the area available in the Vanishing Windows is necessarily small and the number of defining words in the document may be reduced to a low level since other non-defining words will also be there. In order to improve recognition at low screen sizes, a technique for increasing the proportion of defining words (or keywords) in a document was required.

The technique for automatically identifying defining keywords was as follows. The frequency of a word in a document was compared with a standard corpus of English usage. If its frequency was significantly greater than its frequency in the corpus, the word was taken to be a keyword (i.e. a word that uniquely identifies the document).

Three methods for presenting these keywords in reduced documents were developed, which could enhance the visual cues at low screen size. These methods were highlighting, listing and compressing the keywords. A fourth presentation style, where no transformation of the content of the document took place was also included in the experiment, this was referred to as the normal presentation. The second experiment was therefore carried out to determine if higher identification rates could be achieved using one or more of these presentation styles.
Chapter 8

Conclusions & Further Research

The overall result showed that the *compressed* presentation technique yielded the greatest recognition rate. The remaining three-presentation techniques were very close together in terms of the correct identification percentages. However, the "incorrect" and "don't know" percentages varied between these three styles. The *normal* presentation showed the highest incorrect responses.

This suggested that the Vanishing Windows system should adapt the contents of the document in an inactive window using the compressed version. However, closer examination of subject's preference for one of the presented styles tested revealed a preference for the *highlighted* style. The highlighted presentation retains all other visual cues available from the document such as headings of sections, text in bold or Italics and layout.

The Vanishing Windows System was therefore designed so that the contents of inactive windows, transformed by highlighting keywords, would make an individual document easier to identify.

8.1.3 Vanishing Windows - User Performance

The final experiment tested whether user performance improved when using the Vanishing Windows System. Subjects were tested on search tasks using both Vanishing and Non-Vanishing Windows systems. Three task complexity levels were used.

The results showed that subjects were able to locate desired windows in less time using the Vanishing Windows System than the Non-Vanishing Windows (Windows 95). However, statistically significant differences between the mean times were found only for the high complexity task, that is, a search task where there were eight windows in the search space. Another factor that was considered in the experiment was whether fewer window manipulations were used in the search task when using Vanishing Windows. The result indicated that on average users accomplished the search task using fewer windows operations using the Vanishing Windows than the
Non-Vanishing System. The results were found to be highly statistically significant in all three cases of task complexity (Low – 4 Windows, Medium – 6 Windows and High – 8 Windows).

Many authors' (Thomas and Krogstie, 1993; Kühme, 1993; Debevc, 1993) speculation suggests that users generally feel a loss of control when using an adaptive system using auto-adaptation. The results of this thesis in the form of a user questionnaire showed that, on average, subjects did not find the movement of windows disruptive or a general feeling of loss of control. Further detailed studies need to be carried out to determine under what conditions do users feel a loss of control.

8.1.4 Ecological Validity

One could argue that experiments carried out using text based documents interspersed with a few diagrams are not representative of a multimedia environment (the sort of environment one might see in offices today). Today's applications are concerned with other types of media such as video, speech, graphics, structured drawings, such as CAD and structured database text forms.

However, the concept of Vanishing Windows is not reliant on a word processing environment or any particular medium. The vanishing process will work equally well in any media type. Possible extensions to the system could include new inferencing approaches where the type of media is a parameter in the reduction strategy. For example, the current system only crops windows when they shrink, but for windows containing either live or static video, it may be preferably to shrink windows whilst scaling the contents.

8.2 Thesis in Retrospect

There were some questions raised in the thesis which were not investigated in detail. For example, clutter analysis was reviewed in Chapter 2, but was not fully exploited within the thesis. If a good measure for clutter can be established for a windowing
systems then this can be measured in real time to determine if the clutter is beyond a threshold. If clutter level exceeds this threshold then a clutter reduction method as developed in this thesis can be applied.

Again, Bonsiepe’s measure of complexity was briefly reported but not investigated in the thesis, as it was felt that no substantial evidence of the application of the technique to windowing system had been developed to a stage where conclusions could be made. However, software was written to calculate the complexity based on horizontal and vertical alignment of windows using Bonsiepe’s measure of complexity. The software was a simulation of the computer’s desktop where the user could select the number of windows to display and the software then placed the windows in random positions on the screen. The software then calculated the complexity value. It was found that when the windows were perfectly aligned this produced a low complexity value. However, when windows were not perfectly aligned the complexity value was the same for a given number of windows and not dependent on position of the windows and amount of overlap. A developed measure should encompass position of windows, size of windows and amount of overlap.

Questions raised by experiment
In the scaled version of the cropping experiment and the adapted presentation, subjects were given only the target document to read through and become familiar with the document. However, these experiments could have been expanded to examine the effect on user performance when all the documents that were used in the experiment were given to the subject. One might speculate that the result under this condition would be lower response times for identification and possibly a marginal increase in number of correct identifications. In addition, the length of time given to subjects may have had an effect on the subject performance and again further experiments could have been setup to tests these conditions.

Increasing the number of subjects for experiment 3 (Vanishing Windows verses Non-Vanishing Windows), may have changed the quantitative results marginally but the
Overall findings would have remained essentially the same. However, the qualitative results would have benefited from a greater number of subjects and provided a much clearer understanding of what subjects’ liked/disliked about the system. Furthermore, subjects could have been grouped by level of expertise and correlated with preference for the system.

8.3 Conclusions
Shneiderman (1983) first coined the term ‘Direct Manipulation’ in 1983 and produced some well-defined guidelines which have been utilised by software interface designers over the years. However, there is a lack of clearly defined design principles and guidelines specifically for intelligent user interfaces. Specific design principles need to be developed for intelligent user interfaces rather than re-using ones that have been developed for direct manipulation. For example, Shneiderman (1992) presented eight golden rules of interface design, one of which is the need for interface consistency. However, an objective of an adaptive interface is to adjust itself to fit the user, situation or environment, and this may violate the interface consistency rule. Therefore, work needs to be done to harmonise the design principles for direct manipulation interfaces with ones developed for intelligent user interfaces.

This observation is supported by Höök (1998) who outlines a number of requirements that need to be addressed before intelligent user interfaces become real and usable. One of these requirements is the development of specific usability principles for intelligent interfaces, as opposed to the re-use of principles developed for direct manipulation systems.

It is obvious that the Vanishing Windows System violates this interface predictability principle but in this case, the violation is acceptable. This acceptance maybe due to the fact that it is very clear to the user what the system is changing. Kühme (1993) refers to ‘Transparency of Adaptation’ and states “... supporting the user in developing an adequate mental model of the system’s adaptation mechanisms and
strategies..." can diminish some of the problems of adaptations such as users feeling a loss of control, distributed or confused by unexpected adaptations carried out automatically by the system.

There may be situations where violation of the interface predictability principal can cause problems and others where they do not. These situations need to be identified and developed into guidelines for the design of intelligent user interfaces.

There have been many prototype systems developed by the research community that addressed the problem of 'Window Thrashing' but they failed to provide extensive evaluation of the system. Such systems are: ROOMS, Henderson and Card, 1986; CUBRICON, Funke et al., 1993; AWM, Stille et al., 1997; MaW3-3D Window Manager, Leach et al. 1997; and Elastic Windows, Kandogan and Shneiderman, 1996. The only exception is the Elastic Windows system where evaluation was carried out. However the Elastic Windows System was designed to optimise user performance during task setup therefore a direct comparison between the system and Vanishing Windows cannot be made. However their evaluation results do concur with the findings of this thesis, that as the number of windows displayed increases, higher performance differences between adaptive and non-adaptive window management are observed.

There is still potential for further research in the area of adaptive window managers. As a result of the work performed in this thesis, research in window management can advance further by comparing the system and techniques developed as part of this thesis to any future window management system. The thesis has developed concepts, built a working prototype and has presented an extensive evaluation of the system.

8.3.1 Possible Functionality for Future Versions of Vanishing Windows

Experience with the current system has suggested some possible future improvements. When vanishing for a window has ceased, the window could then
begin to expand again, to occupy some of the unused screen space. All four edges of the window would begin to expand simultaneously and as each edge hits other windows or reaches the edge of the desktop window. Expansion for this edge halts, all other edges continue to expand until no edges can expand further.

For the system to be usable and to be acceptable by users, manual overriding of adapting process should be provided. Some of the manual override facilities that could be provided are:

- Prevent an individual window from vanishing or expanding by selecting an option.
- Prevent an individual window from adapting the toolbars
- Prevent an individual window from adapting the contents of the document by highlighting keywords

These may be prevented at the individual window level or for the whole system. For example, a user should be able to specify that no toolbar adaptation is to take place but that vanishing and adapting the contents of the document should be active.

8.3.2 Potential Applications

Future Window Managers could employ the techniques developed for the Vanishing Windows system. As a first stage, almost any Window Manager can provide hooks for application developers to adaptively remove and replace toolbars as the applications become inactive and active.

Apart from Vanishing Windows being an intelligent Window Manager, the technique for resizing and migrating windows can be applied at the application level. For example the Visual Basic programming environment tends to use many windows - windows that are designed by the application developer, or windows that are part of the development environment such as toolbox, property window, project
window. The Vanishing process can be applied so that screen real estate is managed by the system.

Consider another application where a terminal displays in separate windows current news stories which are being reported from the field. As a new story arrives at the terminal, this becomes the most important story, as time passes and other stories arrive at the terminal, older stories (windows) begin to shrink in the background. This scenario can also apply for a stock market analyst workstation. These are speculative applications, which would require further exploration to merit usefulness of the vanishing techniques on these applications.

8.4 Further Research

Lane et al. (1997) identified two areas of future work:

- Further experimentation is needed to determine to what extent the "Cluttered Desktop" increases the completion time.
- Further research is needed to see whether independent variables such as number of windows opened, percentage of windows hidden, and frequency of subtasks, will affect user completion times and error rates.

The first area of work has been addressed by this thesis and has shown that as the number of windows displayed increases so does the time to located a particular window containing a document (see chapters 6 and 7). However, the second area of work has not been addressed in this thesis and is potential area for future work.

8.4.1 Addressing Problems of System Limitations

There are some limitations of the system, which need to be addressed in order to create a viable commercial solution. They are:

- System Performance
- Online extraction of Keywords
• Additional Functionality to Constrain Windows from Vanishing

The Vanishing Windows System has been shown to improve user performance. However, the performance of the system needs to be improved. Considerable processing is required to determine valid and useful manoeuvres which windows could deploy. This slows the system, therefore algorithms need to be streamlined to achieve greater system performance.

The current implementation of the keyword extraction is performed off line. The system can be further developed to extract keywords online. Again system performance needs to be addressed.

The current Vanishing Windows system assumes that a window is in use when it is active (i.e. has control of the mouse and keyboard) however, windows displaying information currently required can also constitute as the window being currently in use. Two examples of a window begin used in such a manner are: (1) A Web browser is set to search the Web for some information whilst this is progressing the user wishes to monitor the progress of the Web search and continue working on other tasks. Thus, both the task window and the Web browser windows are being used. (2) Items from one window are being compared or copied from one window to another window. Under both conditions the user must be able to specify that they are being used therefore preventing from any automatic window manipulations. These constraints must be easily specified and intuitive methods for specification developed. One method that can be used is to ‘pin windows on the desktop’ as used by Sun’s Window Manager – Open Look. This method allows child windows to be pinned onto the desktop so that they are easily accessible.

All windows vanished at the same rate. There is potential for further research in varying the vanishing rate for each window. This rate can be determined by the original window size. One possible idea may be to reduce large windows at a slower rate than small windows. The assumption being that a larger window is of more
value to the user and may like it stay visible for longer. The assumption need to be validated and the idea of variable vanishing rate evaluated.

8.4.2 Extend the Evaluation

Evaluating Vanishing Strategies

Two strategies for moving and shrinking windows were outlined in Chapter 4 - the self-centred strategy and the co-operative strategy. Only the self-centred strategy was implemented. Further work can be carried out to implement the co-operative strategy and evaluation carried out to determine efficiency of these two strategies. Efficiency can be measured both in terms of the computations required and how effective is the final layout of the screen. Deviation from its original size and position can be used to measure the efficiency of layout since the system should minimise the deviation, so as not to confuse users' positional information regard location of a window.

Extensive Evaluation of Auto-Adaptation

Further experimentation can be carried out to determine to what extent will users accept auto-adaptation.

- What are the limits, conditions and scenarios in which auto-adaptation fails?
- How would users react to an increase in the size of the active window to take up space which has been released by inactive windows vanishing

System Learns Users Window Usage Patterns

The system can be extended further so that it begins to learn window usage patterns, such as window layouts.
REFERENCES
References


Card S.K. (1985). *Windows: Why they were invented, How they help*. The Office, March 1985, pp. 52-54,


Collins English Dictionary Paper back, ISBN 0 00 433245 8


APPENDICES
Appendix A. Ancillary Information For Chapter 3

A.I. Task Sheet for Observation of Windows Usage

Introduction

The task is centred on a scenario where a number of Windows applications are being used. The objective of the experiment is to gather general information on the ways in which windows are being used for particular tasks. In particular how Windows are manipulated in order to carry out certain tasks. The test subject is encouraged to speak clearly into the microphone, commenting on what they are doing, in particular reasons for manipulating windows around [re-arranging, minimising, maximising, shrinking, closing applications, and so].

The experiment consists of 22 sub-tasks. Each sub-task is small enough and not too complex to be completed in a few minutes.

Courier text indicates the names of directories; files, menu options or key presses that you should use. Boxed Courier text should be typed in. The formatting of the text (bold, Italics, font size etc.) is left up to you.

The Experiment

You are an employee in a large comic book store. You are to produce a report to your manager explaining the sudden increase in sales and any proposals to help capitalise on this upturn. To achieve this, do the following:

Task 1

(1) Open the File Manager
(2) Create a new directory called c:\report on the PC hard drive to store the report information.
(3) Open Microsoft Word
(4) Enter the following text:
Sales Report
From:
Loughborough University Store
Haslegrave Building
Loughborough
Leicestershire

To: Mr Joe Kerr
Comic Shop Proprietor
Arkham Asylum
Gotham City
Gotham

Dear Manager,
This short report is intended to give you a brief summary of the sales of comic books in the Loughborough University store. As you are well aware, the recent refurbishment of the store has helped to increase the range of products that we can stock. This extended range has improved the sales considerably.

Another explanation of the upturn in the project may be the extensive advertising campaign conducted in the area. The sales started to pick up during August, this may be due to the influx of students arriving in Loughborough to attend the university.

(5) You should format the above text to appear as professional as possible.
After formatting, save the document as salesrep.doc in the c:\report directory created earlier.

Task 2
You then decide to include a diagram of the new comic shop layout in the report.

(7) Open Microsoft PowerPoint application from the Microsoft office group.
(8) Create a blank new presentation.
(9) Use PowerPoint to create a presentation to look like:
(10) Save the presentation in the c:\report directory with filename planview.ppt.

(11) Activate the Word application.

(12) You should now include the diagram between the first and second paragraphs of the report. To do this you should select the Object item on the Insert menu, and use the Create From File tab to select the planview.ppt file (in the c:\report directory). The diagram will then appear in the document.

Task 3

You then decide to include some sales figures in the report.

(13) Open sales.doc located in c:\salesdata\sales.doc, which contain the sales figures.

(14) Some of the monthly total has not been calculated. Open up the calculator application (Accessories Group) and calculate the monthly totals. Save the document.

(15) Open up Excel (located in Microsoft Office Group) and produce a table as shown below. Use the monthly sales figures from the sales.doc document for the whole year (Jan-Dec).

<table>
<thead>
<tr>
<th></th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>June</th>
<th>July</th>
<th>Aug</th>
</tr>
</thead>
<tbody>
<tr>
<td>Superman</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Batman</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spiderman</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(16) Save the workbook as sales.xls in c:\report directory.
(17) Use the **Insert, Chart, On This Sheet** menu selection to create a chart on the spreadsheet. The chart should look something like the one shown below. The data for the chart below is not the actual data to be used in this experiment. Actual data can be found in `sales.doc`, used previously.

![Chart](image)

**Figure 1**

(18) Save the spreadsheet then copy the chart to the clipboard by highlighting it and pressing **CTRL-C** or use the edit menu to copy.

(19) Go to the Word application and paste the chart (**CTRL-V** or use the edit menu to paste) at the bottom of the document.

(20) Add the following text underneath the chart.

> As can be seen by the chart, the Superman and Batman titles had a small increase in sales during the month of August. Only the Mature Reader Spiderman imprint has sustained a gradual increase in monthly sales.

**Task 4**

Finally, you have decided to include a chart showing the quarterly sales figures.

(21) Using the monthly sales figures from `c:\saledata\sales.doc` calculate the quarterly sales figures using the calculator application and enter the data in a new Excel Worksheet called `qsales.xls`. Save the excel document in `c:\report\qsales.xls`
(22) Produce graph, save excel document and copy graph to report document.

The task has been completed, thank you for your time.
A.2. Questionnaire to Collect User Details

QUESTIONNAIRE

SUBJECT NUMBER:

Have you used Microsoft Windows 3.1?

☐ Yes  ☐ No

If Yes, How FREQUENTLY?

☐ Every Day  ☐ Couple of Times a week  ☐ Couple Times a month

Have you used Microsoft Word? And if so how frequently do you use it?

☐ Every Day  ☐ Couple of Times a week  ☐ Couple Times a month

☐ Microsoft Excel

☐ Every Day  ☐ Couple of Times a week  ☐ Couple Times a month

☐ Microsoft Power Point

☐ Every Day  ☐ Couple of Times a week  ☐ Couple Times a month
A.3. Transcribed Videotaped Data

Subject: Tim Appleyard

Open (Presentation Manager, 00.00)
  Open (Main)
    Open (File Manager)
      Task (Create Directory, ST 00.10)
    Exit (File Manager, 00.27)
  Close (Main)
  Open (MSOffice Group)
    Open (MSWord, 00.37, N-Max)
      Task (Type Text, ST 00.38)
    Min (MSWord, 5.06, Min)

Open (PowerPoint, 5.41) couple of failed attempts previously
  Task (Draw Diagram, 5.42)
Close (PowerPoint, 18.08)
Restore (MSWord, 18.12, Max)
  Task (Insert PP Diagram, CT 19.39)
  Task (Open Doc, 19.40) {open another doc in word – sales doc}
Alt-Tab (Presentation Manager, 20.21)
  Min (MSOffice Group)
  Open (Accessories Group)
    Open (Calc, 2036)
  Min (Presentation Manager, 20.37)
    Move (Calc, 20.43, Top Right)

TASK EXECUTION PHASE

(This sequence is repeated until monthly total calculated – distinct layout)
Alt-Tab (MSWord)
Alt Tab (Calc)
  Task (Calc Months Total)
  Activate (MSWord)
  Task (Type Total)

Task (Completed, 25.08)

TASK SWITCHING PHASE

Min (MSWord, 25.11)
Close (Calc, 25.17)
Restore (Presentation Manager, 25.19) {retorted to normal}
  Close (PM – Accessories, 25.22)
  Open (PM – MS Apps, 25.28)
  Open (MSExcel, 25.38) {normal is maximised}

Task (MSExcel, Produce Table Headings)
Appendix

Alt-Tab (PM, 26.36)
Alt-Tab (MSWord, 26.36) \{two consecutive alt-tabs\}
  Task (MSWord, Copy Superman Sales Figures)
Alt-Tab (MSExcel 26.58)
  Task (MSExcel, Paste Superman Sales Figures)
Alt-Tab (MSWord, 27.18)
  Task (MSWord, Copy Batman Sales Figures)
Alt-Tab (MSExcel, 27.30)
  Task (MSExcel, Paste Batman Sales Figures)
Alt-Tab (MSWord, 27.36)
  Task (MSWord, Copy Spiderman Sales Figures)
Alt-Tab (MSExcel, 27.47)
  Task (MSExcel, Paste Spiderman Sales Figures)
  Task (MSExcel, Save File @ 27.57)
  Task (MSExcel, Generate Graph Completed @ 32.30) \{Some confusion experienced as to which data set is used to generate graph\}
  Task (MSWord, Paste Graph) \{Paste into sales.doc rather than report.doc\}
  Task (MSWord, Completed @ 34.33)

\textbf{TASK SWITCHING/ ENVIRONMENT SETUP PHASE} (sub task 21)

Min (MSWord, 34.34)
Min (MSExcel, 34.39) \{Screen shows Desktop with MSWord, MSExcel & PM all Minimised\}
Restore (MSWord, 34.42) \{i.e. restored to previous setting - maximised\}
  \{Time spent think about what to do next\}
Alt-Tab (MSExcel, 35.51) \{restored to previous setting - maximised\}

\textbf{TASK EXECUTION PHASE}

Task (MSExcel, Create new doc)
Task (MSExcel, Produce table headings)

\textbf{TASK ENVIRONMENT SETUP PHASE}

Alt-Tab (MSWord, 36.34, Normal Maximised)
Alt-Tab (Presentation Manager, -, Normal)
  Close (PM – MS Apps Group)
  Open (PM – Accessories Group)
  Open (Calc, 36.42)
Min (Presentation Manager, 36.44)
Move (Calc, 36.48) \{move to top right\}
Alt-Tab (MSExcel)
Alt-Tab (MSWord, 36.52)

\textbf{TASK EXECUTION PHASE}

Task (start time, 36.57)
Appendix

Repeat
Alt-Tab (Calc, 36.53) \(\text{layout MSWord maximised with Calc top right corner}\)
Task (Calc, Calculate Quarter Sales figure)
Alt-Tab (MSWord)
Alt-Tab (MSExcel)
Task (MSExcel, Type Quarter Total)
Until \(\text{Quarter 1, 2, 3 & 4 sales calculated}\)
Task (Completed, 40.33)
Task (MSExcel, Generate Graph)
Task (MSExcel, Copy Graph)

TASK SWITCHING/ENVIRONMENT SETUP PHASE

Min (MSExcel, 41.13) \(\text{Screen - MSWord Maximised, Calc top right active}\)
Close (Calc, 41.17) \(\text{leaves MSWord Maximised}\)
Task (MSWord, Paste Graph)
Task (MSWord, Completed @ 41.45) \(\text{discovers mistake}\)
Task (MSWord, correct mistake @ 43.55) \(\text{pasted into wrong doc sales.doc should be report.doc}\)

END.
Subject Number: 2-Diana Hawkay

Open (Program Manager, 00.00)
  Open (Main, 00.07)
    Open (File Manager, 00.13) {maximised}

TEP

  Task (Create Directory, COM 00.45)

TESP/ITXP

  Min (File Manager, 1.07)
  Open (MSOffice Group, 1.14)
    Open (MSWord, 1.17, N-Max)

TEP

  Task (MSWord Type Text, ST 1.27)
    Task (MSWord File Save As, 6.28)
    Task (MSWord File Close, 7.38) {*why close file?*}
    Close (MSWord, 7.54) {DT – PM Active normal, FM minimised}

TESP/ITXP

  Open (MS PowerPoint, 8.00) {normal maximised}
    Task (PP- Draw diagram, ST 8.20)
    Task (PP – File Save As, 20.53)
    Task (PP – File Close, 22.27)
  Exit (MS Power Point, 22.34) {DT – PM Active normal, FM minimised}

TESP/ITXP

  Open (MSWord, 22.44) {normal maximised}
    Task (MSWord – File open salesrep.doc, 23.12)
    Task (MSWord – Insert Object, 23.37)
    Task (MSWord – File Open sales.doc, 24.50)

TESP/TSP

  Min (MSWord, 25.42)
    {PM active – trying to open Calc, hidden by MS office group + Main group}
  Move (Program Manager, 26.02)
  Move (Program Manager, 26.04)
    Close (PM – MS Office Group, 26.23)
    Close (PM – Main Group, 26.28)
    {DT – PM all groups closed, FM & MSWord minimised}
  Open (PM- Accessories Group, 26.34)
    Open (Calc, 26.38)
  Restore (MSWord, 26.58) {normal max}
  UNMax (MSWord, 27.20)
  Move (MSWord, 27.23)
  Move (Calc, 27.28)
  Move (MSWord, 27.34)

TASK EXECUTION PHASE (Sub-Task 14)

  Repeat
    Task (Calc month total, ST 27.40)
Appendix

Activate (MSWord)
Task (Type Total)
Activate (Calc)
Until Superman monthly total calculated
Move (MSWord, 29.45) \{decided to move window back and forth\}
Move (MSWord, 29.48)
Repeat
  Task (Calc month total)
  Activate (MSWord)
  Task (Type Total)
  Activate (Calc)
Until Batman monthly total calculated

Activated (Program Manager, 30.43) \{Accidental activation\}

\{During calculation - accidentally switched to Program Manager (PM). PM became the top
window overlapping other required windows. The following sequence of operations were
required to correct the layout\}

Activated (Calc, 30.45) \{Now Calc top most window followed by PM, then MSWord – require
MSWord to be on top of PM\}

Activated (PM, 30.47)
Min (PM, 30.49) \{minimised PM to get it out of the way\}

\{Resumed to task i.e. Calc monthly total for remain months\}

Task (Calc monthly totals completed, 34.09)

TASK ENVIRONMENT SETUP PHASE

Move (MSWord, 34.18) \{To reveal PM\}
Max (PM, 34.28)
  Close (PM – Accessories Group, 34.33)
  Open (PM – MS Office Group, 34.40)
  Open (MS Excel, 34.45) \{normal = Maximised\}

TASK EXECUTION PHASE – Sub Task 15

Task (MSExcel Create table, 35.18)
Min (MSExcel, 36.46)
Min (PM, 36.53) \{Desktop – MSWord Active, Calc visible, Min – FM, PM, MSExcel\}

Move (MSWord, 36.56)
Move (MSWord, 37.20)
Move (MSWord, 37.35) \{moved to reveal Excel which is iconised\}
Restore (MSExcel, 37.37) \{restored to normal (max)\}
  Task (MSExcel – more typing)
UNMax (MSExcel, 38.09)
Activate (MSWord, 38.22)
Activate (MSExcel, 38.30)
Move (MSExcel, 38.33)
Activate (MSWord, 38.36)
  Task (MSWord - Copy Superman monthly total, 38.46)
Appendix

Activate (MSExcel, 38.53)
Task (MSExcel – Paste, 39.04)
Activate (MSWord, 38.22)
Task (MSWord – Copy Batman monthly total, 40.01)
Activate (MSExcel, 40.05)
Task (MSExcel – Paste, 40.09)
Activate (MSWord, 40.12)
Task (MSWord – Copy Spiderman monthly totals, 40.51)
Activate (MSExcel, 40.54)
Task (MSExcel – Paste, 40.57)

TASK SWITCH/ENVIRONMENT SETUP PHASE
Activate (MSWord, 41.00) {Switch between application for no apparent reasons}
Activate (MSExcel, 41.02) {but maybe to see what are on these windows}
Move (MSExcel, 41.11)
Min (MSExcel, 41.13) {DT – MSWord Active, minimised – FM, PM & MSExcel}
Move (MSWord, 41.19)
Restore (MSExcel, 41.24) {normal unmaximised}
Resize (MSExcel, 41.31) {for no apparent reason}
Max (MSExcel, 41.36)

TASK EXECUTION PHASE – Sub Task 16, 17, 18, 19 & 20
Task (MSExcel – File Save As, 41.48)
Task (MSExcel – Produce Graph, ST 43.01)
Task (MSExcel – File Save As, 44.32)
Task (MSExcel – Copy chart, 44.56)

Min (MSExcel, 45.03) {DT – MSWord Active (not Max), Calc inactive, Minimised – FM, PM & MSExcel}
Move (MSWord, 45.07) {to see what is behind it}

Task (MSWord – Window switch to Salesrep.doc, 45.20)
Task (MSWord – Paste Chart, 45.39)
Task (MSWord – Type text, 45.59)

TASK SWITCHING/ENVIRONMENT SETUP PHASE

Task (MSWord – Window switch to Sales.doc, 47.19)
Max (MSWord, 47.28)
Min (MSWord, 47.36)
Restore (MSExcel, 47.41) {restored to max}

TASK EXECUTION PHASE – Sub Task 21a (Produce Table
Task (MSExcel – File New, 47.50)
Task (MSExcel – Produce table, ST 48.19)
Task (MSExcel – Produce table, COM 48.59)

TASK ENVIRONMENT SWITCHING
Min (MSExcel, 49.00) {DT – Calc Active, Minimised – FM, MSExcel, PM, MSWord}
Restore (MSWord, 49.05) {Restore to Max}
Max (MSWord, 49.19)
Move (MSWord, 49.22)
Move (MSWord, 49.33)
Move (MSWord, 49.33) \{to show Calc\}
Activate (Calc, 49.35)
Move (MSWord, 49.38)
Move (MSWord, 49.49)
Restore (MSExcel, 49.54) \{Restored to Max\}
Max (MSExcel, 49.57)
Move (MSExcel, 49.59)
Activate (MSWord) \{unnecessary activation\}
Activate (MSExcel) \{unnecessary activation\}
Resize (MSExcel, 50.05)
Activate (MSWord, 50.08)
Move (MSWord, 50.13)
Move (MSWord, 50.22)
Resize (MSWord, 50.26)
Move (MSWord, 50.29)
Move (MSWord, 50.39)

TASK EXECUTION PHASE – Sub Task 21b (Calculate Quarterly figures)
Task (Calc – Add 3 months total, 50.47)
Activate (MSWord, 50.57)
Move (MSWord, 51.00)
Task (MSExcel – type total, 51.03)
Move (MSWord, 51.10)
Activate (Calc, 51.14) \{but Calc covers MSWord hence has to move word\}

\{decides to engage in another TASK ENVIRONMENT PHASE to improve layout\}
Move (MSWord, 51.17)
Move (MSWord, 51.22)
Move (MSWord, 51.43)
Activate (MSExcel, 51.44)
Resize (MSExcel, 51.46)
Move (MSWord, 51.50)
Move (Calc, 51.56)
Move (MSWord, 52.00)
Move (MSExcel, 52.02)
Move (MSWord, 52.04)
Resize (MSWord, 52.11)
Resize (MSWord, 52.14)
Resize (MSWord, 52.21)
Move (MSWord, 52.25)
Resize (MSWord, 52.32)
Move (MSWord, 52.39)

TASK EXECUTION PHASE – Sub Task 21b (Calculate Quarterly figures) RESUMED

Task (Resumed at, 52.50)
Repeat
Task (Calc – Add three-month total) \{occasionally switched to MSWord to scroll text\}
Activate (MSExcel) \{to show next 3 months figures\}
Task (MSExcel – Type total)
Activate (Calc)
Until (quarter 2,3 & 4 Calculated)
Task (MSEXcel – all quarter values calculated, COM 56.45)

TASK EXECUTION PHASE (Some SWITCHING) – Sub Task 22

Max (MSEXcel, 56.46)
  Task (MSEXcel – Produce chart, ST 57.01)
  Task (MSEXcel – File Save As, 57.52)
Max (MSEXcel, 57.58)
  Task (MSEXcel – Copy chart, 58.04)
Activate (MISWord, 58.09)
  Task (MISWord – Switch to salesrep.doc, 58.13)
  Task (MSEXcel – Edit menu selected, 58.23) {accidentally selected edit on MSEXcel}
Activate (MISWord, 58.26)
  Task (MISWord – Paste, 58.30)
  Task (MISWord – File Save As, 58.41)

END.
Appendix

Subject Number: 3-Garath Hayhurst

Open (Program Manager, 00.00) (unmax)
  Open (PM-Accessories group, 00.53) (looking for File Manager but this is in Main)
  Close (PM-Accessories group, 00.54)
  Open (PM-Main, 01.02)
  Open (File Manager, 01.06) (unmax)

TEP
  Task (Create Directory, COM 01.30)

TESP/TXP
  Activate (PM, 1.33)
  Close (PM-Main, 1.35) (DT-PM active unmax, FM inactive)
  Open (MSOffice Group, 1.43)
  Open (MSWord, 1.49, N-Unmax)

TEP
  Task (MSWord Type Text, ST)
  Task (MSWord File Save As, 6.36)

  Min (MSWord, 7.06) (DT-PM active with ms-office group open, FM inactive & MSWord min)
  Open (MS Powerpoint, 7.13, N-Unmax)
  Max (MS PowerPoint, 7.45)
    Task (PP - Draw diagram, ST)
    Task (PP - File Save As, 14.25)
  Min (MS Power Point, 14.52) (task complete DT - PM active, FM inactive, MSWord & PP min)

  Restore (MSWord, 14.55, N-Unmax)
  Task (MSWord – Insert Object, 16.04)

TESP/TSP (for Sub-Task 14)
  Max (MSWord, 16.48)
    AltTab (PM, 17.09) (DT MS-Word max inactive, PM active norm)
    Close (PM – MSWord Group, 17.11)
    Open (PM – Application, 17.14) (Opened but not the one looking for)
    Move (PM – Application, 17.16)
    Close (PM – Application, 17.16)
    Open (PM – Accessories, 17.19) (this is the one that should have been opened)
    Open (Calc, 17.22)
    Min (PM, 17.27) (Starts to tidy up windows)
    Move (Calc, 17.30)
    Move (Calc, 17.36)

TASK EXECUTION PHASE (for Sub-Task 14)
  Task (Calc – Add monthly figures start, 17.51)
  Task (MSWord – Type total, 18.15)

AltTab (Calc, 18.23)
Appendix

Task (Calc – Add monthly figures start, 18.24)
Task (MSWord – Type total, 18.39) (type 14 but this is the wrong value – had to remember value)

Cognitive load
AltTab (Calc, 18.51) (realised wrong value typed So switched back to check)

Task (MSWord – Type total, 18.53) (typed correct value)

AltTab (Calc, 18.58)
Task (Calc – Add monthly figures start, 18.59)
Task (MSWord – Type total, 19.12)

AltTab (Calc, 19.18)
Task (Calc – Add month 4 figures start, 19.19)
Task (MSWord – Type total, 19.47)

AltTab (Calc, 19.54)
Task (Calc – Add month 5 figures start, 19.55)
Task (MSWord – Type total, 20.11)

AltTab (Calc, 20.17)
Task (Calc – Add month 6 figures start, 20.18)
Task (MSWord – Type total, 20.37) (Superman Complete)

(Start monthly total for Batman)
AltTab (Calc, 20.43)
Activate (MSWord, 20.49) (to scroll down to show monthly total not calc)

AltTab (Calc, 21.03)

Task (Calc – Add month 1 figures ST, 21.04)
Task (MSWord – Type total, 21.18)

AltTab (Calc, 21.23)
Task (Calc – Add month 2 figures ST, 21.24)
Task (MSWord – Type total, 21.38)

AltTab (Calc, 21.41)
Task (Calc – Add month 3 figures ST, 21.42)
Task (MSWord – Type total, 21.55)

AltTab (Calc, 21.59)
Task (Calc – Add month 4 figures ST, 22.00)
Task (MSWord – Type total, 22.28)

AltTab (Calc, 22.33)
Task (Calc – Add month 5 figures ST, 22.34)
Task (MSWord – Type total, 22.46)

AltTab (Calc, 22.50)
Task (Calc – Add month 6 figures ST, 22.51)
Task (MSWord – Type total, 23.03) (Batman Completed)
Appendix

{Scroll down to Spiderman uncalculated monthly figures}

AltTab (Calc, 23.19)
Task (Calc – Add month 1 figures)
Task (MSWord – Type total, 23.31)

AltTab (Calc, 23.34)
Task (Calc – Add month 2 figures)
Task (MSWord – Type total, 23.47)

AltTab (Calc, 23.53)
Task (Calc – Add month 3 figures)
Task (MSWord – Type total, 23.57)

AltTab (Calc, 24.21)
Task (Calc – Add month 4 figures)
Task (MSWord – Type total, 24.37)

AltTab (Calc, 24.40)
Task (Calc – Add month 5 figures)
Task (MSWord – Type total, 24.57)

AltTab (Calc, 25.00)
Task (Calc – Add month 6 figures)
Task (MSWord – Type total, 25.25)

TASK ENVIRONMENT SETUP PHASE

Min (MSWord, 26.06) {Calc Active, FM inactive, PM, PP & MSWord minimised}
Close (Calc, 26.09)
Min (FM, 26.11) {All minimised PM, PP, MSWord, FM}
Restore (PM, 26.12) {Normal = UnMax}
Close (PM – Accessories group, 26.13)
Open (PM – MSOffice group, 26.18)

TASK EXECUTION PHASE – Sub Task 15

Open (MSExcel, 26.23) {Normal = UnMax}
Max (MSExcel, 26.25)

Task (subtask 15 ST, 26.25)

AltTab (PM, 26.43)
AltTab (MSWord, 26.44)
Task (MSWord Copy, 27.18)

AltTab (MSExcel, 27.20)
Task (MSExcel Paste, 27.23)

{Mistake made – Pasted on top line of Excel, i.e. written over header (Jan, Feb, etc.)
So had to retype}
Appendix

AltTab (MSWord, 28.25)  \{Copy and Paste Superman data\}
Task (MSWord - Copy)
AltTab (MSExcel, 28.39)
Task (MSExcel Paste, 28.46)

AltTab (MSWord, 29.15)  \{Copy and Paste Batman data\}
Task (MSWord - Copy, 29.38)
AltTab (MSExcel, 29.48)
Task (MSExcel Paste, 29.50)

AltTab (MSWord, 29.52)  \{Copy and Paste Spiderman data\}
Task (MSWord - Copy, 30.00)
AltTab (MSExcel, 30.02)
Task (MSExcel Paste, 30.04)

TASK EXECUTION PHASE – Sub Task 16, 17, 18, 19 & 20
Task (MSExcel FileSaveAs, 30.12)  \{Sub Task 16\}
Task (MSExcel – Produce Graph, ST 31.39)
Task (MSExcel – File Save As, 32.40)  \{Sub Task 17 completed\}
Task (MSExcel – Copy chart)

AltTab (MSWord, 32.56)
Task (MSWord – Switch Document, 33.08)
Task (MSWord – Paste Chart, 33.16)

\{Pasted chart to big so decided to undo and try again\}

AltTab (MSExcel, 33.34)
\{Reduced size and copy chart again\}

AltTab (MSWord, 33.59)
\{Paste into doc\}

Task (MSWord – Type text ST, 34.34)  \{Sub Task 20\}
Task (MSWord – File Save, 35.42)

TASK SWITCHING/ENVIRONMENT SETUP PHASE

Task (MSWord – Window switch to Sales.doc, 47.19)
Max (MSWord, 47.28)
Min (MSWord, 47.36)
Restore (MSExcel, 47.41) \{restored to max\}

TASK EXECUTION PHASE – Sub Task 21a (Produce Table)

AltTab (MSExcel, 36.13)
Task (MSExcel – File New, 36.18)
Task (MSExcel – Produce table COM, 37.02)

TASK ENVIRONMENT SWITCHING
AltTab (MSWord, 37.03)
Task (MSWord – switch to weekly sales doc, 37.11)

{13 alt tabs executed to check if Calc is there}

AltTab (PP, 37.34)
AltTab
AltTab
AltTab (PM, 37.37)  {Desktop – PM Active unmax, PP inactive max, some minimised}
   Close (PM – MSOffice group, 37.38)
   Open (PM – Accessories group, 37.40)
   Open (Calc, 37.43)

{Decides to much stuff Visible on Screen decides to min}
Min (PP, 37.48)
Min (PM, 37.50)  {Desktop – Calc Active, MSWord (Max) inactive}
Move (Calc, 37.51)

TASK EXECUTION PHASE – Sub Task 21b (Calculate Quarterly figures)

{Superman Q1}
Task (Calc – Add 3 months total, -)
AltTab, AltTab (MSExcel, 38.14)
Task (MSExcel – Type Total, -)

Repeat
   AltTab, AltTab (MSWord, -)
   AltTab, AltTab (Calc, -)
   Task (Calc – Add 3 months total, -)
   AltTab, AltTab (MSExcel, )
      Task (MSExcel – Type Total, -)
For (Q1, Q2)

AltTab, AltTab (MSWord, -)
AltTab, AltTab (Calc, -)

{However Q4 Not Visible – need to scroll}
Activate (MSWord, 39.07)  {Scroll in word to show Superman Q4}

AltTab, AltTab (Calc, 39.09)
Task (Calc, Add Sub Totals)
AltTab, AltTab (MSExcel, 39.20)
Task (MSExcel – Type Total, 39.23)

{Batman ST 39.35}
Repeat
   AltTab, AltTab (MSWord, -)
   {Scroll in word to show Batman data}
   AltTab, AltTab (Calc, -)
   Task (Calc – Add Total, -)
   AltTab, AltTab (MSExcel, -)
   Task (MSExcel – Type Total, -)
Appendix

For (Q1, Q2)

AltTab, AltTab (MSWord)
AltTab, AltTab (Calc)
Activate (MSWord, 40.44)
{Scroll in Word to show Q3 & Q4}

Repeat
AltTab, AltTab (MSWord, -)
{Scroll in word to show Batman data}
AltTab, AltTab (Calc, -)
Task (Calc – Add Total, -)
AltTab, AltTab (MSEXcel, -)
Task (MSEXcel – Type Total, -)

For (Q3, Q4)
{Spiderman ST 41.49}
Repeat
AltTab, AltTab (MSWord, -)
{Scroll in word to show Spiderman data}
AltTab, AltTab (Calc, -)
Task (Calc – Add Total, -)
AltTab, AltTab (MSEXcel, -)
Task (MSEXcel – Type Total, -)

For (Q1, Q2)

AltTab, AltTab (MSWord, -)
{Scroll to show Spiderman Q3 & Q4 data}
AltTab, AltTab (Calc, -)
Task (Calc – Add Totals, -)
AltTab, AltTab (MSEXcel, -)
Task (MSEXcel – Type Total, -) {However, wrong value typed – cognitive load}

AltTab, AltTab (MSWord, 43.05)
{Forgot Which Q need to be Calculated so switches back}
AltTab (MSEXcel, 43.11)
AltTab (MSWord, 43.14)
AltTab, AltTab (Calc, 43.17)

Task (Calc – add totals for Q3 again, -) {Corrects previous mistakes}
AltTab, AltTab (MSEXcel, 43.30)
Task (MSEXcel – Type Total, -)

AltTab, AltTab (MSWord, 43.34)
AltTab, AltTab (Calc, 43.36)
Task (Calc, Add totals for Q4, -)
AltTab, AltTab (MSEXcel, 43.54)
Task (MSEXcel – Type Total, -)

Task (MSEXcel – all Q Values Calculated, COM 44.12)

{Sub Task 22 – Produce graph}
Task (MSEXcel – Produce graph, 44.19)
Task (MSEXcel – FilaSaveAs, 44.50)
Task (MSExcel – Copy, 45.03)
AltTab, AltTab (MSWord, 4511)
Task (MSWord – Switch doc, 45.23)
Task (MSWord – Paste, 45.29)
Task (MSWord – FileSave, 45.33)

END.
Appendix B. Ancillary Information for Chapter 5

B.1. Objective, and Instruction for Subjects for Experiment 1: Cropped Verses Scaled Presentation Styles

OBJECTIVE
The primary objective of the experiment is to determine the effect window size and presentation style has on a subject's ability to recognise a given document. This document is presented in a window in one of two possible styles (Cropped or Scaled).

INSTRUCTION FOR SUBJECT
You will be given a document to read for 10 minutes. Familiarise yourself with the document – get a feel for what the subject matter and the contents of the document.

You will then be presented with a number of screens serially. Each time a window containing a word document will appear on the screen at a pseudo random size. Without resizing, moving or scrolling the window you simply answer to the following question:

Is this the document that I was asked to familiarise myself with?

Three buttons are presented on the top of the screen – YES, NO, Don’t Know

Press

YES – if it is the document

No – if it is Not the document

Don’t Know – if you don’t know if it is the document for whatever reason

Your response and the response times are being logged. You are allowed to refer back to the document but this will effect you response time. You are allowed to withdraw from the experiment at any time.

While conducting the experiment can you please THINK ALOUD about why you think it is the document, it is not the document or you were unable to say (don’t know). For example not enough information – yes I recognise..., no I don’t think it was ... and so on.
Appendix C. Ancillary Information for Chapter 6

C.1. Objective, and Instruction for Subjects for Experiment 2: Adapted Presentation Styles

**OBJECTIVE**
Searching for a particular window in screen containing a number of windows can be rather difficult. Users tend to recognise content of the document, but usual the whole content of the document can not be recalled. However, users will generally tend to remember the subject of the document and possibly begin to recognise some of the words that can be used to identify and distinguish the document from other documents. These unique words are what we refer to as being keywords.

To aid recognition of a document, the document can be adapted so that the keywords are highlighted within the document. By highlighting the keywords, users should be able scan quickly through the highlighted words to determine whether this is the target document. The document can also be adapted using a number of other techniques, such as, presenting all the keywords as a list or removing all the non keywords from the document and replacing them with spaces. The document will then only contain keywords, spaces and more keywords. The spaces can be compressed hence packing the keywords within a small locality, hence increasing the keyword density.

The aim of this experiment is to determine which one of the possible four techniques for adapting the document yields higher recognition. The four possible techniques are Normal, List, Compressed and List.

**INSTRUCTION FOR SUBJECT**
You will be given a document to read for 10 minutes. Familiarise yourself with the document - get a feel for what the subject matter and the contents of the document. In particular try to identify words, that will uniquely identify that particular document.

There are two parts to the experiment. Part A, only presents a single window and Part B, displays 2,3, or 4 windows from which the subject has to identify the target document.

**Part A**
You will then be presented with a number of screens serially. Each time a window containing a word document will appear on the screen at a pseudo random size. Without resizing, moving or scrolling the window you simply answer to the following question:

Is this the document that I was asked to familiarise myself with?

Three buttons are presented on the top of the screen – YES, NO, Don’t Know
Press YES – if it is the document

No – if it is Not the document

Don’t Know – if you don’t know if it is the document for whatever reason

Part B

Each time a sample screen is present, click on the window you think is the target document and then press the Next Trial Button. If you don’t know, which one is the target document then press the ‘don’t know’ button.

For both parts of the experiment, your response and the response times are being logged. You are allowed to refer back to the document but this will effect your response time. You are allowed to withdraw from the experiment at any time.

While conducting the experiment can you please THINK ALOUD about why you think it is the document, it is not the document or you were unable to say (don’t know). For example not enough information – yes I recognise …, no I don’t think it was … and so on.
C.2. Data Exploration: Descriptive Statistics for Experiment 2 – Accuracy of Identification

Table 1 presents the basic descriptive statistics for the number of correct identification of the target document, across all screen utilisation groups. Statistics are shown for each of the presentation styles (Compressed, Highlighted, List and Normal).

<table>
<thead>
<tr>
<th></th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>4.14</td>
<td>3.78</td>
<td>3.82</td>
<td>3.81</td>
</tr>
<tr>
<td><strong>Median</strong></td>
<td>5.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
</tr>
<tr>
<td><strong>Std Deviation</strong></td>
<td>1.20</td>
<td>1.27</td>
<td>1.28</td>
<td>1.27</td>
</tr>
<tr>
<td><strong>Sample Variance</strong></td>
<td>1.45</td>
<td>1.61</td>
<td>1.63</td>
<td>1.61</td>
</tr>
<tr>
<td><strong>Kurtosis</strong></td>
<td>2.66</td>
<td>1.48</td>
<td>2.04</td>
<td>1.26</td>
</tr>
<tr>
<td><strong>Skewness</strong></td>
<td>-1.73</td>
<td>-1.42</td>
<td>-1.58</td>
<td>-1.27</td>
</tr>
<tr>
<td><strong>Minimum</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Maximum</strong></td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 1: Descriptive Statistics for Number of Correct Identification of Target Document for the four Presentation Styles, across all Screen Utilisation Groups

The remaining tables in section C.2. present descriptive statistics for the number of correct identification of target document, grouped by screen utilisation percentage.

<table>
<thead>
<tr>
<th>Screen Utilisation % 0-9%</th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>1.89</td>
<td>1.06</td>
<td>1.28</td>
<td>1.44</td>
</tr>
<tr>
<td><strong>Median</strong></td>
<td>2.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.50</td>
</tr>
<tr>
<td><strong>Mode</strong></td>
<td>2.00</td>
<td>1.00</td>
<td>1.00</td>
<td>2.00</td>
</tr>
<tr>
<td><strong>Std Deviation</strong></td>
<td>1.45</td>
<td>0.87</td>
<td>1.41</td>
<td>1.30</td>
</tr>
<tr>
<td><strong>Sample Variance</strong></td>
<td>2.11</td>
<td>0.76</td>
<td>1.98</td>
<td>1.67</td>
</tr>
<tr>
<td><strong>Kurtosis</strong></td>
<td>-0.35</td>
<td>1.36</td>
<td>0.28</td>
<td>2.03</td>
</tr>
<tr>
<td><strong>Skewness</strong></td>
<td>0.35</td>
<td>1.08</td>
<td>1.16</td>
<td>1.06</td>
</tr>
<tr>
<td><strong>Minimum</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Maximum</strong></td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 2: Descriptive Statistics for Number of Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 0-9%)
### Appendix

#### Screell UtilisaEion [%

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>10-19%</th>
<th>Presentation Style</th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>3.56</td>
<td></td>
<td>3.72</td>
<td>3.56</td>
<td>3.11</td>
<td></td>
</tr>
<tr>
<td><strong>Median</strong></td>
<td>4.00</td>
<td></td>
<td>4.00</td>
<td>4.00</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td><strong>Mode</strong></td>
<td>4.00</td>
<td></td>
<td>5.00</td>
<td>4.00</td>
<td>2.00</td>
<td></td>
</tr>
<tr>
<td><strong>Std Deviation</strong></td>
<td>1.29</td>
<td></td>
<td>1.45</td>
<td>1.39</td>
<td>1.13</td>
<td></td>
</tr>
<tr>
<td><strong>Sample Variance</strong></td>
<td>1.67</td>
<td></td>
<td>2.10</td>
<td>1.79</td>
<td>1.28</td>
<td></td>
</tr>
<tr>
<td><strong>Kurtosis</strong></td>
<td>-0.17</td>
<td></td>
<td>-0.76</td>
<td>0.20</td>
<td>-0.75</td>
<td></td>
</tr>
<tr>
<td><strong>Skewness</strong></td>
<td>-0.87</td>
<td></td>
<td>-0.76</td>
<td>-1.04</td>
<td>0.03</td>
<td></td>
</tr>
<tr>
<td><strong>Minimum</strong></td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td><strong>Maximum</strong></td>
<td>5</td>
<td></td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>4</td>
<td></td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3:** Descriptive Statistics for Number of Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 10-19%)

#### Screell UtilisaEion [%

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>20-29%</th>
<th>Presentation Style</th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>4.33</td>
<td></td>
<td>4.22</td>
<td>3.94</td>
<td>4.11</td>
<td></td>
</tr>
<tr>
<td><strong>Median</strong></td>
<td>4.00</td>
<td></td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td><strong>Mode</strong></td>
<td>5.00</td>
<td></td>
<td>5.00</td>
<td>4.00</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td><strong>Std Deviation</strong></td>
<td>0.69</td>
<td></td>
<td>1.00</td>
<td>0.54</td>
<td>0.96</td>
<td></td>
</tr>
<tr>
<td><strong>Sample Variance</strong></td>
<td>0.47</td>
<td></td>
<td>1.00</td>
<td>0.29</td>
<td>0.93</td>
<td></td>
</tr>
<tr>
<td><strong>Kurtosis</strong></td>
<td>-0.58</td>
<td></td>
<td>5.74</td>
<td>1.21</td>
<td>6.02</td>
<td></td>
</tr>
<tr>
<td><strong>Skewness</strong></td>
<td>-0.55</td>
<td></td>
<td>-2.07</td>
<td>-0.07</td>
<td>-2.02</td>
<td></td>
</tr>
<tr>
<td><strong>Minimum</strong></td>
<td>3</td>
<td></td>
<td>1</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td><strong>Maximum</strong></td>
<td>5</td>
<td></td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>2</td>
<td></td>
<td>4</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**Table 4:** Descriptive Statistics for Number of Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 20-29%)

#### Screell UtilisaEion [%

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>30-39%</th>
<th>Presentation Style</th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>4.06</td>
<td></td>
<td>4.22</td>
<td>3.89</td>
<td>4.72</td>
<td></td>
</tr>
<tr>
<td><strong>Median</strong></td>
<td>4.00</td>
<td></td>
<td>4.00</td>
<td>4.00</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td><strong>Mode</strong></td>
<td>4.00</td>
<td></td>
<td>4.00</td>
<td>4.00</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td><strong>Std Deviation</strong></td>
<td>0.54</td>
<td></td>
<td>0.55</td>
<td>0.83</td>
<td>0.58</td>
<td></td>
</tr>
<tr>
<td><strong>Sample Variance</strong></td>
<td>0.29</td>
<td></td>
<td>0.30</td>
<td>0.69</td>
<td>0.33</td>
<td></td>
</tr>
<tr>
<td><strong>Kurtosis</strong></td>
<td>1.21</td>
<td></td>
<td>0.18</td>
<td>1.74</td>
<td>3.85</td>
<td></td>
</tr>
<tr>
<td><strong>Skewness</strong></td>
<td>0.07</td>
<td></td>
<td>0.16</td>
<td>-1.15</td>
<td>-2.07</td>
<td></td>
</tr>
<tr>
<td><strong>Minimum</strong></td>
<td>3</td>
<td></td>
<td>3</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td><strong>Maximum</strong></td>
<td>5</td>
<td></td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>2</td>
<td></td>
<td>2</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5:** Descriptive Statistics for Number of Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 30-39%)
### Table 6: Descriptive Statistics for Number of Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 40-49%)

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>Presentation Style</th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>40-49%</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>4.44</td>
<td>4.28</td>
<td>3.83</td>
<td>4.17</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>5.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>5.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>Std Deviation</td>
<td>0.78</td>
<td>0.46</td>
<td>0.62</td>
<td>0.51</td>
<td></td>
</tr>
<tr>
<td>Sample Variance</td>
<td>0.61</td>
<td>0.21</td>
<td>0.38</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td>Kurtosis</td>
<td>-0.45</td>
<td>-0.94</td>
<td>4.49</td>
<td>0.92</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>-1.03</td>
<td>1.09</td>
<td>-1.59</td>
<td>0.32</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

### Table 7: Descriptive Statistics for Number of Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 50-59%)

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>Presentation Style</th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>50-59%</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>4.56</td>
<td>4.50</td>
<td>4.06</td>
<td>3.78</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>5.00</td>
<td>5.00</td>
<td>4.00</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>5.00</td>
<td>5.00</td>
<td>4.00</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>Std Deviation</td>
<td>0.62</td>
<td>0.62</td>
<td>0.42</td>
<td>0.59</td>
<td></td>
</tr>
<tr>
<td>Sample Variance</td>
<td>0.38</td>
<td>0.38</td>
<td>0.17</td>
<td>0.30</td>
<td></td>
</tr>
<tr>
<td>Kurtosis</td>
<td>0.39</td>
<td>-0.10</td>
<td>4.30</td>
<td>6.36</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>-1.09</td>
<td>-0.84</td>
<td>0.47</td>
<td>-2.57</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

### Table 8: Descriptive Statistics for Number of Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 60-69%)

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>Presentation Style</th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>60-69%</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>4.67</td>
<td>3.89</td>
<td>4.79</td>
<td>4.56</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>5.00</td>
<td>4.00</td>
<td>5.00</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>5.00</td>
<td>4.00</td>
<td>5.00</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>Std Deviation</td>
<td>0.49</td>
<td>0.47</td>
<td>0.43</td>
<td>0.71</td>
<td></td>
</tr>
<tr>
<td>Sample Variance</td>
<td>0.24</td>
<td>0.22</td>
<td>0.18</td>
<td>0.50</td>
<td></td>
</tr>
<tr>
<td>Kurtosis</td>
<td>-1.60</td>
<td>2.16</td>
<td>0.14</td>
<td>0.65</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>-0.77</td>
<td>-0.45</td>
<td>-1.46</td>
<td>-1.35</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
### Table 9: Descriptive Statistics for Number of Correct Identification of Target Document for the Four Presentation Styles (Screen Utilisation 70-79%)

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>Compressed</th>
<th>Presentation Style</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>4.94</td>
<td>3.83</td>
<td>4.72</td>
<td>4.72</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>5.00</td>
<td>4.00</td>
<td>5.00</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>5.00</td>
<td>4.00</td>
<td>5.00</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>Std Deviation</td>
<td>0.24</td>
<td>0.79</td>
<td>0.58</td>
<td>0.67</td>
<td></td>
</tr>
<tr>
<td>Sample Variance</td>
<td>0.06</td>
<td>0.62</td>
<td>0.33</td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td>Kurtosis</td>
<td>18.00</td>
<td>0.52</td>
<td>3.85</td>
<td>3.90</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>-4.24</td>
<td>-0.50</td>
<td>-2.07</td>
<td>-2.30</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

### Table 10: Descriptive Statistics for Number of Correct Identification of Target Document for the Four Presentation Styles (Screen Utilisation 80-89%)

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>Compressed</th>
<th>Presentation Style</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>4.83</td>
<td>4.17</td>
<td>3.78</td>
<td>3.72</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>5.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>5.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>Std Deviation</td>
<td>0.38</td>
<td>0.71</td>
<td>0.55</td>
<td>0.57</td>
<td></td>
</tr>
<tr>
<td>Sample Variance</td>
<td>0.15</td>
<td>0.50</td>
<td>0.30</td>
<td>0.33</td>
<td></td>
</tr>
<tr>
<td>Kurtosis</td>
<td>2.04</td>
<td>-0.78</td>
<td>0.18</td>
<td>-0.26</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>-1.96</td>
<td>-0.25</td>
<td>-0.16</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

### Table 11: Descriptive Statistics for Number of Correct Identification of Target Document for the Four Presentation Styles (Screen Utilisation 90-100%)

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>Compressed</th>
<th>Presentation Style</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>4.78</td>
<td>4.00</td>
<td>4.78</td>
<td>4.50</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>5.00</td>
<td>4.00</td>
<td>5.00</td>
<td>4.50</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>5.00</td>
<td>4.00</td>
<td>5.00</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>Std Deviation</td>
<td>0.55</td>
<td>0.59</td>
<td>0.55</td>
<td>0.51</td>
<td></td>
</tr>
<tr>
<td>Sample Variance</td>
<td>0.30</td>
<td>0.35</td>
<td>0.30</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td>Kurtosis</td>
<td>6.36</td>
<td>8.50</td>
<td>6.36</td>
<td>-2.27</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>-2.57</td>
<td>-1.89</td>
<td>-2.57</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 9, Table 10, and Table 11 provide descriptive statistics for the number of correct identifications of target documents across four presentation styles, categorized by screen utilisation percentages of 70-79%, 80-89%, and 90-100%, respectively.
C.3. Data Exploration: Descriptive Statistics for Experiment 2 – Response Time for Correct Identification

Table 12 presents the basic descriptive statistics for response time for correct identification of the target document, across all screen utilisation groups. Statistics are shown for each of the presentation styles (Compressed, Highlighted, List and Normal).

<table>
<thead>
<tr>
<th>Presentation Style</th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>11.40</td>
<td>25.00</td>
<td>12.40</td>
<td>22.25</td>
</tr>
<tr>
<td>Minimum</td>
<td>.60</td>
<td>.00</td>
<td>.00</td>
<td>.75</td>
</tr>
<tr>
<td>Maximum</td>
<td>12.00</td>
<td>25.00</td>
<td>12.40</td>
<td>23.00</td>
</tr>
<tr>
<td>Mean</td>
<td>4.0117</td>
<td>4.5379</td>
<td>4.0354</td>
<td>4.7002</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>2.2252</td>
<td>2.9228</td>
<td>2.4144</td>
<td>2.8217</td>
</tr>
<tr>
<td>Variance</td>
<td>4.952</td>
<td>8.543</td>
<td>5.829</td>
<td>7.962</td>
</tr>
<tr>
<td>Skewness</td>
<td>1.195</td>
<td>2.678</td>
<td>1.103</td>
<td>2.067</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>1.502</td>
<td>14.477</td>
<td>1.495</td>
<td>9.306</td>
</tr>
</tbody>
</table>

Table 12: Descriptive Statistics for Response Time for Correct Identification of Target Document for the four Presentation Styles, across all Screen Utilisation Groups

The remaining tables in section C.3. present descriptive statistics for response time for correct identification of target document, grouped by screen utilisation percentage.

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-9%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>7.00</td>
<td>25.00</td>
<td>11.00</td>
<td>11.00</td>
</tr>
<tr>
<td>Minimum</td>
<td>1.00</td>
<td>.00</td>
<td>.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Maximum</td>
<td>8.00</td>
<td>25.00</td>
<td>11.00</td>
<td>12.00</td>
</tr>
<tr>
<td>Mean</td>
<td>3.4007</td>
<td>4.3686</td>
<td>4.5833</td>
<td>4.7408</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>1.9071</td>
<td>6.3172</td>
<td>2.7784</td>
<td>3.6183</td>
</tr>
<tr>
<td>Variance</td>
<td>3.637</td>
<td>39.906</td>
<td>7.720</td>
<td>13.092</td>
</tr>
<tr>
<td>Skewness</td>
<td>1.173</td>
<td>3.047</td>
<td>.732</td>
<td>1.181</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>1.630</td>
<td>10.169</td>
<td>1.959</td>
<td>.169</td>
</tr>
</tbody>
</table>

Table 13: Descriptive Statistics for Response Time for Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 0-9%)
### Table 14: Descriptive Statistics for Response Time for Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 10-19%)

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>Presentation Style</th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-19%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>8.25</td>
<td>6.30</td>
<td>11.00</td>
<td>21.00</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>.75</td>
<td>1.20</td>
<td>1.00</td>
<td>2.00</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>9.00</td>
<td>7.50</td>
<td>12.00</td>
<td>23.00</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>3.6528</td>
<td>3.4194</td>
<td>3.7339</td>
<td>4.6989</td>
<td></td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>2.1932</td>
<td>2.0418</td>
<td>2.8385</td>
<td>4.9360</td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td>4.810</td>
<td>4.169</td>
<td>8.057</td>
<td>24.364</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>.901</td>
<td>.786</td>
<td>1.939</td>
<td>3.339</td>
<td></td>
</tr>
<tr>
<td>Kurtosis</td>
<td>.626</td>
<td>-.553</td>
<td>3.835</td>
<td>12.308</td>
<td></td>
</tr>
</tbody>
</table>

### Table 15: Descriptive Statistics for Response Time for Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 20-29%)

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>Presentation Style</th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>20-29%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>6.50</td>
<td>5.50</td>
<td>7.75</td>
<td>9.80</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>1.50</td>
<td>1.00</td>
<td>1.50</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>8.00</td>
<td>6.50</td>
<td>9.25</td>
<td>10.80</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>3.7978</td>
<td>3.9444</td>
<td>4.4806</td>
<td>4.0322</td>
<td></td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>1.7452</td>
<td>1.7524</td>
<td>2.2977</td>
<td>2.4363</td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td>3.046</td>
<td>3.071</td>
<td>5.280</td>
<td>5.935</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>1.325</td>
<td>-.382</td>
<td>.918</td>
<td>1.528</td>
<td></td>
</tr>
<tr>
<td>Kurtosis</td>
<td>2.130</td>
<td>-.943</td>
<td>.239</td>
<td>2.769</td>
<td></td>
</tr>
</tbody>
</table>

### Table 16: Descriptive Statistics for Response Time for Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 30-39%)

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>Presentation Style</th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>30-39%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>7.50</td>
<td>8.65</td>
<td>5.42</td>
<td>7.40</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>1.25</td>
<td>1.60</td>
<td>.25</td>
<td>1.20</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>8.75</td>
<td>10.25</td>
<td>5.67</td>
<td>8.60</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>3.4400</td>
<td>4.9378</td>
<td>2.8761</td>
<td>3.7289</td>
<td></td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>1.9367</td>
<td>2.0417</td>
<td>1.6601</td>
<td>2.1337</td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td>3.751</td>
<td>4.169</td>
<td>2.756</td>
<td>4.553</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>1.473</td>
<td>.830</td>
<td>-.092</td>
<td>1.317</td>
<td></td>
</tr>
<tr>
<td>Kurtosis</td>
<td>2.155</td>
<td>1.638</td>
<td>-.892</td>
<td>1.418</td>
<td></td>
</tr>
</tbody>
</table>
### Appendix

| Screen Utilisation % | Compressed | Presentation Style | | | Normal |
|----------------------|------------|--------------------|--------|--------|
| **40-49%** | **Range** | **10.40** | **6.67** | **8.50** |
| | **Minimum** | **.60** | **1.00** | **.75** |
| | **Maximum** | **11.75** | **7.67** | **9.25** |
| | **Mean** | **4.1556** | **4.0050** | **4.4400** |
| | **Std. Deviation** | **2.8404** | **2.1098** | **2.2201** |
| | **Variance** | **8.068** | **4.451** | **4.929** |
| | **Skewness** | **.707** | **.327** | **.459** |
| | **Kurtosis** | **-0.008** | **-1.042** | **-2.09** |

**Table 17:** Descriptive Statistics for Response Time for Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 40-49%)

| Screen Utilisation % | Compressed | Presentation Style | | | Normal |
|----------------------|------------|--------------------|--------|--------|
| **50-59%** | **Range** | **10.40** | **6.67** | **8.50** |
| | **Minimum** | **.60** | **1.00** | **.75** |
| | **Maximum** | **11.75** | **7.67** | **9.25** |
| | **Mean** | **4.1556** | **4.0050** | **4.4400** |
| | **Std. Deviation** | **2.8404** | **2.1098** | **2.2201** |
| | **Variance** | **8.068** | **4.451** | **4.929** |
| | **Skewness** | **.707** | **.327** | **.459** |
| | **Kurtosis** | **-0.008** | **-1.042** | **-2.09** |

**Table 18:** Descriptive Statistics for Response Time for Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 50-59%)

| Screen Utilisation % | Compressed | Presentation Style | | | Normal |
|----------------------|------------|--------------------|--------|--------|
| **60-69%** | **Range** | **8.25** | **11.20** | **9.80** |
| | **Minimum** | **1.00** | **1.20** | **1.80** |
| | **Maximum** | **9.50** | **12.40** | **11.60** |
| | **Mean** | **3.6611** | **5.3806** | **5.0778** |
| | **Std. Deviation** | **2.3129** | **3.4189** | **2.2913** |
| | **Variance** | **5.349** | **11.689** | **5.250** |
| | **Skewness** | **.963** | **1.117** | **.753** |
| | **Kurtosis** | **-2.332** | **-3.45** | **2.904** |

**Table 19:** Descriptive Statistics for Response Time for Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 60-69%)
<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>Presentation Style</th>
<th>Compressed</th>
<th>Highlighted</th>
<th>List</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>70-79%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>7.20</td>
<td>9.67</td>
<td>6.60</td>
<td>9.20</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>1.20</td>
<td>1.33</td>
<td>1.00</td>
<td>1.20</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>8.40</td>
<td>11.00</td>
<td>7.60</td>
<td>10.40</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>3.7444</td>
<td>5.3967</td>
<td>3.5583</td>
<td>5.0678</td>
<td></td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>1.6575</td>
<td>2.6322</td>
<td>2.0097</td>
<td>2.7130</td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td>2.747</td>
<td>6.929</td>
<td>4.039</td>
<td>7.361</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>1.237</td>
<td>.333</td>
<td>.843</td>
<td>.320</td>
<td></td>
</tr>
<tr>
<td>Kurtosis</td>
<td>2.585</td>
<td>-.359</td>
<td>-.126</td>
<td>-.701</td>
<td></td>
</tr>
</tbody>
</table>

Table 20: Descriptive Statistics for Response Time for Correct Identification of Target Document for the four Presentation Styles (Screen Utilisation 70-79%)
Appendix D. Ancillary Information for Chapter 7

D.1. Outline and Instruction for Subjects for Experiment 3: Empirical Study of Vanishing Windows

OBJECTIVE

Subjects are asked to complete a number of tasks using two windowing systems. The two systems are Vanishing Windows and a Non-Vanishing Windows such as Microsoft Window. Inactive Windows begin to un-overlap after a predefined time. This should reduce some of the clutter and enable the user to locate windows. When inactive windows are reactivated the window returns to its original size and location.

The aim of this experiment is to determine which one of the windowing system do users perform better in terms of task completion times. Both the times to locate a window and the time to complete each task are recorded. All user interactions are also recorded using Lotus Screen Cam Software.

INSTRUCTION FOR SUBJECT

Each task is in the form a question. For each task, the subject is asked to locate a document contained in a window and a question is asked. There are 30 questions in total, 15 questions are answered using Vanishing Windows system and 15 using the Non-Vanishing Windows system. The 15 questions are sub-divided into batches of 5. Each of the 5 questions is at different complexity levels. There are 3 complexity level, these are LOW, MEDIUM and HIGH. For each complexity levels different number of windows constitutes the search space. For example for the LOW complexity level, a total of 4 windows are present in the search space. For MEDIUM and HIGH complexity levels there are 6 and 8 windows in the search space respectively.

There are two buttons on the interface – “Found” and “Answered”. When the desired window has been located, make it the active window then press “Found”. Continue to answer the question and when answered press “Answered”.

You are allowed to move, resize and maximise windows to locate and answer the question.

You are allowed to use any of the menu items to answer the question.

You are not allowed to minimise or close any of the windows.
Appendix

D.2. Task Sheet for Experiment 3

Subject:
Interface Order: VWIN (2) WIN (1)
Task Order: LOW HIGH MEDIUM

Data Set 2 - LOW (Continue when you hear a beep)

Find the paper on Adaptive Toolbars.
Does the following word “least” appear in section 2.1, paragraph 2?
If the word appears in the section then select the word.

Find the paper titled “Vanishing Windows: Intelligent…”
In the “Conclusions and Future Work” section, select the 3rd sentence of 1st paragraph.
What is the sentence?

Find the paper titled “The Future of Advanced Communication…..”
Does section 2 have more diagrams than section 4?

Find the paper on Adaptive Toolbars.
• This is an example of a bullet point

How many bullet points (like the one shown above) are there in the document?

Find the paper titled “Video and Data Conferencing…..”
How many columns and row does table 1 have?

Data Set 2 - HIGH (Continue when you hear a beep)

Find the paper on “Challenges of Mobile Computing”
Does the paper contain any references to other papers by the authors of this paper?

Find the paper on “Failure Analysis by Cased Based Reasoning”
How many tables are there in the document?

Find the paper on “Recognition Accuracy and User Acceptance of Pen Interface”
How many tables and how many figures are there in section 4?

Find the paper on “Challenges of Mobile Computing”
What is the title of Table 3?
Find the paper on “Building Large Knowledge...”
   Who are the authors of reference 5?

Data Set 2 – MEDIUM (Continue when you hear a beep)

Find the paper on “Building Large Knowledge...”
   Does the following sentence appear in the section 3.1, paragraph 2?

   Many of the classes in the WFB can be naturally organised into taxonomies according to multiple attributes.

   If it does, select the sentence.

Find the paper on “Failure Analysis by Cased Based Reasoning”
   How many words are there in the last paragraph of section 2?

Find the paper titled “Vanishing Windows: Intelligent...”
   How many times does the word “intelligent” occur in the first paragraph of the Conclusions and Future work section?

Find the paper on “Building Large Knowledge...”
   How many sentences are there in last paragraph of the conclusion section?

Find the paper on Adaptive Toolbars.
   Equation 2 shows the equation for Toolbar creation time, what is the equation?
Appendix

SECTION 2

Data Set 1 – LOW (Continue when you hear a beep)

Find the paper on "Vanishing Windows: A Window Management Technique"
  • This is an example of a bullet point

How many bullet points (like the one shown above) are there in the document?

Find Paper Titled "Visual Recognition of Windows:...."
  Does the word "Contents" appear in section 3, paragraph 2?

Find the paper on "Video Conferencing: Hardware..."?
  In section 6.1.1, select the 2nd sentence of paragraph 1. What is the sentence?

Find the paper on "Multimedia Conferencing Architecture...."
  Does section 2 have a greater number of paragraphs than section 4?

Find Paper Titled "Visual Recognition of Windows:...."
  How many columns and row does table 4 have?

Data Set 1 – HIGH (Continue when you hear a beep)

Find Paper Titled "Mobile Worker: Access..."
  How many figures are there in the document?

Find Paper on Expert Systems for Threat Analysis In Radar Warning Receivers
  Does paragraph 8 have more words than paragraph 5?

Find the paper on WEB-Based Framework for Distributed Systems
  What is the title of figure 3 and select the title?

Find the paper on Mobile Worker
  In Section 3.4 what is the 2nd word of the 2nd sentence of the 2nd paragraph and select that word?
Appendix

Find the paper on "Wearable Computers: Application...."
How many references are there in the document and select the 5th reference?

Data Set 1 – MEDIUM (Continue when you hear a beep)

Find the paper on "Multimedia Conferencing Architecture...."
How many times does the word "good" occur in section 4, paragraph 5?

And select the last occurrence of "good".

Find the paper on "Vanishing Windows: A Window Management Technique"
How many words are there in section 1, paragraph 4?

Find the paper on "Expert Systems for Threat Analysis...."
Who are the authors of reference number 11?

Find the paper on "A Web-Based Framework...."
Does the paper contain any references to other papers by the authors of this paper?

Find the paper on "Video Conferencing: Hardware...."?
What is the title of section 8.2.2?
D.3. Questionnaire for Experiment 3

PART A: BACKGROUND INFORMATION

Age: (Optional) Subject Number:

Sex: ☐ male ☐ female

Studies/Occupation:

For how many years have you been using a computer?

☐ Less than one year ☐ Between one and three years
☐ More than three years

Have you used any versions of Microsoft Windows and for how long?

☐ No, have not used Microsoft Windows at all
☐ Less than 1 year
☐ Between 1 and 3 years
☐ More than 3 years

On average, how much time do you spend per week working on Microsoft Windows?

☐ Less than 1 hour
☐ Between 1 and 3 hours
☐ Between 4 and 10 hours
☐ Over 10 hours

PART B: ADAPTATION

Did you find the highlighted keywords on Inactive Windows helpful in identifying the document you were looking for?

☐ Very helpful ☐ Helpful ☐ No Difference ☐ Unhelpful ☐ Annoying

Did you find the removal of the toolbar on Inactive Windows helpful in identifying the document you were looking for?

☐ Very helpful ☐ Helpful ☐ No Difference ☐ Unhelpful ☐ Annoying

PART C: VANISHING WINDOWS

Did you find Vanishing Windows...

☐ Very helpful ☐ Helpful ☐ No Difference ☐ Unhelpful ☐ Annoying

Did you feel a loss of control when working with Vanishing Windows?

☐ Always ☐ Most of the time ☐ No Opinion ☐ Some Times ☐ Never

Did you find the automatic movement of windows (i.e. Vanishing) disruptive?
Always  Most of the time  No Opinion  Some Times  Never

Which version of Window would you choose to work with?

Vanishing  No Preference  Non-Vanishing

Why?

Any further comments?
Appendix E. Papers

E.1. Published Papers

Vanishing Windows – A Technique for Adaptive Window Management.

Tunu Miah and James L. Alty

LUTCHI Research Centre
Department of Computer Studies
Loughborough University
Loughborough
Leicestershire LE11 3TU
UK.
Email: [T.Miah, J.L.Alty]@lboro.ac.uk

Abstract. Windowing system offers many benefits to users. Users can work on multiple tasks concurrently; or work with a number of windows, each connected to different remote machines or applications. Unless these windows are managed efficiently, users can easily become overwhelmed by the number of windows currently open and begin to lose their way round the desktop. This can lead to a state where the desktop is cluttered with windows. At this stage “window thrashing ” occurs, as users begin to perform window management operations (move, resize, minimize and etc.) in order to locate relevant pieces of information contained in one of several open windows.

This paper identifies a number of problems experienced by users of windowing systems such as those provided by Microsoft Windows, X-Windows or the Apple Macintosh. It outlines a technique for reducing screen clutter when using such systems. The technique is known as Vanishing or Fading Windows. One of the features of this technique is that some of the burden of the window management operation is taken over by the system, allowing the user to focus more on application domain activities.

Keywords: Screen Clutter, Vanishing Windows, Cognitive Overload, and Adaptive Window Manager.

1 Introduction: Window Management Overhead

The development and implementation of windowing systems has had a dramatic impact on the way people think about, and use, computers. New ways of interacting with computer applications have become available. It is now possible, for example, to perform multiple tasks in parallel, and to view the results of one task while performing another. The window approach supports the way that people really work. Bannon et al. (1983) analysed the workflow of people using a traditional (non-windowing) computer system. They found that people seldom completed one task in a continuous time frame. Instead, they switched from application to application in response to events happening inside and outside the computing environment. The advantages of windowing systems also arise from how people manage their desks. Malone (1993) investigated the way people arranged papers on their desktops. He observed that people tended to position papers to reinforce the way they categorised tasks. This correspondence helped them to structure their work, and served to remind them of unfinished tasks. Further, it was observed that users frequently rearranged the materials on the desktop to match changing priorities.
However, a fundamental problem of any windowing system is that the screen can quickly become cluttered with the number of windows concurrently opened by the user. Funke et al. (1993) define clutter as a condition involving highly dense and overlapping information. There is already some evidence that clutter adversely affects user performance. Bly and Rosenberg (1986) found that, for a database management task, almost half of the users' time was spent in managing the window-based interface.

Users also spend a substantial amount of their time moving, resizing, and scrolling windows (Bury et al., 1985; Card Henderson, 1987; Sandberg-Diment, 1984; Steinbrecher, 1984). These tasks are added to the user's existing application domain tasks, and do not contribute to user productivity. Johnson-Laird (1985:p.37) put it this way "we are now (with the introduction of windows) challenging a not completely confident or competent user to control, what is in effect, several different computers."

Originally it was felt that the advantages of windowing systems easily outweighed the additional burden imposed by window management. Research on this question, however, is not completely supportive of this belief. Bury, Davies and Darnell (1985) investigated the impact of windowing system on completion times for a series of information retrieval tasks. Contrary to the author's predictions, subjects took a significantly longer amount of time to complete the task in the windowing environment than in a non-windowing environment. A detailed analysis of the results suggested that the time spent on the task itself was indeed less in the windowing condition, but the extra time required by window management operations increased the overall time spent. Experiments by Bly and Rosenberg (1986) also suggested that a large determinant in the time it takes to solve a problem using windowing systems is the time spent manipulating the windows themselves.

Davies, Bury and Darnell (1985) also found that, for tasks requiring supplemental information relative to the primary task, user performance was more error-free in a windowing environment than in a non-windowing environment. However, subjects took a significantly longer time to complete the tasks. Their study indicated (like Bly and Rosenberg) that the additional time spent resulted from additional window management operations. Their data also indicated that the reduction in errors was not simply the result of having spent more time on the task. A significant time differential was evident even when all the errors had to be corrected. Apparently, the overhead of window management added a significant time burden.

Kandogan and Shneiderman (1997) also identified similar problems of window management, which add a time overhead to task completion. Their system, called Elastic Windows, was aimed at improving spatial layout and providing functionality for rapid multiple windows operations, thereby decreasing some of the overheads of managing the windowing environment. In their system - Elastic Windows - as one Window increases in size the others reduce proportionally.

Their results showed that task environment set-up times, for medium and high complexities for the Elastic Windows, were lower than that of the independent overlapping windowing system. But a low complexity task showed no statistically significant differences. For task environment switching, all results supported the Elastic Windows Interface. The differences were statistically significant except for low-to-medium and low-to-high environment switching. In addition, task execution times for all complexities were statistically significantly shorter for Elastic Windows, although there were some exceptions for the low complexity situation.
Appendix

It is reasonable to conclude from the above studies that the advantages of window management systems are offset by the resulting additional management operations that users need to perform. The Elastic Windows approach has shown that there are opportunities for improving window management strategies to gain greater user performance. However, it is unknown whether an overall efficiency gain would result as users became more experienced, since extensive training maybe necessary (Bly and Rosenberg (1986)) before the users of a windowing system are able to recall and execute the management operations quickly and accurately.

This paper addresses the possibility of automating and assisting the user in the task of managing and efficiently accessing windows. Our approach is based on the premise that, by freeing the user's cognitive resources from the task of managing the window aspects of the interface, more of these resources are available for application domain activities. As the problems and application tasks confronting the users become more complex and information intensive, the potential of this approach for improving overall human system performance is enhanced.

2 Adaptive Window Management Systems

An adaptive or intelligent window management system changes the window configuration in response to the environment. This adaptation can be informed either from a user model or from algorithms that determine user or system behaviour.

Funke et al. (1993) describe a system called the CUBRICON Intelligent Window Manager (CIWM), a partially knowledge-based and partially algorithm-driven system that automates windowing operations. The CIWM automatically performs window management functions such as window creation, sizing, placement, removal, and organisation. These operations are accomplished by CIWM without any direct human input, although the system provides for user over-ride of CIWM decisions. The knowledge base contains information regarding tasks and the different type of presentations that are available to CIWM - such as tables, maps, text, forms, graphs, and graphic illustrations. These presentation types define the window types available in the CIWM system.

Window layout is managed automatically by the system and can switch between the different window types. Algorithms are used to determine window size, the position of a window and windows which are no longer needed. The CIWM system is very much dependent on the task being performed. A better approach would be to generate a knowledge base for generic tasks and adapt the system, based on these generic tasks.

Stille et al. (1997) have proposed an Adaptive Window Manager (AWM) based on their earlier work on Adaptive Automatic Display Layout (A2DL) (Stille et al., 1996). The system automates the layout of the windows on the display screen according to current user and task domain, and gradually learns the users' layout requirements. This is a better approach than CIWM as it also learns new layout context of users.

Kandogan and Shneiderman (1996) approached the same problem of the user having to spend too much time on managing the windowing system from a different perspective. Rather than adapting the windows automatically, their approach was to provide rapid window management operations. For example, closing one application closes all related applications in this context.
They have also conducted an evaluation of their system (Elastic Windows, Kandogan and Shneiderman, 1997). Their experiment compared Elastic Windows with Independent Overlapping Windows in terms of user performance times on task environment setup, switching and four task execution times. They found a statistically significant performance difference in support of the Elastic Windows interface for most of the tasks. For some tasks, there was a ten-fold speed up in performance.

One interesting solution to the screen real-estate problem is a 3D Window Manager (Leach et al., 1997). Their system MaW³ - uses a 3D space with windows arranged in a tunnel. The user is positioned in the middle of the mouth of the tunnel looking toward the other end. The tunnel, and windows in it, are displayed with a perspective projection. Windows are essentially 2D, i.e. their work area is 2D. However, they have 3D frames, decoration and buttons.

The 3D Window manager was designed to address the issue of “window trashing” (Henderson and Card, 1986). It addressed it in four main ways:

1. through window “hanging”. Hanging some or all of the windows allows the user to obtain a global view of the window locations. It also allows better allocation of window real estate.
2. through the scaling of window size in inverse proportion to distance down the tunnel. Windows which are not in use may be pushed back down the tunnel where they will be small but visible.
3. by reducing the mouse movement required to access windows.
4. with transparency. If windows are made transparent, obscured windows can be seen (although selection becomes difficult).

However, extensive evaluation needs to be carried out on the 3D-window manager to quantify whether these solve the window-trashing problem. The problem is not as simple as reducing mouse movement but is more concerned with identifying required windows without the need for too many searches and switches between windows.

Any adaptive window management system needs to answer the following questions:

- What goal is adaptation trying to achieve?
- What should be adapted?
- How should adaptation take place?
- Under what conditions should adaptation take place?

Our domain for adaptation is the Window Manager and our objective is to reduce user window management operations, so that the user is free to concentrate on the task at hand and not spend unproductive time managing the interface.

**3. Vanishing Windows: An Adaptive Window Manager**

One main drawback of all window-based system is that the screen quickly becomes cluttered with the number of windows being displayed. This drawback is due to the limited screen size available for displaying the information. A simple solution to the problem is to increase the
screen size or the screen resolution. Over the years this has been done. Technology has enabled larger screens and greater screen resolution to be achieved. Nevertheless, the screen real estate problem still remains, the fundamental reason being that no matter how much screen size or resolution is increased, users will always tend to occupy the available space. This problem is analogous to the problem of the disk space usage.

To overcome the problem of screen clutter, screen usage must be managed carefully and automatically. Although users can do this themselves by making sure that they close windows that are no longer of any use to them, this puts an overhead on the task they are trying to accomplish. Hence users tend to try to accomplish the task (writing a report, whilst searching the Internet for relevant references) and only perform any windows management operation when there is a need to do so.

This need to manipulate windows arises for two reasons:

- The task is very complex, hence good layout out of the screen will reduce some of these complexities. These may be cognitive complexities (i.e. mathematical calculations) or memory complexities (i.e. trying to remember a piece of information displayed on a window).

- Searching for a specific piece of information which may have been used previously but whose location has been forgotten. In such cases, users tend to close down unwanted application/windows and move or minimise windows that are still required until the specific information is retrieved.

We therefore propose what we call the Vanishing Windows technique to reduce user window management operations through the provision of limited automation of these activities. By automating this management operation, users can concentrate on achieving the task at hand and spend less time in manipulating the interface.

3.1 Concept and Issues

The basis of the Vanishing Windows approach is the gradual reduction in screen real estate requirements for an unused window as time proceeds. This reduction strategy releases real estate for the active window and reduces unwanted clutter. A window that is apparently not needed is gradually reduced in size until it reaches the size of an icon.

The reduction of inactive window size progressively increases the overall visibility of windows on the desktop (less overlap). However, the visibility of individual window content will also reduce progressively. Therefore techniques are also needed to minimise the screen real estate occupied by inactive windows whilst maximising the visual cues available within these windows (See section 4).

The Vanishing Windows approach poses at least two important questions:

- How do we decide that a window is no longer in use?
- How do we reduce the window size?

Once a window has been opened by a user, it is difficult to know when it is of no further use (or at least no further use in the near future). There are three possible solutions to this difficulty - adopt a Task-Based approach (decide on window activeness based upon task characteristics),
adopt a Task-Independent approach (define a common strategy for all tasks), or adopt a User-Modelling approach (decide upon window activeness according to past user actions and preferences).

The Task-Based approach has some difficulties. Firstly, it will necessarily be task specific, and, because of the generic nature of some applications (for example a word processor might be used for EMAIL), it may be difficult to decide what task is actually being carried out. Secondly, the way in which a task is carried out might well depend upon the type of user running the application (for example a novice or an expert).

A User Modelling approach will be appropriate if we need personalised alternative vanishing strategies. In this approach user habits on particular tasks would be noted, learned, and then used as a basis for window management decisions.

The Task-Independent approach is the most appropriate at this stage because we do not wish to introduce too many variables into the initial system. In order to present a consistent system to the user we simply reduce the size of the inactive windows in a linear fashion.

In reality, it is likely that a combination of these approaches would be successful, but we have initially adopted a simple approach. The User Modelling approach may be adopted in future developments of the system.

Our Task-Independent solution uses a limited amount of static knowledge, which is domain independent and makes two assumptions to simplify the design.

Assumption 1: A window not interacted with for a certain period is not required in the near future by the user. This period we call the "time-out" period. We know, of course that this assumption is not always valid.

Assumption 2: Users will require a continuous reminder of the reducing importance of unused windows. They will need to know when a window has exceeded its time-out period and is progressively being reducing. They will also need to know when it has effectively been removed from the system.

These assumptions are implemented by allowing the windows to slowly vanish after the initial time-out period. The vanishing effect will clearly be visible but should not intrude too strongly into the users conscious activity. A slowly reducing window will be perceived peripherally. Should the user decide that, at any time during the vanishing cycle, the window should be re-activated, a simple selection will prevent further vanishing and restore it to its original size and location. The time-out period will then be reset. A user may also hasten the vanishing process by iconising it if required. Windows are never allowed to vanish completely, their minimum size is an icon.
3.2 Vanishing Strategies

Thus our fundamental strategy is to vanish inactive windows over a period, but there are several methods that can be applied. Each method will have a different impact on the users. Possible strategies are:

- After a time out period, the inactive windows start to shrink in size, giving the impression that the window is vanishing.
- After a time out period the active window begins to grow while the inactive windows begin to shrink in size.
- After a time out period, the inactive windows become un-overlapped by moving to the sides and adopt a tiling type layout.

**STRATEGY 1:** After a time out period Inactive windows starts to shrink in size, giving the impression of the window vanishing or fading into the distance.

![Figure 1: Vanishing Windows by Means of Window Resizing](image)

The example in Figure 1 shows this strategy in use. The user is using four windows; a word processor (MS-Word), MS-Power Point, and two directory views (MS Explorer). The user is currently interacting with the word processor. Figure 1a, b and c shows the screen after a short time. Since MS-Word is the one being interacted with, all other windows are beginning to vanish. In Figure 1d, all inactive windows have completed vanished into icons.

**STRATEGY 2:** After a time out period, the Active Window begins to grow while the inactive Windows begin to shrink in size.

This is a plausible solution but the distraction effects on users needs to be assessed. How will users feel when the window they are working on actually begins to change size?
STRATEGY 3: After a time out period, Inactive Windows become un-overlapped by moving to the sides and possibly adopting a tiling type layout. Figure 2 shows this strategy being applied to a windowing system. The main window being used remains unaffected, whilst the other two windows which are not being utilized, are slowly beginning to un-overlap, until eventually they are distinct and are tiled.

![Figure 2: Hybrid System (Tile/Overlapping)](image)

This method is a cross between a tiling system and an overlapping windowing system. It is a hybrid of the two systems and may offer the benefits of each. It maximises screen usage. Further maximisation of screen usage can be achieved if desired by expanding the inactive windows to fill the vacant space similar to most tiling based system. Cohen et al. (1986) demonstrated a similar concept. When a window is enlarged by the user, all other windows are shrunk by a proportional amount. This was called the RTL/RTL system (Siemens Research and Technology Laboratories Rectangular Tiled Layout).

4. Visual Cues for Identification

Although a window may not be required for a considerable length of time, a time will often come when a user wishes to reactivate it. Additionally, a user might see a window in the process of vanishing and decide it ought to be reactivated.

To do this, the user must identify the correct window. However, if it has reduced considerably in size (or even iconised) it might not be easy for the user to determine the content or role of any particular reduced window. Even if the window is not iconised, the window title might no longer be wholly in view. Secondly, only a fraction of the contents might be visible.
The question therefore arises as to how to manage the visual aspects of the window as it reduces. Two immediate techniques suggest themselves:

- Reduce the whole window appearance in proportion to the reduction state. This will maintain the overall view of the window but rapidly reduce actual readability, as shown in Figure 3b. The border of the window would not reduce in proportion (as happens currently when reductions are carried out in windows). This means that the basic window title is visible at normal size and gradually becomes clipped. If the "shape" of the content of the window was a distinguishing feature, this reduction strategy might not increase the Information Density.

- Reduce the window size but keep the content the same size. This is like clipping (Crop); one simply sees less of the window contents, as shown in Figure 3c.

The former approach (Scaled) provides better cues as to the overall function of the window (for example an EXCEL work sheet would be visible, if not readable). The second approach (Cropped) provides readable cues (but which are progressively becoming partial and not distinctive). The scaling technique is also more useful when two windows of the same type are reducing (e.g. EXCEL) because the distinctive patterns will remain visible for a considerable part of the reduction period.

We can therefore reduce our windows either by Cropping or by Scaling. But which one of the two provides better visual cues so that a user can identify and reactive windows when they are vanishing?

5. Effects of Window Size Variation and Presentation Style on Recognition of Window Contents
An experiment was constructed to determine how window size and presentation style (i.e. Cropped and Scaled) affected a user's ability to recognise the contents of a window.

Software was written to present one of four possible documents, in various window sizes, and in one of the two possible presentation styles. The four documents were research papers on different subject areas. There was a paper on Mobile Computing, a paper on Information Technology and the Construction Industry, a report on Video Conferencing and a paper on the Application of Fuzzy Logic to Radar Receivers. The window size was allowed to vary between 0-100% of the screen size, where 100% of the screen size was 600 by 440 pixels. The screen was a normal laptop size. (25cm x 19cm). The 10 subjects who took part were computing research students. They were all familiar with technical computer documents.

Subjects were given a paper copy of the target document (Mobile Computing) and asked to familiarise themselves with it. A 15-minute time limit was placed on this process. During the experiment, subjects were allowed to refer back to this paper document if required. However, subjects were made aware of the fact they were being timed, hence referring back would impose a penalty.

The software presented two hundred sample screens (of varying sizes) chosen at random from the four documents (including the target document) and the subject was simply asked to respond by click on one of three buttons - “Yes” this is the target document, or “No” this is not the target document, or “Don’t Know”. The response time between presentation and answer was recorded. All answers were then analysed to determine if they were correct, incorrect or “don’t know”.

Half the above presentations were Cropped. The other half were Scaled. These were mixed at random.

5.1 Analysis of Results

The results were analysed to determine the accuracy of identification in each presentation styles (Cropped and Scaled).

Higher accuracy of document identification was obtained when reducing the size of the window using the cropping technique as opposed to the scaling technique (see Table 1). This suggests that it is easier to identify the window using the cropping reduction technique compared to that of the scaling technique.

<table>
<thead>
<tr>
<th></th>
<th>Correct</th>
<th>Incorrect</th>
<th>Don’t Know</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cropped</td>
<td>70.9%</td>
<td>3.3%</td>
<td>25.8%</td>
</tr>
<tr>
<td>Scaled</td>
<td>68.2%</td>
<td>3.3%</td>
<td>28.5%</td>
</tr>
</tbody>
</table>

Table 1: Percentage of Correct, Incorrect and Don’t Know answers for both Reduction Strategies.

The overall accuracy is increased from 68.2% to 70.9% in favour of the cropping strategy. However, this is not a very significant increase in performance. More detailed analysis is required to determine how accuracy is effected at different screen utilisation percentages for both styles of reduction.
Figure 4 & Figure 5 shows how the accuracy of identification varies as the screen utilisation varies for scaled and cropped presentation respectively. The screen utilisation was later grouped into bin sizes of 10 for further analysis. This is shown in Figure 6 and Figure 7.

Both reduction strategies follow a general trend, that is, they tend to have lower accuracy at the lower end of the screen size. This is what we would expect to see. In addition the differences disappear at the higher end of screen utilisation. The critical point seems to be at about 30% screen utilisation.
A plot of both reduction strategies for number of correct answers against screen utilisation percentage confirms that, for screen utilisations of <30%, the crop reduction strategy shows a higher performance than that of the scaled strategy. In addition, as we expected, above 30% screen utilisation both strategies seem to level off (see Figure 8).
A comparison was carried out between the means of number of correct identification for the cropped and scaled reduction strategies for <30% screen utilisation using Student’s t-test. A significant (t(9)=4.34, p<0.001, 1-tail) difference was found between the two groups.

6. Removal of Redundant Information
One of the fundamental arguments for Vanishing Windows system is that inactive windows waste valuable screen real estate. In addition, they clutter the screen to a certain extent. Searching for the desired window is one of the activities that a user has to perform in a multiple windowing environment. As the number of windows on the desktop increases, the search task becomes increasingly difficult and time consuming. Overlapping windows can obscure a window, hiding relevant information. Hence, users have to move these windows to reveal the information. Once moved, the visible window may not be the one the user desires and the whole process is repeated until the desired or target window containing the required information is found.

Visual clues or cues are required for identification of the desired window. If the window is reduced or obscured, then less of the workspace is available to provide valuable information for the user to make a decision. In addition, if the window is reduced to such a size where only the toolbars are displayed (with no workspace) then it becomes difficult for the user to decide whether this window contains the required information or not. Thus, valuable workspace is also wasted within an application by toolbars. Miah et al. (1997) demonstrated a technique for reducing wasted workspace in an application by limiting the number toolbars displayed and adaptively displaying only the required toolbars based on user needs. Although this adaptive toolbars approach does increase valuable workspace area for active windows, it does nothing...
for inactive windows. However, for inactive windows does the toolbar serve any purpose? The answer is almost certainly "No". In this case our Vanishing Windows system can take advantage and remove all toolbars from inactive windows. When the application is reactivated, the toolbars can be put back in the application. Removing the toolbars from the application window will increase the overall visible workspace. Hence, the Vanishing Windows system can reduce the window further without compromising loss of visual cues from the application.

![Figure 9: Toolbars Wasting Valuable Workspace – (a) Window without Toolbars and (b) same window with toolbars.](image)

As can be seen from Figure 9, simply removing the toolbars and the menu bar from the window reveals more information about the contents of the document.

Microsoft applications such as MS-Word, MS-Excel, MS-PowerPoint and others all have a fixed toolbar width. The length of the toolbar reduces as the length of an application window is reduced. The items on the toolbar are cropped as this reduction takes place. However, the menu bar behaves differently. As the window size is reduced, the items on the menu bar wrap around to the next line. This, in effect, increases the width of the menu bar. Hence, the visible workspace within an application is reduced.
The wrapping of the menu bar occurs when the width of the window is less than the total width of the items on the menu bar.

Figure 10 shows how the overall usable area varies as the window size varies. The usable area is defined as the total window area minus the area taken by the menu bar and toolbars. If a window is placed on the diagram as shown, then the point at which the right edge of the window crosses the menu bar line can be taken as a point and a rectangle drawn. The area of this rectangle is the area occupied by the menu bar.

We can also calculate the usable area for a window containing 1, 2, 3 or N toolbars simply by using the appropriate toolbar line on Figure 10.

For a Window:

\[ U_A = (W_L W_H) - (MB_A + N_{TB} TB_A) \]  

Equation 1

Where:

- \( U_A \) = Usable Area
- \( W_L \) = Window Length
- \( W_H \) = Window Height
- \( MB_A \) = Menu Bar Area
- \( MB_{Lmax} \) = Menu bar Max Length
- \( N_{TB} \) = Number of Toolbars
- \( TB_A \) = Toolbar Area
- \( TBW \) = Toolbar Width
- \( MB_{Hmax} \) = Menu bar Max height
- \( W_{Lmin} \) = Min Window Length
Appendix

\[ TB_A = TB_w W_L \text{ Equation 2} \]

Using the model described in Figure 10 the menu bar area is as follows:

\[ MB_A = \begin{cases} W_H W_L; & (W_L \leq W_{L\text{min}}) \text{ and } (W_H \leq MB_{H\text{max}}) \\ MB_{H\text{max}} W_L; & (W_L \leq W_{L\text{min}}) \text{ and } (W_H > MB_{H\text{max}}) \\ \left( \frac{50}{231} \right) W_L; & W_{L\text{min}} < W_L \leq MB_{L\text{max}} \\ TB_w W_L; & W_L > MB_{L\text{max}} \end{cases} \]

6.1 Recalibrated Results

Using the data from the experiment outlined in section 5, we can recalculate the usable screen utilisation using Equation 1. This removes the menu and toolbars and generates a set of results based only on the usable area.

![Graph](image)

*Figure 11: Comparison of Cropped Reduction Strategy With Toolbar and Without Toolbar (Calculated)*
As expected both strategies show better performance with the toolbars removed.

6.2 Further Strategy for Improving Visual Cueing

We have found that the size of a window affects identification of a target document. The identification task becomes difficult at sizes below 30% of screen utilisation. If we want to further reduce the window size without compromising identification accuracy, we need new strategies for improving visual cueing. Three useful types of information can aid in the identification of a document. These are:

- Diagrams and pictures
- Document format (font, size, spacing, layout and so on)
- Document contents in particular keywords that characterise the document

Diagrams and pictures tend to provide the best visual cues for identification. Subjects more easily remember diagrams contained in a document than words. Document format is usually difficult to recognise unless it is very distinctive. In addition, subjects identifying a document based solely on the format of a document can not be entirely sure that it is the target document. They still require additional information to make positive identification. Keywords can be used to identify a document with reasonable confidence, so long as unique keywords can be found (using the term 'keyword' to mean any word that uniquely identifies the document).

The number of unique keywords will decrease as the number of documents increases, since words that were previously unique to one document, may no longer be unique when two or more documents are compared. This is demonstrated in Figure 13—it shows three different documents (1- a paper on “Mobile Workers”, 2- a paper on “IT and Construction” and 3- a report on “Video Conferencing”). There will be several keywords for each of the papers. For example, document 1 has keywords like “PDA”, “Mobile”, “Network” and “Computing”. However, when document 1 and 3 are the possible choices then the keywords like “Mobile”,

![Figure 12: Comparison of Scaled Reduction Strategy with Toolbar and Without Toolbar (Calculated)](image-url)
“Network” and “Computing” must be excluded as these are contained in both documents. They do not distinctly identify a specific document.

During the experiment subjects were asked to think aloud whilst making their decision about whether the document presented was the target document or not. Subjects quickly began to identify keywords belonging to the target document. They used these keywords to determine whether the document was the required document or not. In addition, they also began to identify keywords that were not part of the target document.

Concentrating the keywords might be used to improve the performance of users in identifying documents. This could mean that we could further reduce the window size and yet still retain acceptable identification accuracy.

Our initial approach will be as follows: A machine-readable dictionary is to be used to automatically identify keywords based on the frequency of a particular word used in normal English language. The frequency of each word in a document will be compared with the frequency in the dictionary for this word. If this frequency is higher in the document than that in the dictionary then this word is a candidate keyword for this particular document.

7. Implication of Results for the Vanishing Windows System

The results present in this paper have implications for the design of the inferencing engine of the adaptive window system. We can now set limits for the minimum window size to which windows should reduce. We have also determined that it is better to reduce windows using the cropping presentation style as opposed to scaling. The Vanishing Windows system is well suited for most applications where the main medium is text based interspersed with a few diagrams. Multimedia applications such as video need to be considered further and their implication for the Vanishing Windows approach addressed separately.
In any window system, unnecessary embellishment (such as the menu and toolbars) should be removed, and the unused screen real estate used to provide additional cues for identification. When users reactivate a window, the menu and toolbars can be put back in the window.

Höök (1998) outlines a number of requirements that need to be addressed before intelligent user interfaces become real and usable. One of these requirements is the development of specific usability principles for intelligent interfaces, as opposed to the re-use of principles developed for direct manipulation systems. For example, Shneiderman (1998) presented eight golden rules of interface design, one of which is the need for interface consistency. However, an objective of an adaptive interface is to adjust itself to fit the user, situation or environment, and this may violate the interface consistency rule. Therefore, work needs to be done in harmonising the design principles for direct manipulation interfaces with ones developed for intelligent user interfaces.

It is obvious that the Vanishing Windows System violates this interface consistency principle. At present we believe that for the Vanishing Windows case, the violation is acceptable. However, there may be situations where violation of this principal can cause problems. These situations need to be identified and developed to guidelines for the design of intelligent user interfaces.

8. Conclusions and Future Direction

Further experimentation is planned. The first set of experiment will expand on the experiment detailed in section 4 incorporating more than one window on the screen and examining the effect this has on the search time. Mori and Hayashi (1993) conducted some experiments to determine the effect multi-window system has on users task performance. Their experiments indicated that peripheral windows did interfere with the user’s main task activity. We would like to explore the effect that multi-windows have on the search time.

The number of windows will be increased as the experiment progresses, so that the effect the number of windows on the search time can be observed. This makes the search task more difficult as the screen becomes crowded or cluttered and will also enable us to get a feel for the effect clutter has on search time and how can we measure clutter. The approach will also enable us to examine the main attributes that contribute to clutter in a windowing system.

The second set of experiments will test the hypothesis that better performance can be achieved by identifying unique keywords for a document. Pages containing several of these unique keywords should be easier to identify than those containing none or a few keywords. Thus, if partial views contain more of these keywords, then even in a crowded or cluttered screen, it should be easier to identify the desired window. As a screen becomes crowded, the individual windows can adapt themselves to present maximum visual cues, even if the windows contain partial or obscured views.

The results of these experiments will be used to design a fully integrated vanishing windows exemplar and evaluated.

8. References
Appendix


Visual Recognition of Windows: Effects of Size Variation and Presentation Styles

Tunu Miah and James L. Alty

LUTCH! Research Centre
Department of Computer Studies, Loughborough University
Loughborough, Leicestershire LE11 3TU, UK.

Email: [T.Miah, J.L.Alty]@boro.ac.uk

Abstract

One of the problems of windowing systems today is that the desktop quickly becomes cluttered with the number of windows in use. This can hinder the user during task performance. Users become overwhelmed and disorientated by the number of windows. They quickly lose track of documents and spend much of their time locating the document by manipulating the windowing environment rather than working on the task at hand.

We propose a Vanishing Windows approach that will reduce the window manipulation required by the user and aid the user in search activities. This technique gradually reduces the screen real-estate requirements for an inactive window. The reduction of inactive window size progressively increases the overall visibility of windows on the desktop (less overlap). However, the visibility of individual window content will also reduce progressively.

This paper develops techniques for maximising the visual cues available for individual windows, even at small screen sizes. These techniques are empirical tested.

Keywords: Screen Clutter, Vanishing Windows, Cognitive Overload, and Adaptive Window Manager.

1 Introduction

Window systems provide multiple views on computer applications. These views can either present different aspects of the same task, aspects of different tasks, or a mixture of the two. Only one window is currently active and frequently obscures the non-active windows. To make a window active, the user needs to:

a) Decide the new task
b) Locate the appropriate inactive window
c) Activate it

Deciding on the new task and activating it are straightforward actions, but on a crowded screen, locating the appropriate inactive window is a search task.

In choosing the required window, the search task is concerned both with recall (what am I looking for?) with recognition (is this it?). Since users usually know what they are looking for (either the form e.g. “Excel window”, “Word window” or content of the window e.g. “letter about expenses”) the task is mainly recognition.

In accomplishing this recognition task, the user needs visual cues from the different windows (either in form or content) They are therefore seeking a collection of objects in a defined target domain.

Human beings use memory when searching for previously used information [10]. Some simple generalisations about visual information processing are beginning to emerge, one of which is the distinction between two levels of processing (entire and focused) [16]. Some aspects of visual processing are accomplished simultaneously (that is, for the entire visual field at once) and automatically (that is, without attention being focused on any one part of the visual field). Other aspects of visual processing depend more on focused attention and are performed serially. Treisman [14], [15] has also reported that the visual search for a target distinguished along a single attribute (for example, colour, shape or size) is conducted in parallel. For such targets, the target ‘stands out’ or ‘pops out’ and the search time is independent of the number of irrelevant items in the set. In contrast, the search for an item defined by the conjunction of two attributes is conducted serially, and the search time increases as the set
becomes larger. Treisman [14], [15] has concluded that the visual system is incapable of conducting parallel search over two conjunct attributes simultaneously.

In a multiple window situation, a user often has to identify the content of a window when only presented with a partial or scaled view. Visual cues to the identity the contents of the window are partially hidden, and users have to make judgements about the identity of the window (or document) based on the limited cues available. Below are examples of partial and scaled views.

In multiple window situations, a common problem facing a user is the identification of the content of a window when a partial or scaled view is presented. Visual cues about the identity of the contents of the window are partially hidden. Users have to make judgements about the identity of a window (document) based on the limited cues available. Often a window will exhibit a partial view, but it could also offer a scaled view. Below are examples of what we mean by partial and scaled view.

![Figure 1: Examples of Partial and Scaled Views](image)

The advantages of windowing environments have led to their rapid and wide-ranging acceptance. However, there are costs associated with window management. The user must now assume the window management burden. Windows must be created and displayed, placed at desired locations, moved to uncover needed information, re-sized, exposed and hidden, rearranged to meet changing needs and desires, and 'move out of the away' (or left to clutter the screen) when no longer needed. When less of the window is visible, users spend a substantial amount of their time moving, resizing, and scrolling windows ([3], [4], [12], [13]). These tasks are in addition to the user's existing application domain tasks, and they do not contribute to user productivity. Of course, many of these operations are performed in a non-windowing environment as well, but they are less complex and often take less overall time.

With the advent of multiple windows, users seldom perform a single task to completion but rather engage in multiple tasks simultaneously [1]. They often halt a task, due to an interruption, attend to the interruption, and then resume the original task. Switching between task and resuming the primary task involves a number of window management operations. It also involves the user in a search task to locate the appropriate windows for the primary and secondary tasks to enable the switch to be made. This search process would be assisted if the user had better visual cues from the target window.

As the number of simultaneous windows increases, techniques are needed to minimise the use of screen real estate by inactive windows.

2 VANISHING WINDOWS

The study by Davies, Bury and Darnell [6] found that, for tasks requiring supplemental information relative to the primary task, user performance was more error-free in a windowing environment than in a non-windowing environment. However, subjects took a significantly longer time to complete the tasks. Their study indicated (like [2]) that the additional time spent resulted from window management operations. Their data also indicated that the reduction in errors was not simply the result of having spent more time on the task. A significant time differential was evident even when all the errors had to be corrected. Apparently, the overhead of window management added a significant time burden.

Kandogan and Shneiderman [9] also identified similar problems of window management, which add time to the overhead of task completion. Their system called Elastic Windows proposes to improve spatial layout and provide functionality for rapid multiple windows operations, thereby decreasing some of the overheads of managing the windowing environment. For example, in the Elastic Windows system, as one Window is increased in size the others reduce proportionally.

It is reasonable to conclude from these studies that the advantages of window management
systems are offset by the resulting additional management operations that users need to perform.

Kandogan and Shneiderman [9] have shown that there are opportunities to improve window management strategies to gain greater user performance. Funke et al. [8] describe a system called the CUBRICON Intelligent Window Manager (CIWM), a partially knowledge-based and partially algorithm-driven system that automates windowing operations. The CIWM automatically performs window management functions such as window creation, sizing, placement, removal, and organisation. These operations are accomplished by the CIWM without direct human inputs, although the system provides for user override of CIWM decisions. The knowledge base contains information regarding tasks and the different type of presentations that are available to the CIWM system - such as tables, maps, text, forms, graphs, and graphic illustrations. These presentation types define the window types available in the CIWM system.

Our approach to this problem is an adaptive window manager known as Vanishing Windows. This is a technique for minimising the screen real estate occupied by inactive windows which progressively reduces their size. Miah et al. [11] demonstrated a technique for reducing wasted workspace, in an application, by limiting the number of toolbars shown and adaptively displaying only the required toolbar based on user needs. Similarly, toolbars on inactive windows can remove themselves, as they serve no purpose when windows are inactive.

A number of reduction methods can be applied:
1. After a time out period, the inactive windows start to shrink in size, giving the impression that the window is vanishing.
2. After a time out period the active window begins to grow while the inactive windows begin to shrink in size.
3. After a time out period, the inactive windows become un-overlapped by moving to the sides and adopt a tiling type layout.

For all case listed above, once a window is re-activated, it returns to its original size and shape. Figure 2 shows the third method being applied to a windowing system. The main window being used remains unaffected, whilst the other two windows, which are not being utilized, are slowly beginning to un-overlap, until eventually they are distinct and are tiled.

This method is a combination of a tiling system and an overlapping windowing system, and may offer the benefits of each. It maximizes screen usage. Further maximization of screen usage can be achieved, if desired, by expanding the inactive windows to fill the vacant space similar to most tiling based system. Cohen et al. [5] demonstrated a similar concept. When a window is enlarged in their system, all other windows are shrunk by a proportional amount. This was called the RTL/RTL system (Siemens Research and Technology Laboratories Rectangular Tiled Layout).

Figure 2: Hybrid System (Tile/Overlapping)

The reduction of inactive window size progressively increases the overall visibility of windows on the desktop (because of less overlap). However, the visibility of individual window content will also reduce progressively. Therefore techniques are also needed to minimise the screen real estate occupied by inactive windows whilst maximising the visual cues available within individual windows.

3 SCALING VERSES CROPPING

Although a window may not be required for a considerable length of time, a user may eventually wish to reactivate it. Additionally, a user might see a window in the process of vanishing and decide it ought to be reactivated.

To do this, the user must identify the correct window. However, if the target window has reduced considerably in size (or even iconised) it might not be easy for the user to uniquely identify it in the set of reduced windows. Even if the window is not iconised, the window title might no
longer be wholly in view. Secondly, only a fraction of the contents might be visible.

The question therefore arises as to how to manage the visual aspects of the content of the window as it reduces. Two immediate techniques suggest themselves:

- Reduce the whole window appearance in proportion to the Reduction State. This will maintain the overall view of the window but rapidly reduce actual readability, as shown in Figure 3b. The layout/shape of the document is maintained.

- Reduce the window size but keep the content the same size. This is like clipping. One simply sees less of the window contents, as shown in Figure 3c.

The former approach provides better cues as to the overall function of the window (for example an EXCEL work sheet would be visible, if not readable). The second approach provides readable cues (but which are progressively becoming partial and not distinctive). The reduction technique is also more useful when two windows of the same type are reducing (e.g. EXCEL) because the distinctive patterns will remain visible for a considerable part of the reduction period.

3.1 Experiment 1: Procedure

A pilot experiment was setup to examine the effects of reducing the window size and the presentation styles (cropping and scaling) on a user’s ability to identify a target document. Software was written to select and display randomly one of four possible documents, at various window sizes and one of two presentation styles (cropping or scaled). To minimise the effect of learning, 200 samples were presented to each user in a random sequence. The software varied each of the samples in terms of the size of the window and the presentation style.

Subjects were given a paper copy of a document and asked to familiarise themselves with it. A 15-minute time limit was placed on this process. The authors acknowledge that, if the subjects wrote the documents then it would be easier for them to identify their own document. However, we are not interested in the absolute time to identify documents but the relative times between the cropped and scaled styles. During the experiment, subjects were allowed to refer back to the document if required. However, subjects were made aware of the fact they were being timed, hence referring back would impose a penalty.

The software presented two hundred sample screens and the subject was asked to respond by selecting one of three buttons — “Yes” this is the document, “No” this is not the document or “Don’t Know”. The system recorded the responses and the time to respond.

3.2 Result of Analysis

Ten researchers and students from the department volunteered for the experiment. The subjects were required to have a computing or engineering background as the document in question was from the computing field.

Initial findings showed slightly better performance was achieved for the cropped presentation style. As expected, lower correct identification of the document was observed at low screen utilisation percentage (i.e. window size as a percentage of the desktop size). As the screen utilisation percentage increased so did the number
of correct identification. In addition, an accuracy level of about 60% was reached for 30% screen utilisation. Beyond this point, the accuracy seems to be very close for both presentation strategies (see Graph 1).

A comparison was carried out between the means of number of correct identification for the cropped and scaled reduction strategies for <30% screen utilisation using Student's t-test. A significant (t(9)=4.34, p<0.001, 1-tail) difference was found between the two groups.

4 ADAPTING THE CONTENTS OF THE WINDOW

4.1 Automatic Keyword Generation

A number of factors influenced recognition performance. Subjects identified the document based on keywords, i.e. whether the document was talking about the right subject matter, and used the format of the document to re-enforce their decisions. Where subjects were unable to identify the document based on keywords, they appeared to use the format type to assist identification (i.e. font, character/line spacing, etc.). This is a rather speculative approach as they could not be entirely sure of a correct identification, unless the format was so distinct that it could clearly be identified.

During experiment 1, subjects were asked to think aloud whilst making their decision about whether the document presented was the target document or not. Subjects commented that the appearance of the texts made it easier to identify the document. For example, they tended to remember the headings of sections, bold or text in italics. This is in line with Foster and Coles [7] study on typographic cueing of printed text. They found that typographic cueing (visual distinction between different levels of text) yields higher scores than non-cued information. Subjects quickly began to identify keywords belonging to the target document. They used these keywords to determine whether the document was the required document or not. In addition, they also began to identify keywords that were not part of the target document.

It was this observation which led to the suggestion that concentrating the keywords might be used to improve the performance of users in identifying documents. This could mean that the window size could be further reduced and yet still retain acceptable identification accuracy.

We therefore used a machine-readable dictionary to automatically identify keywords based on frequency of a particular word used in normal English language. The frequency of each word in a document was compared with the frequency in the dictionary for this word. If this frequency was higher in the document than that in the dictionary then this word was defined as a keyword for this particular document.

The difference between the expected and observed frequency is calculated based on the following formula:

\[ \text{Difference} = \frac{F_o - F_e}{F_e} \]

Equation 1: Difference between Observed and Expected Frequencies for a Word

The higher the difference the higher the probability that this is a potential keyword. Taking a few sample words, we can check to see if this holds true. For words like the, is, and, then, where and so on, the difference in expected and observed frequency should be low. However, for our document words like user, graphical, mobile should show a higher difference in the expected and observed frequencies.
Appendix

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>0.07686</td>
<td>0.09997</td>
<td>0.09845</td>
<td>0.05074</td>
<td>0.09997</td>
<td>-0.27478</td>
</tr>
<tr>
<td>is</td>
<td>0.01471</td>
<td>0.01009</td>
<td>0.45736</td>
<td>0.02097</td>
<td>0.01009</td>
<td>1.07686</td>
</tr>
<tr>
<td>and</td>
<td>0.01676</td>
<td>0.02885</td>
<td>-0.4190</td>
<td>0.02097</td>
<td>0.02885</td>
<td>-0.27304</td>
</tr>
<tr>
<td>then</td>
<td>0.00224</td>
<td>0.00137</td>
<td>0.63294</td>
<td>0.00451</td>
<td>0.00137</td>
<td>2.27567</td>
</tr>
<tr>
<td>where</td>
<td>0.00081</td>
<td>0.00093</td>
<td>-0.1282</td>
<td>0.00180</td>
<td>0.00093</td>
<td>0.92349</td>
</tr>
<tr>
<td>user</td>
<td>0.01042</td>
<td>0.00004</td>
<td>2605.29</td>
<td>NID</td>
<td>NID</td>
<td>NID</td>
</tr>
<tr>
<td>graphical</td>
<td>0.00143</td>
<td>0.00001</td>
<td>1429.90</td>
<td>NID</td>
<td>NID</td>
<td>NID</td>
</tr>
<tr>
<td>mobile</td>
<td>0.00858</td>
<td>0.00004</td>
<td>194.123</td>
<td>0.00067</td>
<td>0.00004</td>
<td>14.3770</td>
</tr>
<tr>
<td>scan</td>
<td>NID</td>
<td>NID</td>
<td>NID</td>
<td>0.01759</td>
<td>0.00005</td>
<td>3517.268</td>
</tr>
<tr>
<td>fuzzy</td>
<td>NID</td>
<td>NID</td>
<td>NID</td>
<td>0.005187</td>
<td>0.00007</td>
<td>740.0271</td>
</tr>
<tr>
<td>radar</td>
<td>NID</td>
<td>NID</td>
<td>NID</td>
<td>0.012404</td>
<td>0.00023</td>
<td>538.3109</td>
</tr>
<tr>
<td>computer</td>
<td>0.001839</td>
<td>0.000013</td>
<td>140.5183</td>
<td>0.000902</td>
<td>0.00013</td>
<td>68.39384</td>
</tr>
</tbody>
</table>

Table 1: Sample Words and their associated Frequencies

NID = Not In Document

It is clear from the sample of words shown in Table 1 that the observed and expected frequencies of words such as the, and, is, then, where and so on are very close to each other. This is true for both documents, as we would expect. These words are common words that are used in the construction of a sentence and are of no value in the identification task. The remaining words in the table show a substantial difference between the expected and observed frequencies, and are potential keywords. However, two words mobile and computer appear in both documents and the difference in frequency is greater in one document than the other document. If the two words are considered as being keywords for both documents and if both documents are compared then these keywords will have to be excluded, as they will not uniquely identify an individual document.

4.2  Adapted Presentation Styles

It would be expected that the ratio of keywords to non-keywords will effect the response time for identification. Hence, increasing the keyword to non-keyword ratio within a visible window ought to reduce the response time for identification and possibly increase the number of correct identification, thereby increasing user performance for a search task.

Keyword to non-keyword ratio can be increased in a document using a number of techniques. These are discussed below.

RETAIN ONLY KEYWORDS. All non-keywords are removed from the document, leaving only keywords. This in effect has a keyword to non-keyword ratio of 100%, i.e. all keywords. These keywords can be displayed in one of two possible presentation styles.

- LIST – Each of the keyword contained within a document is extracted and listed in table. However, because of this technique all visual cues provided by the format (line spacing, font, and so on) of the document are lost. In addition, the relationships between words (the sequence of words, or positional cues) are lost as well. For example, mobile maybe followed by computing or wireless followed by network.

Figure 4: Presenting Keywords in the form of a LIST

- COMPRESSED – This technique again removes all non-keywords from the document, but instead of removing the non-keywords, these are replaced by spaces. This retains some of the positional cues. Keywords followed by
other keywords appear together. However, this technique tends to leave large spaces between keywords and as the window size decreases the visible window portion may only contain spaces. This will hinder the identification process. Thus to minimise this occurring, spaces are compressed and replaces with three dots.

Figure 5: Presenting Keywords - COMPRESSED SPACES

An alternative to the two techniques outlined above is to highlight the keywords in the original document.

- HIGHLIGHTED - All keywords are highlighted within the document. This retains all formats and positional cues provided by the document. However, by highlighting the keywords, subjects attention is drawn to the highlighted words, as they tend to stand out of the document. This relates to the two levels of visual processing - entire and focused (Treisman, 1986).

Figure 6: Presenting Keywords by HIGHLIGHTING

4.3 Experiment 2: Procedure

A similar experimental procedure to that of experiment 1 was used. Subjects were asked to read a paper for 15 minutes. They were familiarised with the four possible techniques (Compressed, Highlighted, List and Normal), Normal being the presentation technique where no adaptation is done.

Again, to minimise the effects of learning, a random sequence of sample screens were presented to the subject. Two hundred samples were presented. The software varied the screen size between 0-100 percentage utilisation of the screen and the adapted presentation styles.

4.4 Results of Analysis

Eighteen subjects volunteered for the experiment. All eighteen subjects were presented with all 200 hundred samples.

The overall result shows that the compressed presentation technique yields the greatest recognition (see Table 2). On average 82.8% of the samples were correctly identified. The remaining three-presentation techniques are very close together in terms of the correct identification percentage. However, the incorrect and don't know percentages vary between these three styles. The normal presentation shows the highest incorrect responses.

<table>
<thead>
<tr>
<th></th>
<th>Correct</th>
<th>Incorrect</th>
<th>Don't Know</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compressed</td>
<td>82.8%</td>
<td>6.2%</td>
<td>11.0%</td>
</tr>
<tr>
<td>Highlighted</td>
<td>75.7%</td>
<td>2.5%</td>
<td>21.8%</td>
</tr>
<tr>
<td>List</td>
<td>76.5%</td>
<td>5.3%</td>
<td>18.2%</td>
</tr>
<tr>
<td>Normal</td>
<td>76.2%</td>
<td>7.5%</td>
<td>16.3%</td>
</tr>
</tbody>
</table>

Table 2: Percentage of Correct, Incorrect and Don't Know answers

A repeat measure ANOVA shows that the mean number of correct identification is significantly different between the different adapted presentation styles; F(3,537)=9.69, p<0.001.

Therefore it can be concluded that increased performance in identification can be achieved by adapting the presentation styles.

4.1.1 Detail Analyse of Screen Utilisation and Identification Accuracy

Graph 2 shows how the identification accuracy varies as the screen utilisation percentage increases for the four presentation techniques. Also shown on the graph is the standard deviation marked by the vertical bar.
To determine if the differences are significant between the adapted presentation techniques for each of the screen utilisation groups (shown in Graph 2), a repeated measure ANOVA was carried out and is shown in Table 3. The most surprising result is that no significance difference was observed for screen utilisation groupings of 10-19% and 20-29%. In addition, the repeated measures ANOVA for above 50% utilisation shows highly significant differences in the sample means.

The most interesting region for our analysis is the lower end of the screen utilisation, less than 30%. As this is where we would like to increase the recognition rates, so that we can reduce the screen real estate occupied by inactive windows without compromising the recognition rate.

The result shows that above 30% screen utilisation subjects tended to score on average 80% correct identification. Between 0-9% the compressed strategy showed higher recognition and this was a statistical significance result. Similarly, for the 10-19 and 20-29 screen utilisation groups the adapted presentation (compressed, highlighted, list) showed higher recognition than the normal. However, there was no statistical evidence to support that the distribution of the sample means were different.

### Table 3: Repeated Measures ANOVA for all Screen Utilisation Groups

<table>
<thead>
<tr>
<th>Screen Utilisation %</th>
<th>ANOVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>00-09</td>
<td>F(3,51)=2.95, p=0.041</td>
</tr>
<tr>
<td>10-19</td>
<td>F(3,51)=1.25, p=0.302</td>
</tr>
<tr>
<td>20-29</td>
<td>F(3,51)=1.51, p=0.224</td>
</tr>
<tr>
<td>30-39</td>
<td>F(3,51)=2.95, p=0.001</td>
</tr>
<tr>
<td>40-49</td>
<td>F(3,51)=2.95, p=0.031</td>
</tr>
</tbody>
</table>

Some subjective comments were also collated from the subjects. They were asked to rank the four presentation styles (compressed, highlighted, list and normal) in order they found easiest to identify the target document.

An overall ranking made by the subject was determined by calculating the mode for each presentation styles (see Table 4).

### Table 4: User Ranking of Presentation Styles

<table>
<thead>
<tr>
<th>Style</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highlighted</td>
<td>1 - easiest</td>
</tr>
<tr>
<td>List</td>
<td>2</td>
</tr>
<tr>
<td>Compressed</td>
<td>3</td>
</tr>
<tr>
<td>Normal</td>
<td>4 - hardest</td>
</tr>
</tbody>
</table>

5 CONCLUSIONS AND FUTURE WORK

Windowing system have an inherent problem in that multiple windows can be opened at any one time, users can easily become overwhelmed by the number of Windows they have opened and begin to lose their way around the different Windows. This leads to a state where the desktop is cluttered with a number of Windows. Additional complications are caused by overlapping Windows, as users try to find the information they are seeking contained in one of the Window. They begin to manipulate the Windowing interface rather than perform the task.

The concept of Vanishing Windows is a favourable candidate for solving the problem of screen clutter management. It hopefully will reduce some of the burden of clutter management from the user and transfer it to the system.

Techniques have also been developed to increase the visual cues provided by windows even if the windows are reducing in size. Two experiments have been carried out to determine (1) whether scaling or cropping the window provides better visual cues and (2) Can
adapting the contents of the window increase the recognition of a window.

When windows are reduced, the result showed that on average users tended to perform better using the cropping reduction strategy than scaling.

The results of the second experiment show that by adapting the contents of the windows by compressing the keywords higher recognition accuracy can be achieved than the normal presentation. However, users commented that they preferred the highlighted presentation as the structure of the document is retained.

Future work will encompass these results into a fully functional vanishing windows system. Objective and subjective measures of the system gathered and empirical evaluation conducted.

6 REFERENCES


