

SpatialHistory: Using Spatial Memory to Recall Information

Kevin Robert Logan

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in
partial fulfillment of the requirements for the degree of

Master of Science

In

Computer Science & Applications

Christopher L. North

Douglas A. Bowman

Deborah G. Tatar

November 30 2012

Blacksburg, VA

Keywords: Spatial Memory, Large Displays, Windows

SpatialHistory: Using Spatial Memory to Recall Information

Kevin Robert Logan

Abstract

Some computer users employ large displays, 6 or more monitors, in order to view a large amount of data on a single desktop at one time. This layout can be useful when the user is performing tasks in which they must view several different information sources at a time. For example, a user may be writing a paper in which they may be simultaneously typing a document, reading another paper, and view a spreadsheet. After the task is completed, the user may close all of the windows, however sometime later they may want to view a document associated with that task. A possible scenario is for the user to know that they were viewing an important document in their top left monitor, but they cannot remember *which* document. SpatialHistory looks to allow a user to recall which windows and documents were open at a certain time spatially. The user may query a particular region of a large display and SpatialHistory will report the windows that were open in that area. Through a user study, we conclude that i) some users organize their large displays in a spatial manner placing certain types of documents and windows in certain places and that ii) our tool has the potential to help users recall previously viewed windows based on a spatial memory of their desktop.

Table of Contents

1	Introduction.....	1
2	Related Work	4
2.1	Large Displays	4
2.2	Spatial Memory	6
2.3	Task Management and Desktop Organization.....	10
3	Design and Implementation of SpatialHistory.....	13
3.1	System Overview	13
3.2	Design Considerations	17
3.3	Project Configuration.....	19
3.4	CloseCount Program.....	19
3.5	Native Code Access	20
3.6	Enumerating Windows.....	22
3.7	Window Data.....	23
3.8	Window Recall	26
4	Methodology	31
4.1	Study Setup.....	32
4.2	User Demographics.....	33
5	Results	34
5.1	Preliminary Stage Results	34
5.2	Main Stage Results	40
5.2.1	Use Cases	41
5.2.2	Detailed Results	43
6	Discussion and Future Work.....	51
6.1	Spatial vs. Global Usage	51
6.2	Behavior	55
6.3	Future Work.....	56
7	Conclusion	59
	Bibliography	60

All images taken by author unless otherwise indicated, 2012.

Table of Figures

Figure 1 - A Large Display	2
Figure 2 - Surface with Structural Grid from [Leifert 2011]. Used under Fair Use, 2012.	7
Figure 3 - Surface with Visual Grid from [Leifert 2011]. Used under Fair Use, 2012.....	7
Figure 4 - Large Display with Grids	8
Figure 5 - SpatialHistory Visual Overview	13
Figure 6 - Desktop Initial State.....	14
Figure 7 - Close PDF.....	15
Figure 8 - Recall Window	15
Figure 9 - Position Mouse	16
Figure 10 - Left Click and Recall Windows.....	16
Figure 11 - Enlarge PDF Window	17
Figure 12 - Reopen PDF Window	17
Figure 13 - The GUI for the CloseCount Program	20
Figure 14 - User32 Interface Example	21
Figure 15 - Creating an Instance of the User32 Interface	21
Figure 16 - Enumerating the Opened Windows	22
Figure 17 - Use of the Handle Tool	25
Figure 18 - Capturing Window Images.....	26
Figure 19 - SpatialHistory Tools Tab.....	27
Figure 20 - SpatialHistory Table View	28
Figure 21 - SpatialHistory Viewer Tab View.....	29

Figure 22 - SpatialHistory Viewer Tab with an Enlarged Window Image30

Figure 23 - 32 Megapixel Display38

Table of Tables

Table 1 - Preliminary Results, Windows Opened by Type35

Table 2 - Preliminary Interview Estimates35

Table 3 - Main Stage Results41

1 Introduction

Large displays have become more prevalent among users performing different types of tasks. There are some obvious advantages to using large displays such as a larger workspace and an ability to view more windows simultaneously. Initially, researchers studied the benefits of large displays in group work, but later they focused on benefits to individual users [1]. The tool developed as a part of this paper deals specifically with individual large display users. Over the course of performing tasks, users generally have several windows open spread across their display as seen in Figure 1. After a particular task is completed, the user may close any number of windows to clear up space for another task. Sometime later, the user may want to recall certain windows used during a previous task. The main question in this situation is how do users recall these windows? Specifically, do users remember which windows were open spatially or strictly by time? Do they remember that particular windows were open on a particular portion of their large display at a particular time, or do they only recall that they were viewing a window *somewhere at some time*? We look to answer these questions, among others, using our SpatialHistory tool.

Our motivation for creating this tool is based on an observed need for its intended functionality. Alex Endert, credited with the original idea for the tool, encountered such a need while working at his large display. He was working on a project in which he was referencing a particular document. The window containing the document was closed on the desktop and sometime later he needed to access it again. He remembered where on his desktop he had situated the document, but could not remember what the file was named or where he saved it. This is a scenario concerned with desktop productivity, multi-tasking, and window management.

We look to cater to these needs with our tool, but only in particular cases, such as the scenario explained above. Additionally, this tool is meant to be helpful over long periods of time; it is not expected to be immediately useful for the user. We are specifically interested in using a large display user's spatial memory of their desktop to help them recall previously opened documents. SpatialHistory is not meant to be a task management system, but rather an aid in spatially recalling previously viewed information.



Figure 1 - A Large Display

SpatialHistory runs while a user works on large display and keeps track of open windows. It then stores 'snap-shots' of the desktop in a database to be recalled at a later time based on a spatial query on the desktop initiated by the user. Users are then able to view screen shots of the individual windows that were open in a particular area of the screen. The user can also restrict a query by time and can opt to eliminate the spatial aspect altogether, getting results encompassing the entire desktop as opposed to a single region. We will discuss more details on

how this works later in the paper. Through a user study this thesis looks to see if a tool which enables users to perform such recalls will be helpful while performing tasks.

In this paper, we will discuss related work involving both large displays and spatial memory. Then we will describe the SpatialHistory tool in more detail including its implementation and functions. Then we will describe the user study with the purpose of answering the questions:

- In which cases will users recall windows that were open on a desktop spatially and in which cases will they recall windows globally (disregarding space)? Alternately, do users recall previously opened windows at all?
- Does the ability to recall previously opened windows on a desktop have any effect on how users work on a large display?

We will then discuss the results of the user study and perform an analysis on the data gathered. Finally, we will conclude the paper with insights and suggestions for future research.

2 Related Work

SpatialHistory is deeply involved in the use of large displays as well as the idea of spatial memory. SpatialHistory is not particularly practical for use as a tool in conjunction with a display any smaller than at least four monitors. The reason for this is directly related to the idea of spatial memory. It is assumed that a user would not need to recall windows spatially when using a common single monitor setup, so this situation is not considered in this paper and user study. Instead, the focus is placed specifically on large displays consisting of 6 to 8 monitors as a single desktop. In addition, the idea of spatial memory takes center attention during the user study. As a result of this focus, this related work section is divided into 3 main subsections, Large Displays, Spatial Memory, and Current Solutions.

2.1 Large Displays

Past research has been mostly concerned with collaborative work on large displays. Only within the past 8 years has research been done on the benefits of large displays for individual users [1]. Daren S. Tan and Peter Scupelli showed that large displays improve performance on spatial tasks such as 3D navigation. Further, a large display helps the user form a mental map of the information on screen. The study showed that a large display immerses the user and encourages the use of more efficient methods of completing tasks [1].

The idea of user immersion in conjunction with the use of large displays is reviewed further by Mel Slater and Martin Usoh. They define the idea of user presence as “a state of consciousness, the (psychological) sense of being in the virtual environment”. This differs from the idea of immersion which is defined as describing a relationship with physical technology itself as opposed to a “state of consciousness” associated with the concept of presence [2].

Another factor that plays into increasing immersion with the use of large displays is the concept of field of vision (FOV). A larger FOV, resulting from the use of a large display, leads to greater immersion [3]. We argue that the higher level of immersion and presence associated with the use of large displays contribute to the ability to recall information on a desktop spatially [1, 2]. In addition, large displays have been shown to benefit geospatial visual analytics tasks [4]. In a paper by Robert Ball and Chris North, the use of large displays is supported further by an extended study of the user experience of displays with three or more monitors. They showed productivity advantages using large displays in some cases [5, 6, 7]. Additionally, Mary Czerwinski has shown large displays to increase user productivity and aid user recognition memory [8]. SpatialHistory relies, in part, on the user's ability to recognize windows so recognition memory with large displays fits well with our topic.

The level of immersion when using large displays has been shown to be a preference over the use of standard displays [9]. In a 2009 paper, Xiaojun Bi and Ravin Balakrishnan found through a user study that there was unanimous preference for use of a large display over a standard one. Large displays encourage richer information tasks, the use of multiple windows for increased throughput, and increases user awareness of other windows and applications running on the desktop which are not the current focus [9]. This supports the idea that users with large displays will recall spatially which windows and documents were open on the desktop. With a higher level of immersion and a greater awareness of secondary applications, the user may be apt to form a spatial map of the current environment [1, 9].

Much research dealing with large displays have focused on simple tasks. In a 2010 paper by Christopher Andrews, Alex Endert, and Chris North, the role of large displays is expanded into the realm of sensemaking. The virtual and physical space merge to create a virtual

workspace that fosters spatial relationships [10]. We argue that the physical and virtual properties that a large display provide enable the user to form a ‘spatial memory’ of their workspace based off of these spatial relationships.

2.2 Spatial Memory

With the discussion of large displays in the previous section, the concept of spatial memory has come up in several instances. With the amount of space at a user’s disposal with a large display, users may begin to form spatial maps of their workspace [1, 9]. There has been recent research concerning the relationship of spatial memory and the user interface.

First, we begin with a discussion on the related work concerning the psychology of spatial memory. It has been shown that memory for visual context guides spatial attention [31]. Given the nature of large displays, there is a vast visual context which helps to guide this spatial attention. Spatial memory recall is often described as a top-down procedure. People generally begin at the top of their mental spatial map and work down into finer details until they recall the desired information [32]. Psychologists use a variety of tests to measure spatial memory, most of which involve grids of some sort [33, 34]. These concepts are factored into the design and user study associated with our tool. We observe the concept of grids and spatial memory next.

Svenja Leifert performed a study that aimed at understanding the influence of a grid based user interface on both spatial and content memory. The study involved testing a user’s memory of certain objects placed on a surface. The surfaces used in the experiment each contained some form of grid. As defined by Leifert, there were two aspects of the concept of a grid. The first refers to the structured alignment and layout of objects on the surface. The second refers to grids that are indicated visually by inserting lines on the surface to make up the grid.

Figures 2 and 3 contain examples of each surface [11]. It was found that both instances of the grid positively affected spatial memory while the visual grid was detrimental to content memory [11]. In addition to grids, certain hand and arm movements were shown to positively affect spatial memory in user interfaces (UI) requiring panning and zooming [12]. Kinesthetic cues also positively affect spatial memory [30].



Figure 2 - Surface with Structural Grid from [Leifert 2011]. Used under Fair Use, 2012.

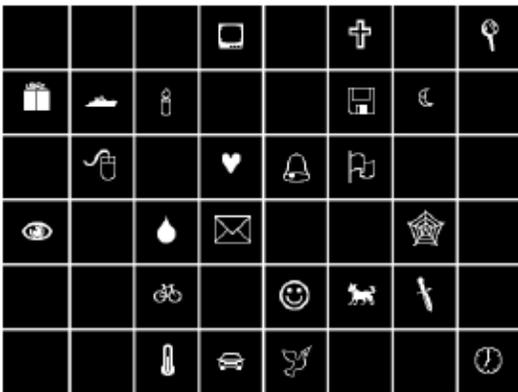


Figure 3 - Surface with Visual Grid from [Leifert 2011]. Used under Fair Use, 2012.

The ideas of spatial memory in conjunction with grids on a user interface and hand movements are related to large displays. In a large display using multiple monitors, the bezels form a visual grid for the user as seen in Figure 4. This type of large display is used in this study making the concept of visual grids directly applicable. In addition, large hand movements are

required to direct the mouse to the extreme corners of a large display. The tool that this paper describes aims to take advantage of spatial memory in order to provide useful functionality to the users of large displays.



Figure 4 - Large Display with Grids

Much research on the topic of spatial memory in conjunction with user interfaces has pertained to the jump from 2D to 3D interfaces. For example, Andy Cockburn aims to distinguish the benefits provided by a 3D interface as opposed to a 2D interface. He looks at an untested theory that moving from a 2D interface to a 3D interface will improve user performance by naturally supporting spatial memory [13, 29]. Through his study, Cockburn found that the effectiveness of spatial memory is largely unaffected by the dimension of the user interface. Essentially, it did not matter whether the user interface was 2D or 3D [13]. Results such as this suggest that future research concerning spatial memory should shy away from the dimensions of the user interface and focus on other aspects. George Robertson shows that a 3D interface for

organizing web pages, called the Data Mountain, is positively affected by spatial memory. The Data Mountain allows a user to informally organize web pages that they visit in a spatial 3D interface. It was shown that users use spatial memory to help them locate pages [35]. This result shows that the 3D interface may very well be a viable option for spatial memory research. The tool documented in this paper deals exclusively with 2D interfaces and aims to observe how spatial memory is factored into completing tasks in normal work. It also incorporates ideas from the spatial memory concept of the Data Mountain in the way we look to exploit users' spatial memories of their displays.

We intend users to use spatial memory to recall or 'refind' information while using the tool documented in this paper. As a result, we explore the research done on information refinding. In a paper, Robert Capra investigates finding and refinding information on the internet. Capra stresses that information finding and refinding are two separate tasks [28]. It was found that the ease of refinding information was affected by the familiarity of the task and the task itself. When a task was familiar, users did not spend as much time using a web based search engine [26, 25]. Additionally, the similarity of initially finding the information and refinding it at a later point also positively affected information refinding [26]. In general, prior task frequency and familiarity positively affect information refinding [27]. Social interactions have also been used to aid information refinding [24]. Blake Sawyer has shown that users may associate physical social groups with certain pieces of information. By using a large display to present information shared between social groups, information refinding can be made easier [24].

These two concepts aiding information refinding, task similarity and physical social interactions can be applied to our tool for recalling windows. For instance, two similar tasks may cause a user to recall a window or document that was involved in one and may be useful in

the other. Additionally, a project team may prompt a user to recall documents in relation to that team's project. In either situation, spatial memory may be used to recall the desired information if the user were working on a large display.

2.3 Task Management and Desktop Organization

After discussing the related work on large displays and spatial memory, we discuss some related current solutions dealing with task management and desktop organization. A key aspect of SpatialHistory is collecting a history of previously opened windows. Jun Rekimoto developed the concept of Time-Machine Computing (TMC) [14]. TMC keeps track of the desktop state for the computer at different time intervals. This allows the user to revert their desktop back to a previous state to recall previous documents [14]. Another similar system developed by Eric Freeman and David Gelernter, Lifestreams, is a 'time-ordered stream of documents' [15]. The system automatically archives files based on the time they were accessed. Further, they argue that 'storage locations are effective only as long as the user remembers them' [15]. Lifestreams is meant to give users a more intuitive means of retrieving previous files and documents [15]. These tools keep track of global history as opposed to spatial history. For example, some operating systems such as Windows have built in 'recent documents' lists which allow you to reopen previously accessed files. We refer to this as global because there is no concept of the area of the desktop; all files are together in the same group. SpatialHistory organizes previously accessed windows and files by desktop location and, hence, is spatial.

In a paper by Scott Fertig, Eric Freeman, and David Gelernter, they investigate the means by which users find and remind themselves of documents [16]. They found that there is a preference for location-based search for finding files in contrast to logical and text-based search

[16]. This supports the spatial aspect of our tool. James F. Allen stresses the importance of maintaining a history of data [17]. He explains that information systems must deal with outdated data. One approach is to delete this data, but this eliminates the possibility to access any information that is not current. He suggests logging data at temporal intervals to allow users to access this information at a later time [17]. SpatialHistory logs possibly outdated documents in temporal intervals much like Allen's suggestions. However, our tool adds the spatial aspect of accessing these documents. In a paper by Sarah Henderson, she details research concerned with understanding how users organize their desktops [18]. She installed software which keeps track of document usage and organization on participants' computers [18]. Similar to SpatialHistory, Henderson's software keeps track of file usage and organization, however, our tool allows users to recall those previously accessed files [18].

Microsoft has conducted research related to desktop and task management. Two such projects are Task Gallery and GroupBar [19, 20]. Task Gallery looks to redesign the desktop with a 3D interface to allow for quick document selection and context changes and to evoke spatial memory and cognition [19]. GroupBar is a Windows TaskBar variant which allows the user to group windows together in the same group on the task bar. This allows windows to be grouped by task for quick and easy context changes [20]. SpatialHistory incorporates some ideas from these projects. The tool is deeply involved with spatial memory and has the potential to make context changes easier for the user.

Each of the concepts and applications described in this section apply to task management or desktop organization. Some deal with the concept of spatial memory, but none directly cater to the user's spatial memory. SpatialHistory provides users direct and easy access to previously opened windows based on spatial positioning catering to specific scenarios such as Endert's case

described earlier. Our tool is meant to deal with these specific scenarios that, as of yet, have not been addressed by other related current solutions.

3 Design and Implementation of SpatialHistory

3.1 System Overview

SpatialHistory keeps track of which windows a user has open on a desktop and stores data for the user to recall at a later time through a query. There are two main functions of the SpatialHistory tool. The first function, referred to as taking a ‘snap-shot’, discovers and stores data based on each window opened at the time of execution. This function happens at certain time intervals specified by the user. The second function, called a ‘recall’, queries the system’s database and forms a report detailing the state of the desktop within user specified parameters. A recall is analogous to how a user remembers previously used windows. When a user recalls a window, we are interested in how they remembered the information. Both of these functions, along with their implementations and integration, will be covered in the following sections. Figure 5 below shows a visual representation of the two main functions in the SpatialHistory tool.

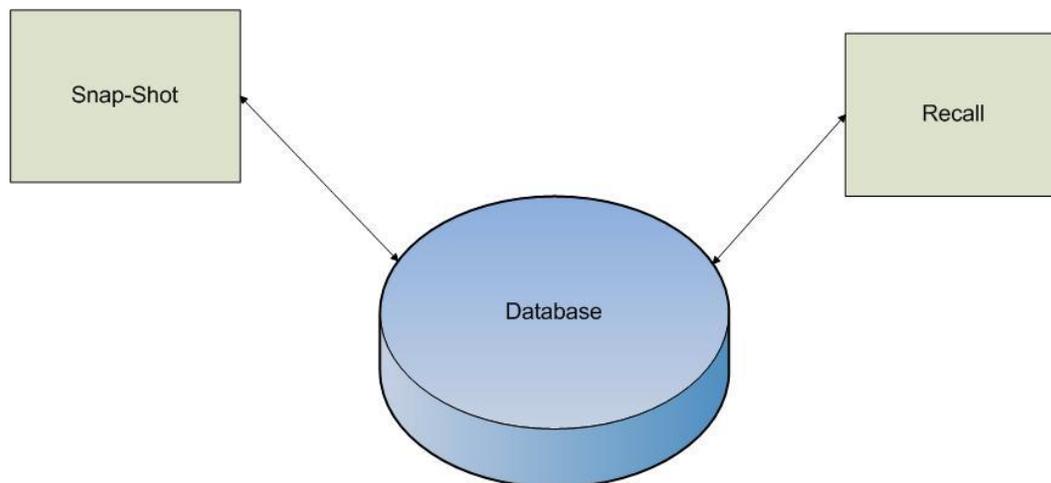


Figure 5 - SpatialHistory Visual Overview

Before going into design considerations, we demonstrate the system through a series of screen captures to give our considerations context. In Figure 6, we see a desktop with several windows open. The SpatialHistory program automatically captures the state of this desktop. The user closes the pdf file located near the left of the screen as seen in Figure 7. Later, the user realizes that they want to read the pdf that was sitting in that position again, but they cannot remember which file it was. They decide to use the SpatialHistory program to recall which windows were open in the general area. They press the Recall button as seen in Figure 8. They move the mouse to the position and left click as in Figure 9. SpatialHistory displays the Viewer tab allowing the user to leaf through thumbnails of recent windows as seen in Figure 10. Among them is the pdf. The user is able to select the thumbnail to show a larger image shown in Figure 11. Finally, they can reopen the pdf window by clicking the large image as seen in Figure 12.

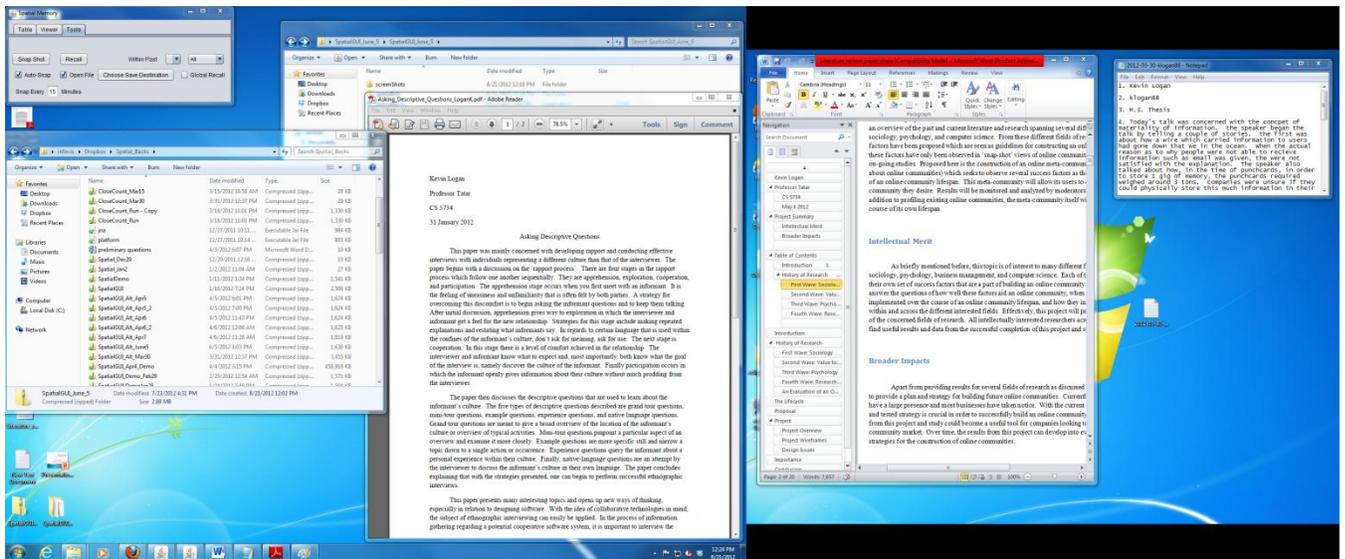


Figure 6 - Desktop Initial State

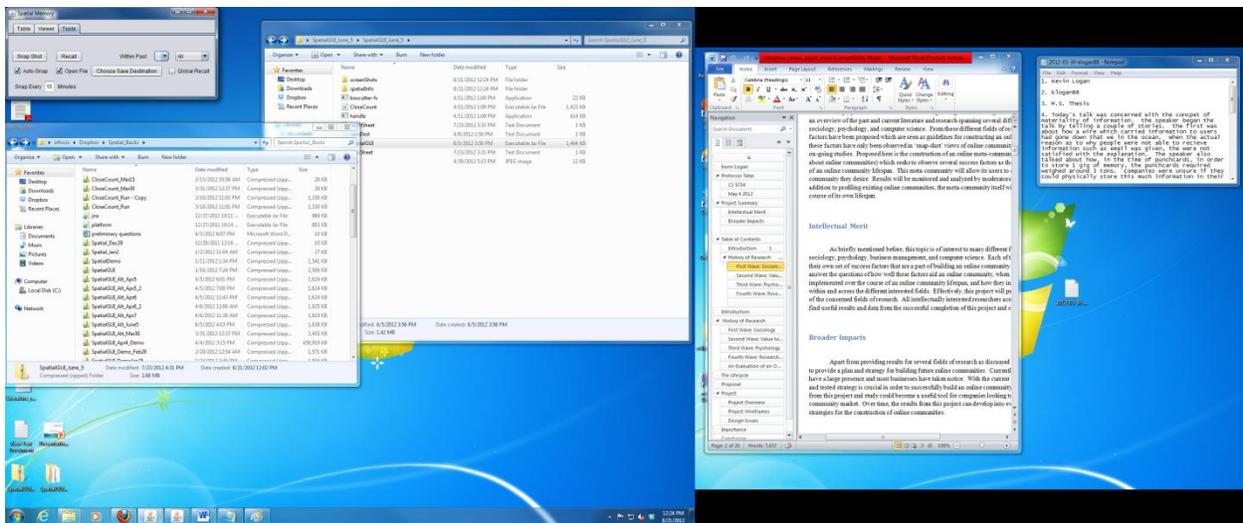


Figure 7 - Close PDF

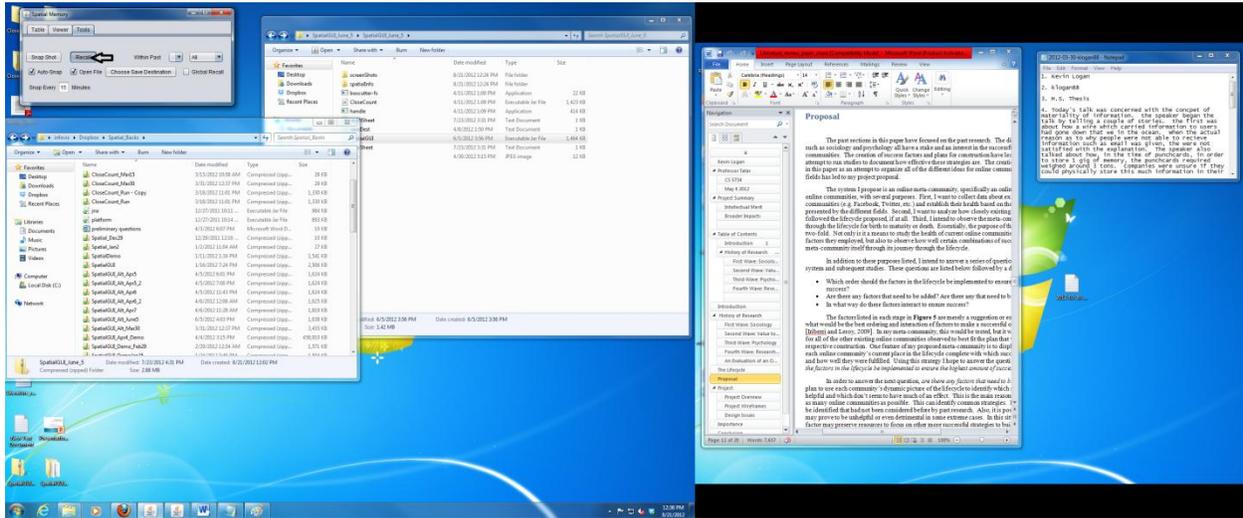


Figure 8 - Recall Window

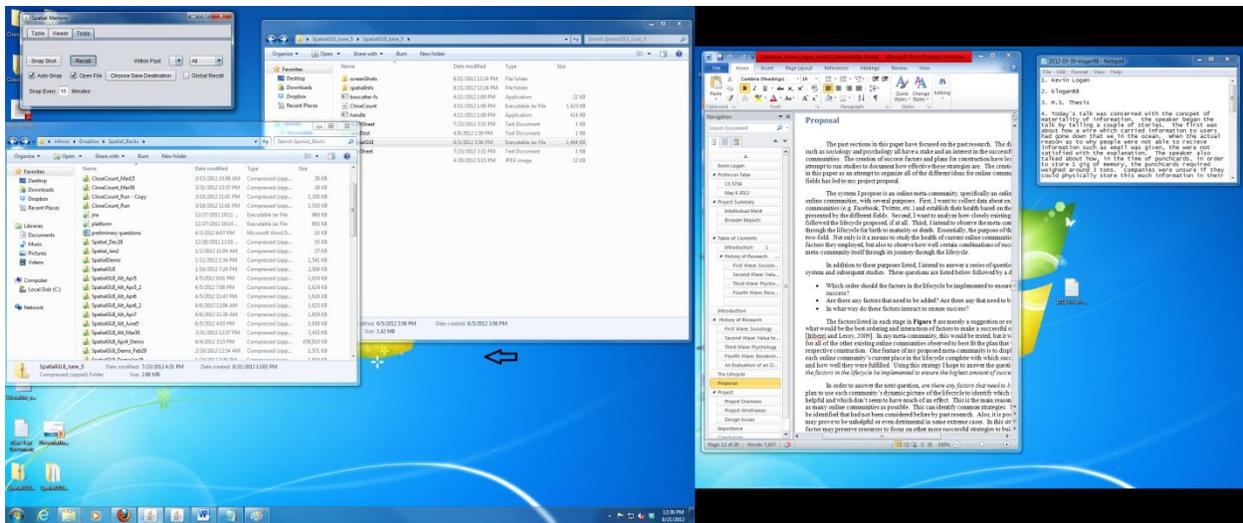


Figure 9 - Position Mouse

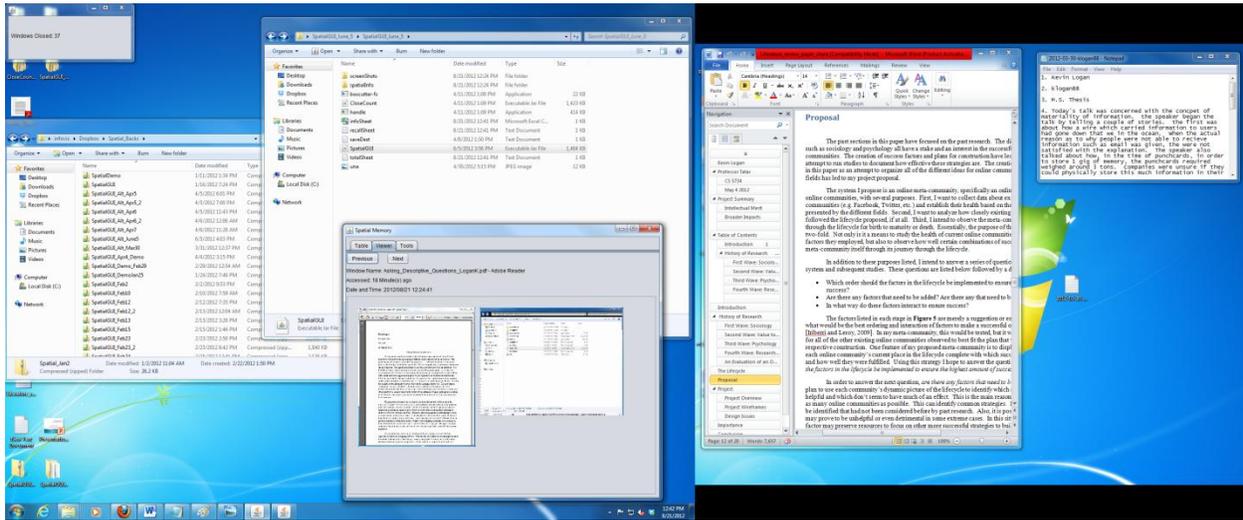


Figure 10 - Left Click and Recall Windows

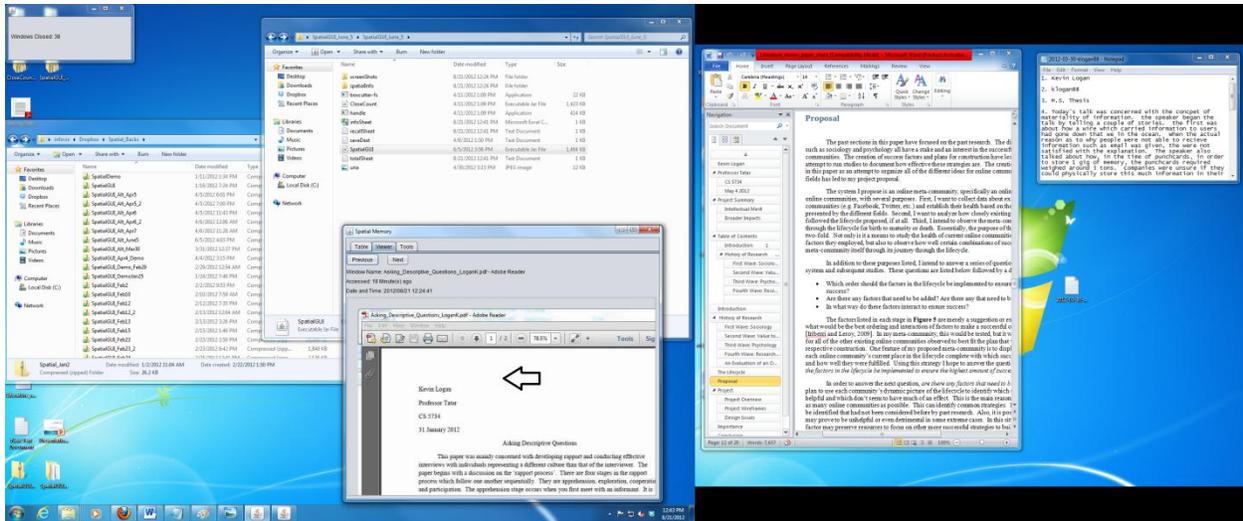


Figure 11 - Enlarge PDF Window

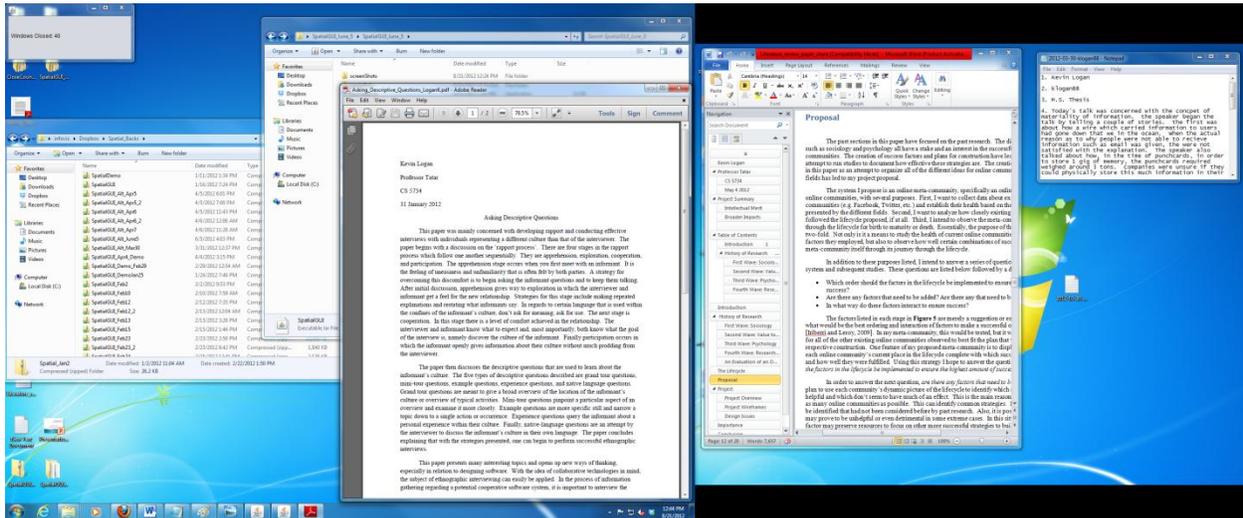


Figure 12 - Reopen PDF Window

3.2 Design Considerations

We designed SpatialHistory as a standalone Java application which is run by the user as they use their desktop for normal work activity. The program collects the appropriate data without input from the user or intrusive computation. It is designed to run alongside the other applications used during the normal completion of tasks.

The choice of language, Java, is significant. The nature of the program requires operating system specific system calls in order to retrieve information about open windows on the desktop at any point in time. Due to this fact, different versions have to be created for Microsoft Windows, Mac OSX, and Linux operating systems. Java, as the primary language, allows for as much of the code to remain the same as possible between the different versions with the use of the Java Virtual Machine. At the time of this paper, only a Windows version of the program has been implemented.

We used the Java Native Access (JNA) package to gain access to the Windows specific system calls through Java [21]. In addition to Windows specific calls, JNA also provides access to other operating system specific calls such as Mac OSX calls. The combination of Java and JNA allows us to keep all pure Java portions of the implementation consistent between future versions of SpatialHistory.

The SpatialHistory tool takes snap-shots of the current state of the desktop, logging which windows were open and where. These snap-shots occur automatically and periodically at the user's discretion. We chose to have the system take snap-shots at regular intervals to keep the computational load at a minimum. The alternative, programming the tool to listen for certain events such as a window moved, would cause a greater computational load as the user is manipulating the layout of the desktop. Logging information at regular intervals, such as every 15 minutes or half hour, reduces this potential computational strain. We designed SpatialHistory to be as lightweight and computationally quiet as possible.

We always gave much consideration to the design of the user interface (UI). The main goal of the UI was to provide the user an easy and intuitive means for viewing their previously

opened windows. The main viewing component of the UI, the Viewer tab, was designed with this goal in mind. The Viewer tab allows the user to quickly view thumbnail versions of window images. Users can then enlarge images and reopen some windows. We display window information at the top of the UI when a particular window image is in focus. In addition to the Viewer tab, we provide a Table tab which simply lists window information in table without any images. This is intended to give the user a faster means of browsing all windows if they know what the window was named. We made the decision to keep the Tools on a separate tab all together to leave more room for the Table and Viewer tabs.

3.3 Project Configuration

When users run SpatialHistory for the first time, they may choose a save destination which stores all of the window data collected by the program. The size of the data can become quite large over the course of a few months, so giving the user a choice, such as an external hard drive, is important.

In addition to choosing where the window data will be stored, users can also choose the frequency of snap-shots, or when the program collects data. As previously stated, this is to keep the computational load of the program light and uniform. The default setting is every 15 minutes, but the user can enter any number of minutes with 3 being the minimum. Most users participating in the study used every 10 to 30 minutes with success.

3.4 CloseCount Program

When the user starts the SpatialHistory tool, a second program starts called CloseCount. This program is not used directly by the user, but instead collects additional information about

system and SpatialHistory usage. Specifically, CloseCount keeps track of how many windows are closed in total across the entire duration of the study. In addition, the program logs how many windows were closed each hour, how many windows are open at each hour, and which types of windows were open such as a pdf or Microsoft Word document. CloseCount displays a simple graphical user interface (GUI) which displays the total number of windows closed as seen in Figure 13. This GUI is simply an easy way to tell that the program is running and functional, as the program stores detailed data in a .csv file. We will go into more detail on how we used CloseCount during the user study later in this paper.



Figure 13 - The GUI for the CloseCount Program

3.5 Native Code Access

In order to perform the tasks required by this project, we must first access the operating system's native system calls. As previously stated, we accomplished this through the use of Java Native Access [21]. The first step in this process is to create a Java interface which extends the standard call library of the specific operating system, in our case, Windows. Each native system call you intend to use must be included within the Java interface to make the link between the Java environment and the native platform. Figure 14 shows an example of such an interface. This code snippet is simply an example and not a fully functional implementation. The interface extends `StdCallLibrary` which contains all of the implementations for the native system

calls. We use the `UNICODE_OPTIONS` map in the `User32` instance to specify the desired platform. We then load the appropriate library using `UNICODE_OPTIONS`. Finally we provide the method headers for the system calls we wish to use in our program. In this example, we have included `EnumWindows` and `GetWindowText`. We go into more detail on what these system calls do in later sections. Obviously, more system calls could be included in the interface depending on our purposes within the program.

```
public interface User32 extends StdCallLibrary {

    static Map UNICODE_OPTIONS = new HashMap() {
        {
            put("type-mapper", W32APITypeMapper.UNICODE);
            put("function-mapper", W32APIFunctionMapper.UNICODE);
        }
    };

    User32 INSTANCE = (User32) Native.loadLibrary("user32", User32.class, UNICODE_OPTIONS);

    interface WNDENUMPROC extends StdCallCallback {
        boolean callback(HWND hWnd, Pointer arg);
    }

    boolean EnumWindows(WNDENUMPROC lpEnumFunc, Pointer arg);

    int GetWindowTextA(HWND hWnd, byte[] lpString, int nMaxCount);
}
}
```

Figure 14 - User32 Interface Example

We now have our own library within Java that we can use to access system information. Within our program, we create an instance of our `User32` interface and use the system calls at our disposal as seen in Figure 15.

```
final User32 user32 = User32.INSTANCE;

user32.GetWindowTextA(hWnd, windowText, 512);
```

Figure 15 - Creating an Instance of the User32 Interface

3.6 Enumerating Windows

After the user has set their preferences in the program, `SpatialHistory` will begin to collect window data from the desktop. We accomplish this with the use of the Windows system call library that we afforded ourselves access to through the `User32` interface. First we use the system call, `EnumWindows` which gives us access to the current collection of open windows through the use of window handles. A window handle is an object we can use to access the properties of each window. One of the parameters of `EnumWindows` requires us to provide a callback method. This contains the code that is run with each window handle in order to perform the desired tasks. Figure 16 outlines how this is accomplished through Java code. Through an instance of our `User32` interface we call `EnumWindows` and define our callback method. The `HWND` object used in the callback method is the window handle. We use the `User32` instance and window handle to get useful information about each window such as the name of the window, the process ID, and its location.

```
user32.EnumWindows(new User32.WNDENUMPROC() {  
    int count;  
  
    public boolean callback(HWND hWnd, Pointer userData) {  
        byte[] windowText = new byte[512];  
        user32.GetWindowTextA(hWnd, windowText, 512);  
        String wText = Native.toString(windowText);  
  
        IntByReference p = new IntByReference();  
        int pid = user32.GetWindowThreadProcessId(hWnd, p);  
  
        int[] rect = {0, 0, 0, 0};  
        user32.GetWindowRect(hWnd, rect);  
  
        return true;  
    }  
}, null);
```

Figure 16 - Enumerating the Opened Windows

3.7 Window Data

The specific data we collect on each window includes the name of the window, the process ID of the window, the spatial location of the window on the desktop, and the associated file of the window such as the Word or Excel document. The name of the window, the process ID, and the spatial location of the window are all easily found through Windows system calls. The code example in Figure 16 shows the use of the `GetWindowText`, `GetWindowThreadProcessID`, and `GetWindowRect` system calls to find each property respectively.

In addition to providing users with the knowledge of previously opened windows in certain area on the desktop, we would also like to enable the user to reopen those previous windows, calling for the need to keep track of a window's associated file. We define an associated file as the main file responsible for filling out a window's content, such as the pdf file displayed in an Adobe Reader window. There are a few reasons why keeping track of this associated file is not straightforward. First, some programs read from a source file, create the image for their window and then close the file making it difficult, from a programmer's point of view, to discover which file was opened. Another reason is that many programs which open windows have several different files opened in addition to the desired associated file. For example, when a user is viewing a pdf in Adobe Reader, more than likely, the file they would expect to open when recalling that particular window is the associated file. However, when Adobe Reader is running in a window on the desktop, it has several files open simultaneously which can make isolating the desired associated file somewhat difficult [22]. A final reason that reopening previous windows is difficult is the fact that many windows opened on a desktop have

no clear associated file to restore a session. For instance, consider a web browser running while using the SpatialHistory tool. The browser is displaying the contents of a particular URL and not a specific file on the local machine. Due to this, a query of the operating system's opened windows and file system will likely not produce the required information for restoring a web browser session. Alternate means are required to restore web browser windows in addition to other types. We document how we solved this problem for a select few types of windows in the following sections, specifically pdf documents and most Microsoft Office documents. Restoring sessions for most other types of windows is currently not implemented in SpatialHistory.

In order to restore previous sessions of Adobe Reader and Microsoft Office programs, we must log the file path of the associated file when collecting window data. We do this using two easily collected window properties, specifically the window name and the process ID, and an additional Windows command line tool called Handle [23]. The Handle tool provides the user with information on current processes running on a computer. We use the Windows command line from our program and call Handle's commands in order to pinpoint the desired associated file for certain windows. We pass Handle the process ID for the appropriate window that we found in the previous step. The tool returns a long list of opened files associated with the process ID as seen in Figure 17. We then use the window name to parse through the data and isolate the desired file. This method would possibly work for additional window types and programs, but would have to be manually implemented. Discussions such as this and other potential improvements are discussed in later sections.

```

C:\Users\Kevin>handle -p 1804

Handle v3.46
Copyright (C) 1997-2011 Mark Russinovich
Sysinternals - www.sysinternals.com

   C: File (RW-)   C:\Windows\winsxs\x86_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8
.0.50727.6195_none_d09154e044272b9a
   2C: File (RW-)   C:\Windows\winsxs\x86_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8
.0.50727.6195_none_d09154e044272b9a
   30: File (RW-)   C:\Windows\winsxs\x86_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8
.0.50727.6195_none_d09154e044272b9a
   6C: File (RW-)   C:\Windows\winsxs\x86_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8
.0.50727.6195_none_d09154e044272b9a
   90: File (RW-)   C:\Windows\winsxs\x86_microsoft.windows.common-controls_659
5b64144ccf1df_6.0.6002.18305_none_5cb72f2a088b0ed3
   B4: Section      \Sessions\1\BaseNamedObjects\1TIPSharedMemory
   F0: File (RW-)   C:\Windows\winsxs\x86_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8
.0.50727.6195_none_d09154e044272b9a
   110: File (RW-)  C:\Windows\winsxs\x86_microsoft.windows.common-controls_659
5b64144ccf1df_6.0.6002.18305_none_5cb72f2a088b0ed3
   114: Section      \Sessions\1\BaseNamedObjects\windows_shell_global_counters
   140: File (RW-)  C:\Windows\winsxs\x86_microsoft.windows.common-controls_659
5b64144ccf1df_6.0.6002.18305_none_5cb72f2a088b0ed3
   148: Section      \Sessions\1\BaseNamedObjects\PrimaryWord12SharedMemoryArea_
S-1-5-21-3914906895-3972884071-3863531042-10001
   17C: File (RW-)  C:\Windows\winsxs\x86_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8
.0.50727.6195_none_d09154e044272b9a
   180: File (RW-)  C:\Windows\winsxs\x86_microsoft.windows.common-controls_659
5b64144ccf1df_6.0.6002.18305_none_5cb72f2a088b0ed3
   230: Section      \BaseNamedObjects\__ComCatalogCache__
   238: Section      \BaseNamedObjects\__ComCatalogCache__
   240: File (R--)   C:\Windows\registration\R00000000000009.clb
   248: Section      \Sessions\1\BaseNamedObjects\1TIPSharedMemory
   254: File (RW-)  C:\Windows\winsxs\x86_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8
.0.50727.6195_none_d09154e044272b9a
   25C: File (R--)  C:\Users\Kevin\AppData\Roaming\Microsoft\Templates\Normal.d
otm

```

Figure 17 - Use of the Handle Tool

The final step in taking a snap-shot of the desktop is to capture and store the graphical images of each window. This information is also collected from the operating system using Windows system calls. Figure 18 below shows code to retrieve window images. First, using the window handle, we retrieve the device context of the window with `GetWindowDC`. The device context is a Windows data structure which contains all of the drawing attributes of the associated window, thus making it possible for us to capture its image. We then use the bounds of the window to determine the image's width and height. With this information, we can then get the associated bitmap of the window and save it as an image. Finally, we save the images with the rest of the window data to be used later.

```

HDC hdcWindow = user32.GetWindowDC(hWnd);
HDC hdcMemDC = gdi32.CreateCompatibleDC(hdcWindow);

RECT bounds = new RECT();

user32.GetWindowRect(hWnd, bounds);

int width = bounds.right - bounds.left;
int height = bounds.bottom - bounds.top;

HBITMAP hBitmap = gdi32.CreateCompatibleBitmap(hdcWindow, width, height);

```

Figure 18 - Capturing Window Images

We have now taken a full snap-shot of the current state of the desktop. The windows and all of the appropriate information including the name, process ID, location, and associated file, are logged. The next step comes when the user wants to recall previously opened windows. We describe the process and implementation of recalling windows in the following section.

3.8 Window Recall

The SpatialHistory tool performs a ‘recall’ when the user wants to query the system for previously opened windows in a certain area on the desktop. Refer back to the scenario presented for recalling windows in Figures 6 – 12 at the beginning of this section for context in using the recall feature of SpatialHistory. Figure 19 below shows the Tools tab of the SpatialHistory interface. This view displays the widgets the user can use to manipulate the SpatialHistory tool. In order perform a recall the user must click the Recall button, after which SpatialHistory performs a mouse hook. The mouse hook allows SpatialHistory to receive mouse events from outside its window pane. The user may click anywhere on the desktop and SpatialHistory can receive the event effectively allowing the tool to perform spatial queries from anywhere on the display. With this functionality, the user can now move the mouse to the area of the screen they want to recall and click. After the mouse is clicked, an invisible bounding

box, approximately the size of one monitor, is drawn with the mouse in the center.

SpatialHistory now knows which area to query. Any window that overlaps this invisible box is included in the results. As seen below in Figure 19, the user can also narrow the search to be within a certain time frame. Figure 19 shows that the user wants to recall windows opened within the past 3 days.

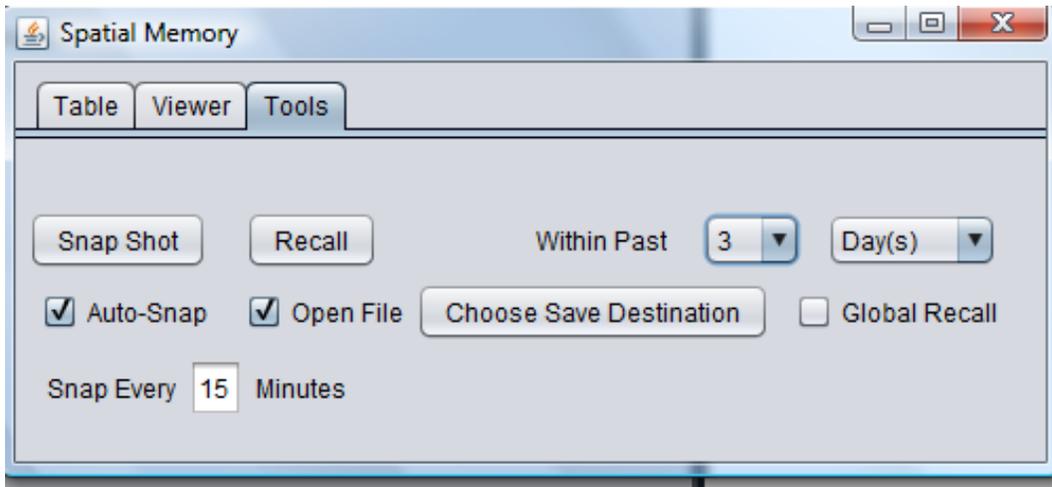


Figure 19 - SpatialHistory Tools Tab

After the user clicks the area they wish to query, SpatialHistory searches through its database and returns the appropriate window data based on its location and date opened. The tool represents the window data results in two different ways, in table form and window image form shown by clicking the Table or Viewer tabs respectively as seen in Figure 19. The Table tab shows the data in a list with each entry on separate row. There are three columns which display the window name, file location if available, and time accessed. If the file location is available, the user can open it by clicking on the appropriate row. An example of the Table tab is given in Figure 20 below. The Table tab is meant to give the user the option of viewing the results in more quickly than leafing through images. However, images of the windows may be

more useful to the user in the event that they do not know exactly which window they had opened. This is the reason for the Viewer tab discussed next.

The screenshot shows a window titled "Spatial Memory" with three tabs: "Table", "Viewer", and "Tools". The "Table" tab is active, displaying a table with three columns: "Window Name", "File Accessed", and "Date Accessed". The table lists various windows and files accessed, with dates ranging from "1 Minute(s) ago" to "1 Month(s) ago".

Window Name	File Accessed	Date Accessed
Start	FILE NOT FOUND	1 Minute(s) ago
Spatial Memory	FILE NOT FOUND	1 Minute(s) ago
SpatialGUI_RUN	C:\Users\Kevin\Desktop\...	1 Minute(s) ago
Mail :: Welcome to VT W...	FILE NOT FOUND	1 Minute(s) ago
Thesis	C:\Users\Kevin\Dropbox\...	1 Minute(s) ago
Artificial_Intelligence__A...	C:\Users\Kevin\Desktop\...	1 Minute(s) ago
thesis_realOne - Micros...	C:\Users\Kevin\Dropbox\...	1 Minute(s) ago
CS 2114 Sum I 12: Assi...	FILE NOT FOUND	3 Day(s) ago
Untitled - Paint	FILE NOT FOUND	3 Day(s) ago
Windows Task Manager	FILE NOT FOUND	1 Month(s) ago
Gmail: Email from Goog...	FILE NOT FOUND	1 Month(s) ago
Psychology	FILE NOT FOUND	1 Month(s) ago
Skype	FILE NOT FOUND	1 Month(s) ago
Save As	FILE NOT FOUND	1 Month(s) ago
Aid Human Memory - A...	FILE NOT FOUND	1 Month(s) ago
Somesthetic Define So...	FILE NOT FOUND	1 Month(s) ago
Location Synonyms, Loc...	FILE NOT FOUND	1 Month(s) ago

Figure 20 - SpatialHistory Table View

The Viewer tab shows the window image version of the resulting recall data. The images are initially shown as thumbnails as see in Figure 21. When the mouse hovers above a window, information about that window is displayed in the top left corner of the interface. A thumbnail is enlarged to full size when a user right clicks the image as see in Figure 22. A second right click on an enlarged window image reopens the associated window if it is available while a left click returns the interface to original thumbnail view. When the user performs a recall, the tool automatically reports the results in the Viewer tab. We intend this to be the default tab for reviewing results as we expect users to recall windows by image over window name.

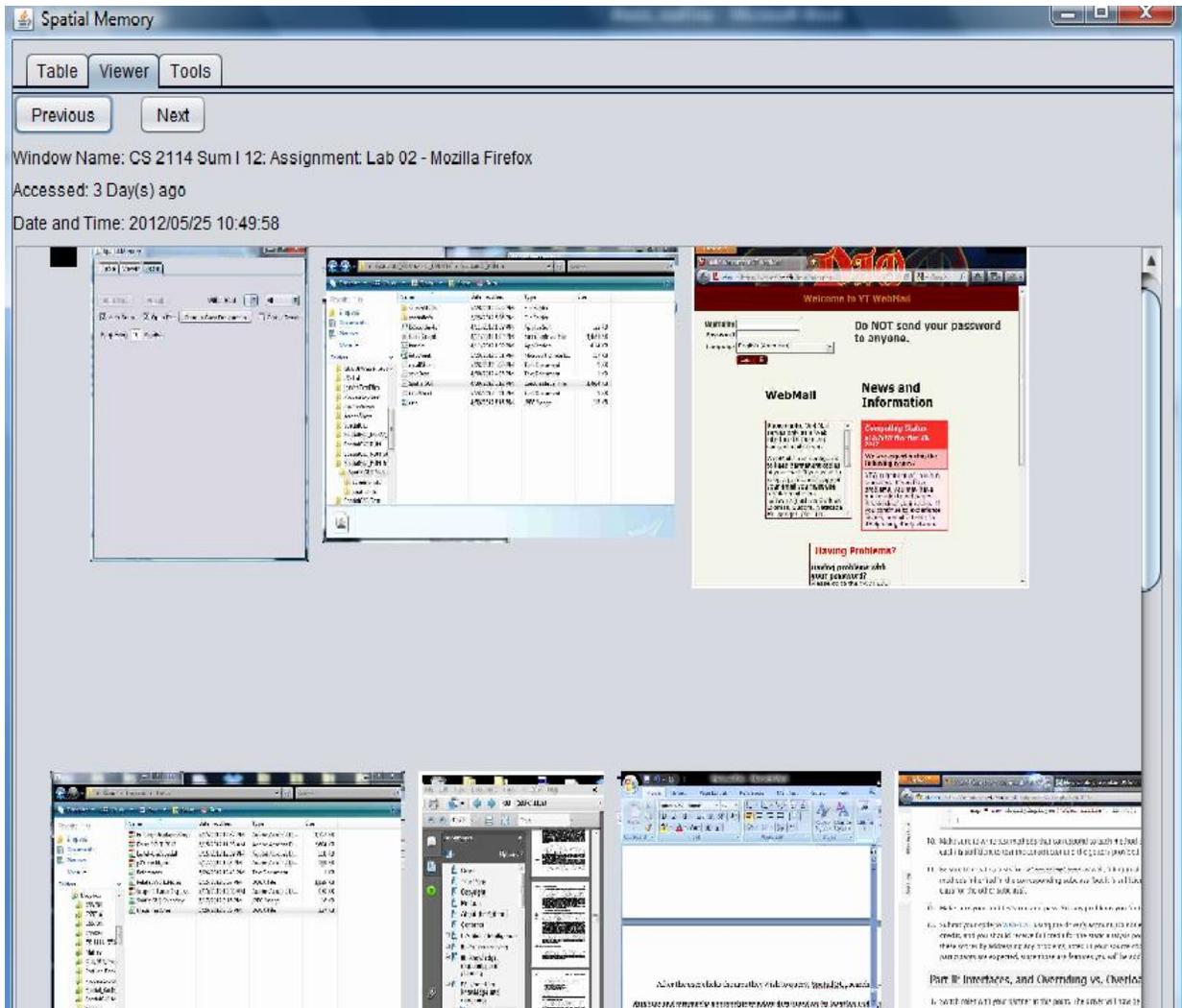


Figure 21 - SpatialHistory Viewer Tab View

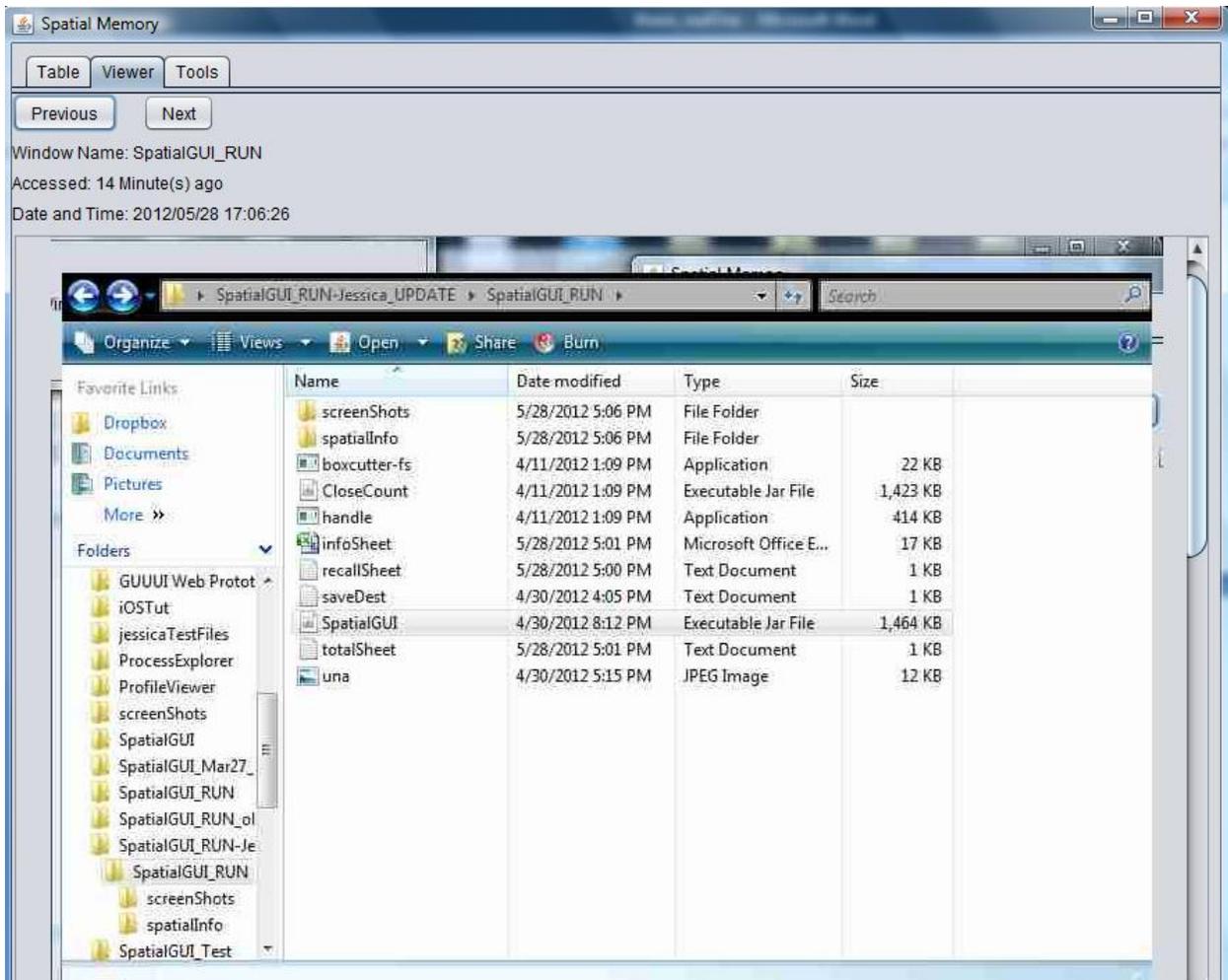


Figure 22 - SpatialHistory Viewer Tab with an Enlarged Window Image

In this section, we have described the SpatialHistory tool in terms of both functionality and implementation. We described how we intend the tool to be used by users and our study participants. In the next section, we describe the methodology for our study, how we set it up, and how we log data. Most importantly, we describe our research questions and hypothesis in more detail.

4 Methodology

The purpose of this study is to evaluate SpatialHistory in terms of usefulness as a tool in addition to answering questions related to large display organization and how users recall window data. As a result, we are directly interested in the following questions:

Q1: In which cases will users recall windows that were open on a desktop spatially and in which cases will they recall windows globally (disregarding space)? Alternately, do users recall previously opened windows at all?

H1: We hypothesize that users will recall windows spatially in cases in which the user forms a spatial layout of their desktop. Specifically, they will recall that certain windows were open in a particular region of the display. We refer to a user's ability to describe the average layout of their desktop as a spatial layout. Users who do not organize their desktops spatially will recall windows globally. In instances in which users form a spatial layout and recall windows spatially, SpatialHistory will prove to be a helpful tool.

Q2: Does the ability to recall previously opened windows on a desktop have any effect on how users work on a large display? Specifically, will participants utilize their workspace any differently knowing they can recall windows?

H2: We hypothesize that our tool has the ability to change the way users work. We predict that users may close windows more often to free up space for different windows since they will be able to recall those windows at a later time. However, we realize that this may not be an immediate result as users may need to become familiar with the tool before long term effects such as this come into play.

4.1 Study Setup

Our study consisted of four participants and Kevin Logan, an author of this paper, each with their own large display desktop with eight monitors each. We intend SpatialHistory to be a tool for use at a workstation over an extended period of time. Due to this, our study required participants to work regularly at a large display as opposed to creating a controlled, stand-alone study.

We designed two stages for our study, a preliminary stage without use of the SpatialHistory tool, and a main stage with full use of the tool. The preliminary stage lasted 3 weeks and the main stage lasted for 2 – 3 months depending on participant availability. Before beginning the preliminary stage, we explained the SpatialHistory tool to the participants and its intended use. Participant understanding before they started using the tool is necessary to answer our research questions. During the preliminary stage, our participants worked normally at their workstations without access to the SpatialHistory tool. Instead, the users ran the CloseCount program which logs data about window usage including when and how many windows are closed or opened and which types of windows are used. At the end of the first stage, we interviewed each participant about their normal work routine and habits. We asked questions specifically about how they organize their large displays and windows.

The second and main stage introduced full use of the SpatialHistory tool as well as the CloseCount program to continue gathering window usage data. After the preliminary stage, we installed the SpatialHistory tool on the participant's machines. The time period for this stage of the study was based on the usage of the tool. The nature of the program requires time to pass in order for the tool to become useful. Due to this, we allowed extended periods of time for

participants to use the tool. Through the course of this time period, participants worked normally at their stations. We continually inquired with participants for any instances of use throughout the study. After the participants had chances to use the tool, we asked them a series of questions to learn about their experience. At the end of the study, we analyzed the participants' responses as well as the data gathered from the CloseCount program.

4.2 User Demographics

We had four participants who each worked regularly on a computer with a large display. All four participants were graduate students studying for a Doctorate. Two of the participants were male, between the ages of 25 and 30, while the other two were female in the same age range. Each had ample previous experience working on their large display and completed much of their daily work on the machines. This study required use of a large display over extended periods of time, so these qualities were required in order to participate.

Due to the necessity of users with large displays, our results are qualitative over quantitative. Also, many users are members of our own lab group since few use such large displays. Additionally, since this study looks to observe normal working habits, it is far from controlled. Our users are simply given a tool and we observe how it is used over the course of time. A common controlled experiment is not appropriate in this case and would not observe natural spatial mapping that occurs while performing tasks over an extended period of time.

5 Results

In this section we present the results of the study. We look at each subject individually in terms of their results in the preliminary stage and main stage. We organize the raw data collected into tables and highlight key comparisons. We also present the subjects' responses to our closing interviews in each stage. This section is divided into two main sections, one focusing on preliminary stage results and the second on main stage results. Each of these sections is a detailed description of the data collected and the subjects' responses to the interview questions of the respective stage. Section 6 discusses the importance of these results and draws conclusions.

5.1 Preliminary Stage Results

The goal of the preliminary stage was to understand how the participants work normally using their large displays. We installed the CloseCount program on their machines and instructed them to work normally as our tool collected data in the background. Nothing was required of the participants during this stage outside of working at their large displays. We intended to collect data on window usage during this stage, summarized below in Table 1. Each column represents a different participant and each row displays a different statistic. CloseCount collects data every hour while it is running. Each statistic is an average of windows that were open simultaneously per hour over the course of the stage. For instance, 'All Windows' shows how many windows on average were opened on the desktop in all. The other statistics are by type such as 'Microsoft Word'. The 'Total Closed' row indicates how many windows each user closed in the preliminary stage. 'Total Hours' indicates how many hours each subject worked. This value comes from the amount of time the user's computer was running so they may be

skewed if the user did not shut down or put their computer to sleep for extended periods of time. The final row ‘Closed Per Hour’ indicates how many windows were closed per hour on average.

	Subject 1	Subject 2	Subject 3	Subject 4	Logan
All Windows	13.77	10.07	8.16	9.85	4.10
Microsoft Word	1.76	3.06	0.97	0.24	0.20
Microsoft Excel	0.04	0.06	0.00	0.00	0.01
Microsoft PowerPoint	0.59	0.02	0.21	0.00	0.00
Other Office	0.00	0.00	0.00	0.00	0.00
PDF	0.94	2.15	0.28	0.00	0.98
Web Browser	4.93	0.99	1.52	1.60	0.36
Other Windows	5.51	3.84	5.18	8.01	2.56
Total Closed	697	111	472	79	347
Total Hours	193	363	317	388	167
Closed Per Hour	3.61	0.31	1.49	0.20	2.08

Table 1 - Preliminary Results, Windows Opened by Type

After the end of the 3 week period for the preliminary stage, we asked each of our participants a few questions about how they complete tasks and organize their workspace. In addition to describing our subjects’ responses, we also summarize them in Table 2 below.

	Subject 1	Subject 2	Subject 3	Subject 4	Logan
All Windows	Current Task Dependent	Current Task Dependent	Current Task Dependent	7	6 – 7
Microsoft Word	1	1	N/A	0	1
IDE	0	0	N/A	1	1
References	Several	Several	N/A	Several	Several
Web Browser	2	1	N/A	1	1

Table 2 - Preliminary Interview Estimates

Table 2 lays out desktop usage based on the subjects’ responses to the preliminary interview questions. Only two participants gave a direct estimation of the amount of windows they have open on their desktop at any given time. Of those two who gave an estimation, they

were incorrect by 2 to 3 windows compared to the actual data in Table 1. Each of the other subjects said that the number of windows opened at any time depended on the task they were performing. Subject 3 did not organize his desktop in a particular way so he gave no further details on the types of windows he has open. The other participants unanimously mentioned at least one web browser and several reference documents such as pdfs or spreadsheets. Most subjects also mentioned Microsoft Word. Two subjects mentioned IDEs for writing code. These responses fit well with the data collected from CloseCount in the preliminary stage. Next we give each of subjects' detailed responses to our interview questions.

We asked the following questions in this preliminary stage:

- Could you give a 'ball park' answer to how many windows you have open on average while working?
- Under which circumstances will you close a window?
- Do you have a certain way that you organize your desktop?

Subject 1 responded that the amount of windows she had opened depended on the task she was working on. She explained that the windows she had opened were usually academic papers she used while working. Generally, she would open papers related to her current task and leave them open until she had completed that task. She said that her computer would often become bogged down by the high volume of windows opened on the desktop while working. Of the study participants, Subject 1 kept the most windows opened at a time during the preliminary stage with about 13 to 14 windows. Subject 1 described that she had a certain way that she organized her desktop. She situated her email and web browser in the top left area of her display. She placed Microsoft Word documents in the center area while academic papers and

references she situated to the right of the display. Subject 1 closed a total of 697 windows during the preliminary stage. This large amount of windows closed was abnormal since, according to Subject 1, she was installing many applications on her machine since she had moved to a new workstation as the study started. Upon further analysis and ignoring data points which suggest a large amount of windows closed in a short amount of time, a more realistic number was extracted at approximately 382 windows closed. With these data points removed, her averages on windows opened remained essentially the same across the board. Subject 1 worked approximately 193 hours.

Subject 1 had a unique experience in that she switched displays at the beginning of the study. She was used to using a 13.8 megapixel display as seen Figure 4 in her normal work, the usage of which is described in the previous paragraph. During her internship, she worked on a larger 32 megapixel display similar to the one seen below in Figure 23. She explained that she used the two types of displays differently. She still had designated positions for her email and time card, however the rest of the space changed depending on what she was working on. The larger space caused her to be less consistent in placing windows. As a result, she said she would recall that she was working on a certain paper or read a certain document in a particular space on the screen, but could not remember the specific documents. She explained that she had never had this problem working on her old display. Coincidentally, by the time the study started she was moved to a 32 megapixel display. Subject 1 participated in both stages of the study using the larger display.



Figure 23 - 32 Megapixel Display

Subject 2 described that she managed her windows and desktop resources based on the current task at hand. In addition, she explained that in general she only focused on one task at a time and rarely multitasked. Subject 2 said that she closed windows containing documents when she was done referencing them. Over the course of the preliminary stage, she closed a total of 111 windows and worked approximately 363 hours. Subject 2 organized her desktop a certain way with her email in the far left side of the display and her active window in the middle. She kept her Windows Explorer windows in the right hand side of the display and references and documents in the bottom right. The results in Table 1 show that Subject 2 kept, on average 10 windows open at any given time. Her focus on writing papers and referencing documents is reflected on the share of Microsoft Word and pdf windows opened.

Subject 3 explained that he tried to minimize the amount of windows opened on the desktop necessary for the current task. He said that a large number of open windows makes the

display seem messy and can hinder work performance. Over the course of the preliminary stage, Subject 3 closed 472 windows and worked approximately 317 hours. When we asked Subject 3 about desktop organization, he responded that he didn't have a particular preferred layout. However, he explained that he enlarges a single window to fill an entire monitor while he works, avoiding layering windows on top of each other. This described behavior is reflected in Table 2. Compared to the other participants, Subject 3 keeps approximately 2 fewer windows open at any time. Additionally, the 472 closed windows over the course of the preliminary stage is a greater number than the others, reflecting his tendency to keep a relatively clean desktop.

Subject 4 described that his desktop is generally overcrowded. He explained that he keeps around 7 windows open at a time using keyboard shortcuts to switch between them. He rarely minimizes windows, layering them on top of one another. He simply brings the windows he wants to use to the front of the pile. The results in Table 1 show that he keeps 9 to 10 windows opened at a time. This slight underestimation could simply be credited to windows which the user wouldn't normally consider. For instance the 'downloads' window that opens when you download a file in Firefox. Subject 4 said that he will close a window if he feels he has completed the task to which it pertains. Additionally, he will close a resource intensive application, such as an Integrated Development Environment (IDE), if he does not think he will need it for a while. He explained he will leave certain windows open when leaving his workstation for an extended period of time to remind himself what he should be working on when he returns. He closed 79 windows over the course of the preliminary stage and worked approximately 388 hours. Subject 4 organizes his desktop a certain way while working. He explained that he moves his current task to the monitor directly in front of him. While

developing, Subject 4 keeps an editor in the center window along with references underneath it or in monitors in his visual periphery.

In addition to our subjects, Kevin Logan, a co-author of this paper, also used the tool on a large display and kept track of his experience. This is possibly a biased account, so it is documented separately from the other subjects. Logan described the way he organizes his large display. He estimated that he keeps 6 to 7 windows open at time while he works. He generally keeps papers he is writing in the top left area of the display. He keeps email in the bottom middle of the display and a web browser right next to it. Logan keeps reference documents such as pdfs in the middle-right to right side of his display. The pdfs that he references are the most difficult thing to recall even in the span of a few days. Logan closed a total of 347 windows during the preliminary stage and worked approximately 167 hours. The preliminary data shows that Logan kept about 4-5 windows open on his desktop at a time, significantly fewer than the other subjects. Logan was new to using a large display, used to only a laptop, so closing many windows and keeping few open may be a habit.

5.2 Main Stage Results

The goal of the main stage was to observe our participants' behavior while using the SpatialHistory tool. We installed the SpatialHistory program on their computers and explained how to use the tool. In addition to SpatialHistory, CloseCount also ran on their machines as they worked. Participants then continued to work normally knowing they had the ability to recall windows on their desktop spatially. At the end of the stage we analyzed the raw data from CloseCount and interviewed each of our participants. We organize these results into two subsections. The first section highlights the 4 use cases for SpatialHistory we extracted. The

second section presents the details of the results. The numeric data extracted is displayed below in Table 3. The layout and semantics of the data in Table 3 is identical to Table 2 except that the data values pertain to the main stages.

	Subject 1	Subject 2	Subject 3	Subject 4	Logan
All Windows	8.81	6.28	6.63	14.22	7.37
Microsoft Word	1.02	0.85	1.12	0.00	0.03
Microsoft Excel	0.41	0.36	0.00	0.00	0.00
Microsoft PowerPoint	0.63	0.06	0.00	0.00	0.00
Other Office	0.00	0.00	0.00	0.00	0.00
PDF	0.31	0.49	0.31	0.00	0.77
Web Browser	2.62	1.34	0.41	1.03	1.64
Other Windows	3.94	3.19	4.78	13.18	4.93
Total Closed	133	596	82	974	343
Total Hours	10	53	87	1319	454
Closer Per Hour	13.3	11.25	0.94	0.74	0.76

Table 3 - Main Stage Results

5.2.1 Use Cases

Before going into our participants' detailed accounts of usage, we highlight 4 specific use cases. Each of these was a possible use of the tool and occurred while users were working normally at their large displays.

1. Subject 1 discovered a possible use of the tool before she began the study. While using a large display, her system restarted and cleared her desktop. The next day she needed to recall a particular resource in a particular part of her display, but could not remember the specific file. Although this was before the study began, she reported that a tool such as SpatialHistory would have been useful.

2. Subject 1 used SpatialHistory when she had to recall several windows that she had opened while working on a particular project. She reported that she used the tool several times in this instance to discover which windows were in each monitor. SpatialHistory organized the results for her and made it easy to view the windows she was looking for. The types of windows included web browsers, Microsoft Word documents, and pdf documents. Subject 1 reported that she had an idea about which types of windows she would find in particular areas, but not specifically. She remembered tasks that she had to complete based on the results from SpatialHistory. As thumbnails populated the Viewer tab, she recalled information and tasks she still needed to complete.
3. Subject 2 used SpatialHistory after her computer crashed and needed to reboot. She queried each of her monitors to recall the windows in those areas. She was able to eliminate her two leftmost monitors since she knew she didn't keep any important information in that region. She mostly recalled text editor windows she was using to write code. She leafed through thumbnails in the Viewer tab to remember the windows she had previously opened.
4. Kevin Logan used SpatialHistory several times to reopen reference documents. He remembered where the documents were on his desktop so he was able to query those particular areas to find the files. The majority of windows he recalled were pdf documents. He queried a particular area of the desktop, leafed through the Viewer tab until he found the correct document, and reopened the window to continue working.

5.2.2 Detailed Results

The detailed responses from our participants are below. We interviewed each of our participants in order to document their experience with SpatialHistory. The specific questions we asked each participant are listed below.

- About how often did you use the Spatial Memory program?
- Can you recall, on average, how often you found the information you were looking for when you used the program?
- Did you find it more helpful searching spatially or globally?
- Did you find yourself closing more windows on your desktop than before knowing that you would be able to recall them later?
- Do you have any comments on the program in relation to design, usability, or any other relevant topic?

Before discussing Subject 1's responses to our questions we explore a significant special case she experienced. As described earlier, Subject 1 made a transition from a smaller display similar to the one shown in Figure 4 to a larger one shown in Figure 23. She explained that on the smaller display she generally had no problem keeping track of the windows she had open. On the larger display she often found herself losing track of the documents and information she was using. After she made this transition she had the opportunity to use SpatialHistory described in the first use case in section 5.2.1.

Subject 1 said that she used the SpatialHistory tool once over the course of the main stage described in the second use case in 5.2.1. The results show that she used the tool 11 times over the course of a month, each in the same instance (i.e. she used the tool multiple times to find

recall information in the same sitting). The need for these recalls resulted from switching from a project she was working on with the Mac OS back to a previous project on the Windows OS. Subject 1 explained that each of these recalls was performed in close proximity of each other attempting to recall previously used windows after a context change. She said that she vaguely remembered the information she contained spatially in each monitor, but could not remember specifics. After viewing the windows returned from SpatialHistory, she was reminded of tasks that she had to complete such as updating her resume and sending certain emails. Subject 1 explained that the thumbnails in the viewer tab were enough to recognize the windows and that she did not have to enlarge them. She mainly used the visual cues provided by the thumbnails to recall the information. An analysis of the raw data detailing the location of her windows showed that she kept certain types of windows, such as Microsoft Word documents and pdfs, in relatively similar areas. She commented that the tool presented her with a large amount of useful information in a short amount of time.

Subject 1 used the spatial recalls almost exclusively as the global recalls returned too much information for her to process at once. She explained that she preferred to search each one of her monitors individually and spatially to narrow the resulting windows. This was a way to not only reduce the amount of information she had to process, but also to make the information more relatable as she had a general idea of what she would find in each area.

When asked about a possible change in desktop organization and behavior, Subject 1 responded that she doesn't predict her behavior would change too drastically, but that the functionality of SpatialHistory would be a useful addition. She added that if she continually recalled certain types of windows in certain area with the tool that she may begin to more strongly associate those types with those areas than she did before. In the preliminary stage

Subject 1 kept 11 -12 windows open at a time compared with the main stage with 8 – 9 windows. She closed approximately 382 windows in the preliminary stage and 133 in the main stage. Subject 1 worked for only 10 hours in the main stage.

Subject 1 provided a few additional comments about the study and the SpatialHistory tool. She explained that the window images that SpatialHistory collects may pose security concerns for some individuals and companies. It is possible that images of important and private documents may be logged. She also commented on another possible use for SpatialHistory in a full desktop restoration. She explained that the ability to restore completely the state of a desktop before a system update or a context switch would be useful in quickly resuming tasks.

Subject 2 explained that she used the SpatialHistory tool once or twice over the course of the main stage. The results show that she used it 3 times over the course of a month. She explained that each of these usages occurred during the same instance as described in the third use case in section 5.2.1. This instance occurred when her computer updated and cleared her desktop. When asked if she found the information she was looking for when she used the tool she responded that it was more of an exploration exercise for her. She used the tool to jog her memory of which specific information was in a certain area of her desktop. The thumbnails provided by the viewer tab were sufficient for Subject 2 to recall information. The cues she used were mostly visual. An analysis of the raw window data showed that she kept certain types of windows in relatively similar locations. She recalled that she knew she had particular types of documents in certain areas but she could not remember which documents. Specifically, she recalled that there would be information associated with a particular project she was working on at the time.

Subject 2 explained that she used the spatial search of the tool over the global search. She explained that she queried each individual monitor in her display to recall what was there. As highlighted in the third use case in section 5.2.1, she could immediately eliminate the two leftmost monitors from her search since she only associated those with her email. She described that when looking for a particular file that she remembers specifically, she finds it easily because she keeps a highly organized file system distributed by class or project. As a result her uses of the SpatialHistory tool only include recalling windows or documents that she does not specifically remember.

When asked if she could predict a change in desktop organization or behavior upon continuous usage of the tool, Subject 2 responded that she may with further experience with using a large display as a whole. She explained that the longer she uses a large display, the more her behavior changes to suit it, such as spreading multiple open windows across her desktop simultaneously. A change in the amount of windows open on the desktop is observed for Subject 2 between the two stages. In the preliminary stage, she kept an average of 10 – 11 windows opened, in the main stage she kept 6 – 7 windows opened on average. She closed 596 windows compared to the 111 windows closed during the first stage. Subject 2 worked for 53 hours in the main stage.

When asked for any further comments about the study or her experience, she commented that the tool would be useful for context switches. For example, she said if she were working on two projects at the same time she would often switch between them in relatively quick succession. In this scenario she could see having the ability to recall a previous desktop state useful.

Kevin Logan described that he used the SpatialHistory program when a few times after his computer restarted automatically to install updates, as described in the fourth use case in section 5.2.1. Logan was often busy writing a paper, referencing other documents while using the tool. When the computer would restart, Logan used the tool to reopen the document windows based on their previous location on the desktop. Although in most cases he remembered what the specific documents were, he associated the information with a certain region of the desktop. He queried the region he remembered the documents were in and reopened the windows. He found the information he was looking for each time he used the tool.

Logan used the spatial search each time he used the program, never using the global search. He always remembered spatially where the previously opened documents were situated. He associated certain regions of his display with certain information. Logan did not notice a change in his behavior between the stages such as closing any more or fewer windows between the two stages. This is reflected in the results, having closed 343 windows, similar to the 347 windows closed in the preliminary stage. Additionally, Logan kept a higher number of windows open on his desktop on average. This may be due to the adjustment to using the large display. Logan worked for 454 hours in the main stage.

Logan explained that the SpatialHistory tool is helpful as a recent documents type application. He did not necessarily forget which particular windows were on his display, but he queried his previously opened windows spatially. He remembered that certain information was located at particular regions of his display. He commented that the tool would fit nicely as a built in operating system tool. For example, on the right-click menu brought up next to the mouse at any arbitrary location on the desktop.

Subject 3 did not find the SpatialHistory tool useful. As a result, we formed a separate set of interview questions listed below.

- Why did you not find SpatialHistory useful?
- Can you recall an instance in which you had to reopen a previously used resource? How did you handle this?
- Do you have any other comments on the program?

Subject 3 commented that he could not envision a situation in which the SpatialHistory tool would be useful. He explained that he situates his windows to the right hand side of his display whenever he is working, regardless of the contents of the window. He tends to open a window, find an open area of his display, and situate the resource accordingly. He claimed that this work style did not allow him to recall previously used resources spatially. Due to this, Subject 3's work style did not differ greatly keeping roughly the same amount of windows open at a time between the two phases. When asked to recall a specific situation in which he had to reopen a previously used resource, he said that he simply looks in the 'recently used documents' list included in Windows. He explained that this is enough for him to retrieve documents; he does not think about previous locations. An analysis of the raw window data shows that Subject 3 occasionally keeps certain types of windows in similar locations. Subject 3 went on to suggest that he could see a need for this tool with context switching. He explained that if he were multitasking and quickly changed tasks, he could see a use for the tool. However, he tends to work on a single task until it is completed so it remained only a suggestion.

Subject 4 found that his work schedule forced him to work on other machines besides his large display. As a result, he did not work on enough tasks on his large display to get adequate

experience using SpatialHistory. Regardless, we interviewed him after the study to salvage some useful qualitative insight. We asked him the questions listed below.

- Can you recall an instance in which you had to reopen a previously used resource? How did you handle this?
- Based on your previous answer, how would you describe your mental map of your desktop (if you have one at all)?
- Do you have any other comments on the program?

Subject 4 described several instances in which he has to recall previously opened windows. He described his frequent usage of Firefox's 'undo closed tab' feature which allows the user to reopen a recently closed browser tab. Subject 4 explained that the feature opens 1 tab at a time in order of the most recently closed tab so he frequently has to open several tabs before he finds the one he wants. Subject 4 also uses text editors for writing code. He said that the recent files lists are rarely long enough so he ends up searching files by a recently modified date. When reopening Microsoft Word documents, Subject 4 generally searches a project folder for the document he wants as opposed to using the recent documents feature. Subject 4 described his mental map of his desktop as having all relevant windows for a single project or task in front of him and minor applications such as a music player or messenger in his periphery.

Although Subject 4 did not have an extensive amount of time to use the SpatialHistory tool, his responses raised some interesting points. He often uses 'recent files' features to reopen windows; however he expressed some frustration with existing ones such as Firefox's and some text editors'. Also, based on his responses, Subject 4 may be more apt to recall windows

globally as opposed to spatially on his desktop. Either way, these points make for some interesting conclusions about possible future works, discussed at the end of this paper.

6 Discussion and Future Work

In this section we explore some the various behaviors and uses of the SpatialHistory tool observed during the study. The results show that some of our subjects found the tool useful and some did not. We explore possible reasons for this disparity in this section. We analyze the subjects' responses and use them to support useful scenarios for SpatialHistory and for suggestions of future work. We organize our discussion in terms of our research questions. First we explore how our subjects used the tool (i.e. spatially or globally). Then we explore possible changes in display use and behavior. Finally, we use our findings to propose future work.

6.1 Spatial vs. Global Usage

Here we introduce the idea of varying degrees of locality when performing spatial recalls. We refer to the varying degrees of locality as the size of the areas a user searches when performing a recall. Subject 1, Subject 2, and Logan found the tool useful, each of which reported that they preferred the spatial recall over the global recall. Subject 1 reported that the spatial recall better organized the results from the tool. Subject 2 reported that the spatial recall was useful because it allowed her to narrow down her searches. Subject 2 was able to eliminate two monitors from her search while using the tool as a result of her knowledge of her normal desktop layout. The functionality of SpatialHistory allowed her to narrow her search spatially based on her mental map of her previous layout. This is an example of differing degrees of locality among SpatialHistory uses. Subject 1 described an instance in which her computer updated overnight and cleared her desktop. When she returned the following day she needed to reference a few documents that she knew were in a particular location on her desktop, but she could not remember which specific documents. This occurred before her study period began so

she did not have SpatialHistory, but she considered it an instance in which the tool would have been useful. This is an example of a smaller degree of locality compared to scenario Subject 2 described. Additionally, Logan often reopened pdf documents while using SpatialHistory. He was able to focus the search to the particular location he associated with the documents. In all three instances, Subject 1, Subject 2, and Logan were able to eliminate certain areas, with varying degrees of locality, from their desktop when searching for previously opened windows.

We found that some users organize their desktops in a spatial manner while others do so in a global manner. The users who organize their desktops spatially may be more apt to find SpatialHistory useful. Subject 1, Subject 2, and Logan reported laying out their large displays in specific ways in the preliminary stage of the study. The fact that a user consistently displays certain types of windows in certain areas suggests that they situate this material in a mental, spatial layout and use spatial memory. In addition to the participants' accounts, the analysis of the raw window data also shows that they organize their desktops in a certain way. Our subjects that found the tool useful organized their desktops in a spatial manner. However, our subjects who did not find the tool useful may have a more global mindset. Subject 3 reported that he did not find the tool useful stating that he could not imagine a scenario in which he would use it throughout the study. In the preliminary stage, Subject 3 explained that he placed newly opened windows in available spaces and did not have a set location for certain types of documents. At the end of the main stage, Subject 3 reported that when he needs to reopen a certain document he often looks through recently opened documents lists. With this evidence, we argue that Subject 3 does not organize his desktop in a spatial manner. Interestingly, the data shows that Subject 3 may sometimes organize some windows in certain places, but we argue that if he does not consider his layout to be spatial, then he will not use this information to recall windows spatially.

Further, when he does need to reopen previous windows, he searches an all-encompassing recent documents list. This is analogous to the global recall feature of SpatialHistory. With this, we argue that Subject 3 keeps a mental global layout of his desktop, recalling windows regardless of their spatial distribution. Due to this, SpatialHistory would inherently be an ineffective tool for Subject 3 due to the global nature of his desktop display and memory. Although Subject 3 never used the global recall feature of SpatialHistory, it would have been appropriate given his behavior. We assume that because Subject 3 did not find our tool's intended functionality useful, he was not motivated to use it as a global recollection tool. Subject 4 recalls windows in a similar way, often using recently opened documents lists. The same analysis would apply to Subject 4; however, he reported that he did not have enough of an opportunity to use his large display to properly test the tool. Regardless of their experience with SpatialHistory, both Subject 3 and Subject 4 suggested that the tool could be useful in the event of context switches between projects.

As displays get larger, it may become more difficult to remember which windows were opened. As a result, SpatialHistory is more useful for users with increasingly larger displays. Subject 1 has experience using two different sizes of displays. Her comments on the differences between the two displays are worth analyzing. Subject 1 has explained that she did not initially see a use for the SpatialHistory tool before switching to her larger display. After this change, she had a more difficult time recalling previously opened windows on her larger desktop. As a result, she reported the aforementioned use for the SpatialHistory tool before her scheduled study period began. During the study a series of context switches between projects caused her to spatially recall her desktop. She reports that the larger size of her display, in addition to her context changes, caused a need for SpatialHistory. In addition, Alex Endert had the initial idea

for the tool while using the same larger display that Subject 1 used. This experience is highlighted in the introduction section of this paper. With this evidence, we argue that increasingly larger displays lead to more likely scenarios and use cases for our tool.

Some SpatialHistory uses differed from our initial predictions. First, Subject 1 and Subject 2 each searched for information monitor by monitor to refresh their memory of which windows they had opened. They were not looking for specific documents, but rather used the tool to remind themselves of tasks. This differs from our original scenario in that they were not searching for specific documents, but still retains the spatial aspect of the window recalls. Subject 1 also restored her entire desktop through SpatialHistory. This is not the intended use for the tool, nor is it optimal for the task, but it can still be helpful in this respect.

From the results and analysis presented here, we can conclude that all of our users had a need to recall previously opened windows. Three of our users, including Logan, recalled windows spatially while two appeared to recall windows globally. There seems to be a correlation between how users organize their desktops and how they recall windows. Users that organize their desktops spatially also recall windows spatially. Users who have no discernible spatial organization for their desktop recall globally. We can conclude that some users do recall windows spatially and, in turn, form a mental, spatial layout of their desktop space and use spatial memory. Additionally, SpatialHistory is a useful tool for those that do form these spatial layouts. Possible scenarios in which users perform spatial recalls are outlined in the user cases. To summarize, our users who form spatial layouts of their desktops recalled windows spatially and those who did not recalled windows globally. The scenarios in which users spatially recalled windows included instances in which they needed to reopen a window that they remember being in a certain place on the desktop. This supports our first hypothesis.

6.2 Behavior

In addition to our questions about spatial memory and the usefulness of SpatialHistory, we also observe the possibilities of behavioral shifts as a result of the tool. In this section we look at our raw data and comments from our subjects in relation to how SpatialHistory may affect behavior after extended usage.

SpatialHistory may have the capacity to change the way a user behaves when working on a large display. The data collected from Subject 1 and Subject 2 showed that they kept less windows open on their desktops while using SpatialHistory. This suggests that the ability to recall previously opened windows influenced them to keep a less cluttered desktop, but could also be credited to the tasks performed at that time or another reason unrelated to the use of SpatialHistory. This conclusion clashes with the data collected on Logan in which the amount of windows opened increased from the preliminary stage to the main stage. However, Logan began using his large display at around the time the study started so the data could be skewed as a result. The most likely scenario is that the users did not have enough time with the tool to notice any substantial behavioral shifts. All three users who found SpatialHistory useful reported that they did not notice any change in the way they organized their desktops. More time with the tool is needed to notice any possible effects. Subject 1 and Subject 2 each reported that they could see a change in their behavior if they were given enough time to incorporate the tool into their daily working activity.

SpatialHistory requires an extended amount of time before it can be used. A common scenario requires a user to work on a meaningful task, leave that task for some time, then return to it needing to recall the previous resources. Subject 1 and Subject 2 each only used

SpatialHistory in one instance each over the course of a month. An interesting point is that Subject 1 only worked for 10 hours over the course of the main stage. If she had worked for longer, she may have found more uses for the tool. Similarly, Subject 2 only worked for 53 hours during the main stage. Subject 4 who did not have a chance to use his display extensively over the course of the study, naturally did not find a use for the tool. In order to develop a change in behavior, experience with the tool would have to extend well beyond a few months. We argue, based off of our subjects' comments, that given enough time with the tool, we could observe changes in behavior among users. This supports our second hypothesis.

6.3 Future Work

In this section we take what we learned from the study and apply it to ideas for future research. We consider the fact that a longer user study will yield more results. We look at comments made by our subjects and consider them for potential projects. We also observe comments and reactions from our subjects who did not find SpatialHistory useful to inspire future work.

Although our user study provided us with evidence to support our hypotheses, a longer study would yield more results. Taking into account that our users only used SpatialHistory about once a month, a study lasting 12 or more months would be useful. The focus of this study would be behavioral shifts observed between users working with and without the tool. We have evidence from our study that these shifts may happen, but in order to fully prove SpatialHistory's usefulness we would need a long, extended study. Subject 1 commented that with enough time, she may associate certain types of documents with certain spatial areas more strongly if she continually found SpatialHistory return those types of windows when performing a spatial query.

This suggests that SpatialHistory would not only work with spatial memory, but may also help foster it. This is an interesting concept that could be better explored with a longer user study. In addition to length, more subjects would benefit the quality of results. Since our users must work on large displays the candidate pool becomes shallow. It could be challenging to find enough users, but the effort would be worth the search.

Some functional improvements would benefit SpatialHistory. First, the tool could be ported to work on other operating systems such as Mac OS and Linux. This would allow more types of participants to use SpatialHistory. Currently the tool only allows the user to reopen certain types of windows. The ability to reopen more types of windows such as web browsers and IDEs would improve the user experience of the tool. The tool was also not used often. This may be improved by making it a more integral part of the operating system. Currently it takes 3 clicks to perform a query. For example, if this functionality were added to the right-click menu of the Windows operating system, it may be more accessible for users.

Alternative UI designs would strengthen the usability of SpatialHistory. In our design, we display image thumbnails in rows and columns. We allow the user to leaf through the images in groups of 12 and one by one when they select an image to enlarge. When the program window is small, it is difficult to see more than a few windows in a single glance in the wide view and difficult to see an entire single window when one is selected. An alternative design would be to allow the user to leaf through the image in a separate window which automatically resizes in order to compensate for different sized windows. Another alternative would be to design a sleek Viewer similar to the way Windows allows the user to cycle through open windows on the desktop. A more aesthetically pleasing design would lead to a better user experience. One possibility would be to combine the Table and Viewer tabs into a timeline

which visually displays images and text information in order of access time. This would potentially provide a crisp and fast visual representation of the window information that would allow the user to quickly get an overview of their past window positioning.

A system which constructed visual patterns of desktop usage for the user would provide an interesting study. SpatialHistory logs window images and along with desktop coordinates in its database. This information could be used to construct a visual representation of how a user organizes their desktop over the course of extended use. This could be used to compare a user's mental map of their desktop to an actual visual representation.

Our subjects often brought up the concept of context switching between different projects. SpatialHistory certainly aids in this respect by allowing the user to recall previously opened windows, but a tool focusing solely on restoring entire desktop states from one context to another, similar to the Time Machine discussed earlier, would lead an interesting study [14]. Subject 3, who did not find SpatialHistory useful, commented that he could see uses for the tool in project context switches. This would provide a useful tool for individuals who do not appear to form a spatial layout of their desktop, in addition to those who do. A tool allowing users to easily and quickly switch between desktop states and contexts would yield potentially useful functionality and an interesting study.

7 Conclusion

We have designed a tool, SpatialHistory, which keeps track of window history based on time accessed and spatial location. The tool allows the user to query a certain spatial location on their desktop and see which windows were opened in that area and at which time.

Through a user study, we found that some users form a spatial layout, using spatial memory, of their large display associating certain types of windows and information with certain locations on their desktop. The users that formed spatial layouts of their desktops preferred to search their window history spatially rather than globally and, as a result, found the SpatialHistory tool useful. Our subjects who did not find the tool useful appeared to view their desktops globally and did not seem to form spatial layouts. Most of our users commented that the tool would be useful for context changes.

SpatialHistory has revealed avenues of future work as a result of our user study. SpatialHistory has shown potential to cause behavioral shifts by users if given enough time with the tool. A longer user study, of 12 months at least, would likely result in evidence for changes in behavior among users. In addition to conducting a longer user study for SpatialHistory, a separate tool which focuses on restoring past desktop states could be researched and developed based on user feedback. Recalling previous windows and information for users with large displays becomes more difficult as display size increases. The development of tools which aid in restoring desktop contexts and recalling windows based on spatial memory lend themselves to future research.

Bibliography

- [1] Tan, D. S., Gergle, D., Scupelli, and Pausch, R. *Physically large displays improve performance on spatial tasks*, ACM Transactions on Computer-Human Interaction (TOCHI), v.13 n.1, p.71-99, March 2006.
- [2] Slater, M., Usoh, M., and Chrysanthou, Y. *The influence of dynamic shadows on presence in immersive virtual environments*, Selected papers of the Eurographics workshops on Virtual environments '95, p.8-21, December 1995, Barcelona, Spain.
- [3] Prothero, J. and Hoffman, H. (1995). *Widening the Field of View Increases the Sense of Presence in Immersive Virtual Environments*. (HITLab Tech Report R-95-5) Seattle: University of Washington, Human Interface Technology Laboratory.
- [4] Shupp, L., Andrews, C., Dickey-Kurdziolek, M., Yost, B., and North, C. *Shaping the Display of the Future: The Effects of Display Size and Curvature on User Performance and Insights*, Human-Computer Interaction, Volume 24, Issue 1 & 2, pages 230-272, January 2009.
- [5] Ball, R., and North, C. *An Analysis of User Behavior on High-Resolution Tiled Displays*, Tenth IFIP International Conference on Human-Computer Interaction (INTERACT 2005), pg. 350-364, (Sept 2005).
- [6] Ball, R., and North, C. *The Effects of Peripheral Vision and Physical Navigation in Large Scale Visualization*, Proceedings of Graphics Interface 2008, pp. 9-16, June 2008.
- [7] Ball, R., and North, C. *Realizing Embodied Interaction for Visual Analytics through Large Displays*, Computers & Graphics (C&G), 31(3): 380-400, June 2007.

- [8] Czerwinski, M., Robertson, G., Meyers, B., Smith, G., Robbins, D., and Tan, D. *Large display research overview*, CHI '06 extended abstracts on Human factors in computing systems, April 22-27, 2006, Montréal, Québec, Canada.
- [9] Bi, X., Balakrishnan, R. *Comparing usage of a large high-resolution display to single or dual desktop displays for daily work*, Proceedings of the 27th international conference on Human factors in computing systems, April 04-09, 2009, Boston, MA, USA.
- [10] Andrews, C., Endert, A., and North, C. *Space to Think: Large, High-Resolution Displays for Sensemaking*, to appear in Proc. of ACM CHI 2010, 10 pages, April 2010.
- [11] Leifert, S. *The influence of grids on spatial and content memory*, Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems, May 07-12, 2011, Vancouver, BC, Canada.
- [12] Jetter, H., Leifert, S., Gerken, J., Schubert, S., and Reiterer, H. *Does (multi-)touch aid users' spatial memory and navigation in 'panning' and in 'zooming & panning' UIs?* Proceedings of the International Working Conference on Advanced Visual Interfaces (ACM New York, NY, USA, 2012), 83-90.
- [13] Cockburn, A. *Revisiting 2D vs 3D implications on spatial memory*, Proceedings of the fifth conference on Australasian user interface - Volume 28, (Australian Computer Society, Inc. Darlinghurst, Australia, Australia 2004), 25-31.
- [14] Rekimoto, J. *Time-machine computing: a time-centric approach for the information environment*, Proceedings of the 12th annual ACM symposium on User interface

software and technology, p.45-54, November 07-10, 1999, Asheville, North Carolina, United States.

- [15] Freeman, E., and Gelernter, D. *Lifestreams: a storage model for personal data*, ACM SIGMOD Record, v.25 n.1, p.80-86, March 1996.
- [16] Fertig, S., Freeman, and E., Gelernter, D. “*Finding and reminding*” reconsidered, ACM SIGCHI Bulletin, v.28 n.1, p.66-69, Jan. 1996.
- [17] Allen, J. F., *Maintaining knowledge about temporal intervals*, Communications of the ACM, v.26 n.11, p.832-843, Nov. 1983.
- [18] Henderson, S. *How do people organize their desktops?*, CHI '04 extended abstracts on Human factors in computing systems, April 24-29, 2004, Vienna, Austria.
- [19] Robertson, G., van Dantzich, M., Robbins, D., Czerwinski, M., Hinckley, K., Ridsen, K., Thiel, D., and Gorokhovskiy, V. *The Task Gallery: a 3D window manager*, Proceedings of the SIGCHI conference on Human factors in computing systems, p.494-501, April 01-06, 2000, The Hague, The Netherlands.
- [20] Smith, G., Baudisch, P., Robertson, G., Czerwinski, M., Meyers, B., Robbins, D., and Andrews, D. 2003. *Groupbar: The taskbar evolved*, Proceedings of the Australian Computer-Human Interaction Special Interest Group (OZCHI).
- [21] Java Native Access Home Page. <https://github.com/twall/jna> . Accessed Oct 19 2012.
- [22] Adobe Website. <http://www.adobe.com/>. Accessed Oct 19 2012.

- [23] Handle Website. <http://technet.microsoft.com/en-us/sysinternals/bb896655.aspx>. Accessed Oct 19 2012.
- [24] Sawyer, B., Quek, F., Wong, W. C., Motani, M., Yee, S. L., and Pérez-Quiñones, M. A. *Using physical-social interactions to support information re-finding*, Proceedings of CHI '12, p. 885-910, May 2012.
- [25] Kawase, R., Papadakis, G., Herder, E., and Nejdl, W. *The impact of bookmarks and annotations on re-finding information*, Proceedings of the 21st ACM conference on Hypertext and hypermedia, June 13-16, 2010, Toronto, Ontario, Canada.
- [26] Capra, R. G. III, and Pérez-Quiñones, M. A. *Using Web Search Engines to Find and Refind Information*, Computer, v.38 n.10, p.36-42, October 2005.
- [27] Capra, R. G. III, and Pérez-Quiñones, M. A. *An investigation of finding and re-finding information on the web*, Virginia Polytechnic Institute & State University, Blacksburg, VA, 2006.
- [28] Capra, R. G. III, Pinney, M., and Perez-Quinones, M. A. (2005). *Re-finding is Not Finding Again*, Technical Report TR-05-10, Computer Science, Virginia Tech.
- [29] Cockburn, A. and McKenzie, B. *Evaluating the effectiveness of spatial memory in 2D and 3D physical and virtual environments*, Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves, April 20-25, 2002, Minneapolis, Minnesota, USA.

- [30] Tan, D. S., Pausch, R., Stefanucci, J. K., and Proffitt, D. R. *Kinesthetic cues aid spatial memory*, CHI '02 extended abstracts on Human factors in computing systems, April 20-25, 2002, Minneapolis, Minnesota, USA.
- [31] Chun, M. and Jiang, Y. (1998). *Contextual Cueing: implicit learning and memory of visual context guides spatial attention*, *Cognitive Psychology* 36 (1): 28–71.
- [32] McNamara, T.; Hardy, J.; Hirtle, S. et al. (1989). *Subjective hierarchies in spatial memory*, *Journal of Experimental Psychology* 15 (2): 211–227.
- [33] Corsi, P. M. (1972). *Human memory and the medial temporal region of the brain*, *Dissertation Abstracts International* 34 (2): 891.
- [34] Mammarella, I.C.; Pazzaglia, F.; Cornoldi, C. (2008). *Evidence of different components in children's visuospatial working memory*, *British Journal of Developmental Psychology* 26 (3): 337–355.
- [35] Robertson, G., Czerwinski, M., Larson, K., Robbins, D. C., Thiel, D., and van Dantzig, M. *Data Mountain: using spatial memory for document management*, *Proceedings of the 11th annual ACM symposium on User interface software and technology*, p.153-162, November 01-04, 1998, San Francisco, California, United States.