

# **ScreenTrack: Using Visual History for Self-tracking Computer Activities and Retrieving Working Context**

Donghan Hu

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 and Applications

Sang Won Lee, Chair  
Scott McCrickard  
Denis Gracanin

May 15, 2019  
Blacksburg, Virginia

Keywords: Human-Computer Interaction, self-tracking, Time-Lapse Video, Task Resumption, Mental Context Reconstruction

# ScreenTrack: Using Visual History for Self-tracking Computer Activities and Retrieving Working Context

Donghan Hu

## Abstract

People spend a significant amount of time using computers at work, at home, or school. Given users switch tasks and are frequently interrupted or distracted while working, reconstructing working context is inevitable. For example, users sometimes need to revisit an arbitrary task from the past to retrieve necessary information (e.g., webpages, files). In this scenario, retrieving working context can be time-consuming or even impossible; users may rely on their memory and may not be able to retrieve the relevant documents that they used before. Sometimes application provides a chronological history of recently opened documents (files, websites). However, it can be challenging to find the right information they need as there are many and users may not recognize from the text-based data (e.g. web page titles, document file name). Therefore, helping them reconstruct mental context and retrieving relevant applications and files can enhance overall productivity. To that end, the concept of self-tracking, which is widely used in health and fitness, is applied to the context of computer usage. In particular, the idea of using a history of a computer screen can provide visuals that users can associate with existing meta-data (file location, web page URL, time). A user can reconstruct working context from the screen visual that they recognize. The idea of using a visual history of a computer screen activities is tested through the development of *ScreenTrack*, a program that captures a computer screen regularly and let a user watch a time-lapse video made of computer screenshots, and retrieve applications, files, and web pages from a snapshot of a screen. I hypothesize that the chronological history of computer screen activities can effectively help users navigate visual working context and retrieve information that is associated with a snapshot. Through a controlled user study, it was found that participants were able to retrieve information that they were asked more quickly with *ScreenTrack* than the control condition with statistical significance ( $p < 0.005$ ). Besides, participants gave positive feedback on the software that they would like to use such software in their computers in various context, but expressed potential concerns of using such software for privacy and computer storage problems. In this thesis, I motivate the need of such software, review the related work, share the design consideration, and introduce design and implementation process, validate the effects of *ScreenTrack* with a controlled user study.

## General Audience Abstract

Nowadays, people spend a significant amount of time using computers at work, at home, or school. Users switch software frequently and are often interrupted or distracted while working. Hence recalling previous working context is inevitable for computer users. Recalling previous working context can take lots of time or even impossible. Because users may rely on their own memory and may not be able to recall and retrieve the relevant documents that they used before. Sometimes software provides a history of recently opened documents (files, websites). However, it can be challenging to find the right information they need as there are many recorded information. And users may not recognize documents of interest from the textual data (e.g. web page titles, document file name). Therefore, helping people restore previous working context and reopening relevant software and files can enhance overall productivity.

I designed and developed a software, called ScreenTrack. This software can take pictures of current computer screen regularly and store them. Later, users can watch a video made of stored screenshots. Based on this video, individuals can recall their previous computer activities and reopen closed software, websites, files, and documents from a snapshot. Through a controlled user study, I found that participants were able to retrieve previous computer activities more quickly under the help of ScreenTrack than without ScreenTrack. With ScreenTrack, participants spend 27.1 seconds on average to reopen a previous closed website, 37% faster than without it. Furthermore, participants gave positive feedback on this software that they would like to use ScreenTrack in the future for various purposes, for doing researches and reading papers.

## **Acknowledgments**

First and Foremost, I would like to acknowledge and thank my advisor Dr. Sang Won Lee for his professional and invaluable guidance during the research. He provided excellent advice throughout my studies at Virginia Tech, both professionally and personally. Without his help, I could not complete my education at the Computer Science Department, Virginia Tech.

I would like to thank my committee members Dr. Denis Gracanin and Dr. Scott McCrickard for their time and excellent suggestions.

I would also like to thank my friends Mantong Wang, Tiancheng Ying, Jing Luo, and Xuan Liu. They helped me in the process of finishing the thesis and the system. I am appreciated to them as volunteers to help me to test the system and provided valuable feedback.

Lastly, I am deeply thankful to all my family members. Thank you for your love, support, encouragement, understanding, and sacrifices. Your love and encouragement have been the most valuable thing in my life.

## Table of Contents

<b>Abstract.....</b>	<b>ii</b>
<b>General Audience Abstract .....</b>	<b>iii</b>
<b>Acknowledgments .....</b>	<b>iv</b>
<b>List of Figures.....</b>	<b>vii</b>
<b>List of Tables .....</b>	<b>ix</b>
<b>Chapter 1 : Introduction .....</b>	<b>1</b>
1.1    Problem Definition and Research Questions .....	1
1.2    Research Contributions: Utilizing a Visual History of Computer Screens to Retrieve Working Context .....	2
1.3    ScreenTrack: Time Lapse Video of Screenshots .....	2
1.4    Thesis Outline .....	4
<b>Chapter 2 : Related Work .....</b>	<b>5</b>
2.1    Self-Tracking for Behavioral Changes.....	5
2.2    Self-Tracking to Facilitate Self-reflection.....	10
2.3    Mental Context Reconstruction for Task Resumption and Reflection .....	11
2.4    Interruptions in Working and Studying.....	14
2.5    Utilizing Visual Histories for Information Retrieval and Mental Reconstruction .....	15
2.6    Conclusion and Thoughts.....	17
<b>Chapter 3 : Design Goals.....</b>	<b>19</b>
3.1    Create a Replayable Visual History of Computer Screens that Works Across Programs.....	19
3.2    Capture a Screenshot with Metadata Available and Display the Visual History with Metadata. ....	19
3.3    Display Metadata and Screenshots Together. ....	20
3.4    Provide Various Ways to Capture and Navigate Temporal History of a Computer Screen. ....	20
3.5    Make the Disk-space Requirement Manageable and Address Privacy Concerns.....	21
<b>Chapter 4 : Software Design .....</b>	<b>23</b>
4.1    Recording Configuration User Interface .....	23
4.2    ScreenTrack Captures and Stores Screenshots in a Local Folder. ....	25
4.3    Frontmost Running Software Identification and Data Collection.....	27
4.4    Time-Lapse Visualization Window .....	29
<b>Chapter 5 : User Study .....</b>	<b>36</b>
5.1    Pilot Experiment and Implementation.....	36
5.1.1 Participants .....	36
5.1.2 Process of the Pilot Experiment.....	36
5.1.3 Findings in Pilot Experiment .....	36
5.2    User Study Design and Implement.....	37
5.2.1 Participants .....	37

5.2.2 User Study Designed .....	37
5.2.3 Web-Surfing Task and Recall Task.....	38
<b>Chapter 6 : Result .....</b>	<b>41</b>
6.1 Result and Analysis for Time Efficiency of Working Context Retrieval .....	41
6.2 Result and Analysis for Perceiving the Values of Using ScreenTrack .....	46
6.2.1 ScreenTrack is Easy to Use and Intuitive.....	47
6.2.2 Single Visual History with Screenshots .....	47
6.2.3 Participants Could Imagine Using ScreenTrack.....	48
6.2.4 Participants Were Concerned with Privacy Issue.....	48
6.2.5 Eating up Computational Resources (Disks / CPU).....	49
6.3 Limitations .....	50
<b>Chapter 7 : Conclusion and Future Work.....</b>	<b>51</b>
7.1 Conclusion.....	51
7.2 Future Work .....	51
<b>References .....</b>	<b>54</b>
<b>Appendix.....</b>	<b>58</b>
Appendix A - Pre-task survey .....	58
Appendix B - Study Procedure.....	60

## List of Figures

Figure 1: User Interface of ScreenTrack. It shows a time-lapse video of screenshots to present a visual history of computer activities and retrieve working context from.....	3
Figure 2: Li et al.'s five stage model of personal informatics. This model includes the process of barriers cascade, iterative stages, user and/or system driven stages, and uni- or multi-faceted. ....	6
Figure 3: Epstein et al., Lived informatic model of personal informatics. This model includes the process of deciding, selecting, tracking & acting, and lapsing. ....	7
Figure 4: Wearable sports camera: GoPro. Users can self-track their physical activities and create a time-lapse video with it. ....	9
Figure 5: Self-tracking software: RescueTime. This software can track users' computer activities on computers and provides summary reports. ....	10
Figure 6: Do not interrupt a programmer (Heeris, 2013). This cartoon well-explain why mental context reconstruction is needed for resuming task, which is challenging. ....	13
Figure 7: Graphic History View of Web Browsing (Ayers and Stasko, 1995). This history view demonstrates how a user navigate web pages over time and what web page they visited from which webpage.....	17
Figure 8: Recording Configuration User Interface. A user can set default values and start self-tracking to record computer screen activities. ....	24
Figure 9: Screenshots folder hierarchy. These folders show how ScreenTrack saves screenshots and metadata into local disk.....	26
Figure 10: Data structure stored in a JSON file for the "Browser" category. ....	28
Figure 11: Data structure stored in a JSON file for the "Productivity" category. ....	29
Figure 12: Data structure stored in a JSON file for the "Programming" category. ....	29
Figure 13: Data structure stored in a JSON file for the "Other" category. ....	29
Figure 14: Time-Lapse Visualization Window. This window has three parts: Time-lapse Video Display, Metadata Panel, and Configuration. ....	30
Figure 15: User Interface Region for Navigating Time of Time-lapse video.....	31
Figure 16: Screenshots before cropping (Top) and after cropping (Bottom). The example application in this image is Finder. ....	34
Figure 17: Expanded drop down menu for different options of time period. ....	35
Figure 18: The procedure of user study. The whole user study contains seven parts: 1) consent Form, 2) Survey, 3) Web-surfing Task 1, 4) Web-surfing Task 2, 5) Recall Task 1, 6) Recall Task 2, and 7) Exit Interview.....	38
Figure 19: Average time spends on retrieving a previous closed website on the control condition and experimental condition. ....	42
Figure 20: Distribution of data points for control condition and experimental condition. ....	43
Figure 21: Boxplot for data points in control condition and experimental condition. ....	44

Figure 22: Average time spends on retrieving a previous closed website for each trial on the control condition and experimental condition. .... 45

Figure 23: In experimental condition, the usage of three features in ScreenTrack to help participants retrieve previous closed websites. .... 46

Figure 24: College view for self-reflection during long time window. .... 53

## List of Tables

Table 1: Related information collected based on software classification.....	28
Table 2: Four sequences for user study with two factors.....	39

## Chapter 1: Introduction

### 1.1 Problem Definition and Research Questions

Nowadays, people spend a significant amount of time on their computers for various purposes like work, study, communication, and entertainment. Most of these tasks are typically done using a single computer. Different kinds of tasks can be interleaved while using the computer (e.g., writing an email, writing a document, watching a movie on a computer). Therefore, a user can easily get distracted with other functions of a computer (emails, social media), or interrupted externally - lunch break, conversation, etc., - during work (Czerwinski et al., 2004). Therefore, frequent context switch and interruption are inevitable (Jin and Dabbish, 2009) and resuming interrupted tasks happen frequently as well while people are using the computer (Trafton et al., 2003). After being interrupted or paused from their work, users need extra time to recall and pick up their previous tasks from where they left off, and they need to retrieve working context - programs that they used, files that they worked on, web pages that visited.

People desire to use time efficiently and to enhance their productivity at work (Rule et al., 2015). There are two potential ways to improve productivity, 1) to change their behaviors (e.g., spending less time on undesirable activities, reducing the number of context switch) via self-reflection (Cuttone et al., 2014). 2) to save time that takes in resuming interrupted tasks (e.g. taking note to help remember the breaking point of the task). The latter case requires a user to recall what they did and retrieve the working context for task resumption. Those studies mean that users need support in recalling the working context for self-reflection and resuming tasks. However, remembering previous work and retrieving working context primarily rely on human memory. Or one can rely on computer programs that archive chronological history files, documents, or web pages that were visited. However, this can be time-consuming and inefficient (Rule et al., 2017). A user needs to recall various information on different levels about paused work before continuing those unfinished tasks (Nahum et al., 2008). For example, they need to figure out whether they were simply productive or not. On the other hand, a user needs to recall the low-level information such as website URLs, a document's file path, and paper's title. This kind of information is challenging to remember (Parnin and Rugaber, 2012). Therefore, users may spend significant time on recalling the context or retrieving specific files and website, which they sometimes fail and may feel frustrated. Based on this situation, the computational support to provide self-tracking, for mental context recovery and working context retrieval in productivity setup, has become necessary (Kalnikaite et al., 2010). These kinds of technologies and tools can benefit people who would like to enhance their productivity through self-tracking and effective task-resumption. Such supporting can play significant roles in improving the recall and making it more effective and relaxing. It can help users become aware of their behaviors, return to past work quickly, spend less time on needless tasks, and enhance their efficiency of productivity. Specifically, the research questions I attempt to answer in this thesis are as follows:

- *(RQ1) Does using the visual history of a computer help users retrieve relevant information more quickly than other existing methods?*
- *(RQ2) How do users perceive the values of using ScreenTrack in their computer usage?*

In this work, to answer these research questions, I have designed and implemented an interactive software, ScreenTrack. This software focuses on tracking computer screen regularly and capture working context - applications that are opened, files that are being used, web pages that are being visited - supporting mental context reconstruction using visual histories of computers based on design implications suggested in previous work (Doherty et al., 2011; Rule et al., 2017). The system was implemented to focus on self-tracking and to record related information in the process of using computers. Additionally, the system provides users with a function that could review their history and retrieve recorded information.

## **1.2 Research Contributions: Utilizing a Visual History of Computer Screens to Retrieve Working Context**

The goal of this research is to design and implement a system that addresses the problems of retrieving working context from the past by self-tracking one's computer screen. In this paper, to address these problems, I have designed and implemented an interactive software: ScreenTrack. In this system, my approach is to provide a visual history of computer screenshots offering ways to seek information through photographic information in addition to existing history methods and to help users retrieve previous application that they have used in the past from a snapshot of the computer screen. This software focuses on supporting information seeking and reconstructing working context using visual histories of computers based on design implications suggested in previous works (Ayers and Stasko, 1995; Mayer and Bederson, 2003; Rule et al., 2017). ScreenTrack is aiming to enhance productivity by reducing the time it takes to retrieve and reconstruct the working context, while people use the computer as their main tools. The topic of our paper is not unique among software (Ayers and Stasko, 1995; RescueTime; Jeuris and Bardram, 2016; Rule et al., 2017), it provides a novel way to associate screenshots of the computer screen with various metadata. Providing users with a visual history of past activities can help individuals retrieve their previous contexts. However, there are several limitations that existed in ScreenTrack. For example, ScreenTrack could not fit and retrieve related information to all kinds of software (e.g., Social software has no useful related information). Besides, the privacy problem should be considered more in future work.

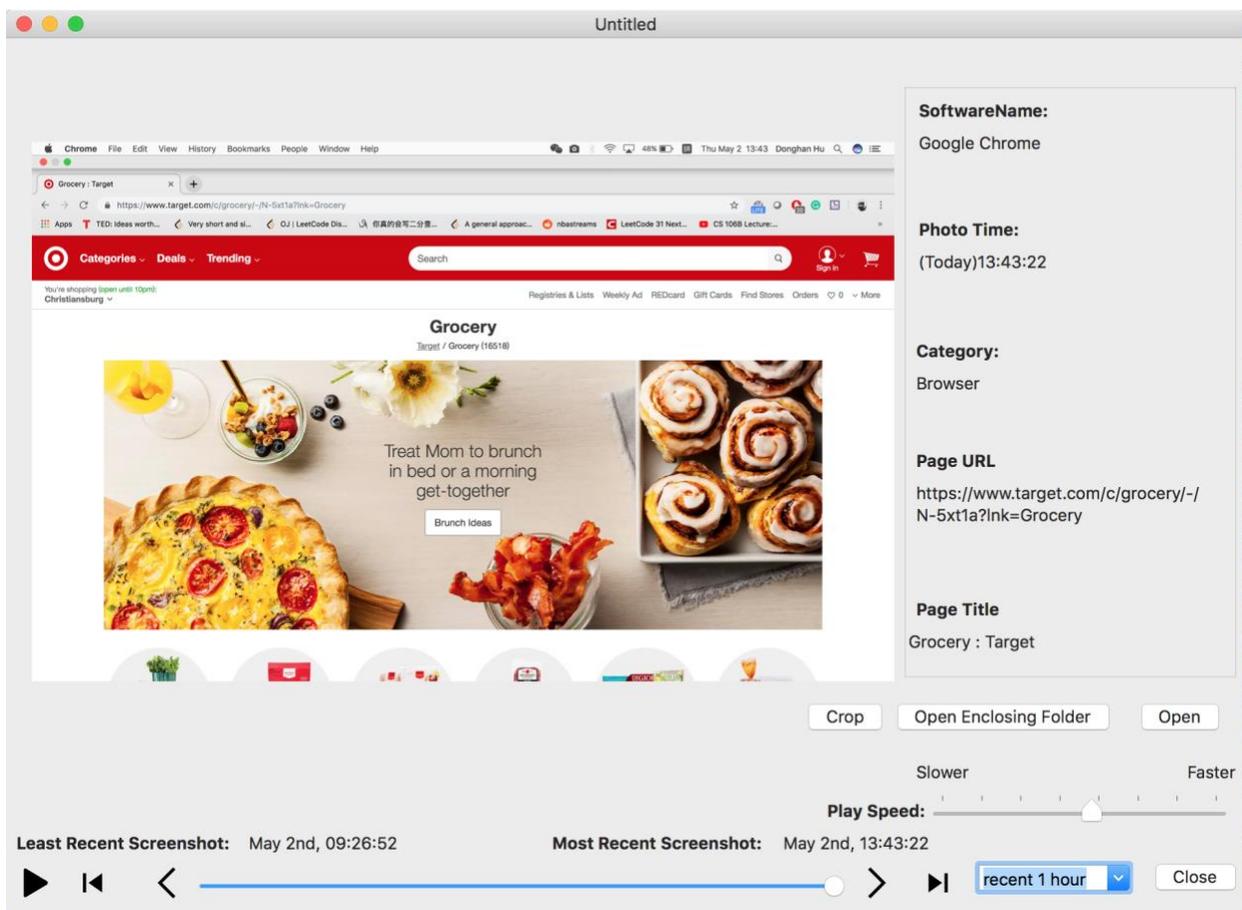
Following are the research contributions from this system and paper:

- The idea of associating visual history of screenshots with metadata and utilizing the connection for ways help users retrieve working context from a screenshot.
- The realization of this idea is through a program, ScreenTrack, which provides an efficient method for people self-tracking their computer activities and generates a time-lapse video of past screenshots.
- Quantitative and qualitative evidence that the system provides an effective and efficient way to retrieve the working context by navigating a time-lapse video.

In the following subsection, ScreenTrack is introduced, and the user study result is summarized briefly.

## **1.3 ScreenTrack: Time Lapse Video of Screenshots**

In this thesis, I designed and developed this system providing the user with a new method to self-tracking based on screen recording (Figure 1). According to the previous work, people remember more details about their daily tasks after reviewing time-lapse video than only rely on their memory and textual summaries (Czerwinski and Horvitz, 2002). The images that recorded the user's daily tasks and computer activities also corroborated the above finding (Sellen et al., 2007). After screen recording, while running ScreenTrack, users could run this system for retrieving working context from a snapshot from the visual history of screenshots. A previous paper stated that one could digitally capture virtually everything they do, see, and hear on the horizon in the world they lived (Hollan, 2012). According to this fact, displaying screenshots which are sorted by time is a useful method while informing user their past computer activities.



**Figure 1: User Interface of ScreenTrack. It shows a time-lapse video of screenshots to present a visual history of computer activities and retrieve working context from.**

The system has two main vital features, 1) Recording and 2) Replaying.

Firstly, the system uses the screen capture function to track screen activities. With a specific time-interval set by users, the system takes a screenshot, identifies the active software (frontmost application), saves the screenshot to a local folder and stores related information.

Secondly, ScreenTrack provides a user with a time-lapse video of one's computer. The user can review their previous work, browse the screenshots to recall computer activities, and reopen

software, websites, and files per snapshot. Through this method, people can easily and quickly find their previous works.

A controlled user study was conducted to verify the effectiveness of ScreenTrack. Specifically, 21 participants were asked to surf the web for online shopping and pick their favorite furniture. Later, they were asked to retrieve the webpage that has the product of their choice. The time it takes to retrieve the web page in ScreenTrack condition was shorter than the one in the control condition, in which they were allowed to use existing methods (web browser history or browsing again). The difference was statistically significant ( $p < 0.005$ ). Besides, the participants were asked if they would like to use such software on their computer, and the majority of them answered that they would like to use ScreenTrack in various context.

There are several potential problems, and limitations existed in ScreenTrack. Firstly, ScreenTrack is designed for a short time window. Currently, users cannot view their visual history before seven days ago. And in my user study, participants were asked to check their past activities for only one hour. Secondly, in the procedure of user study, I did not validate the work for higher-level cognitive tasks, such as reconstructing mental context, facilitating self-reflection, and eliciting behavioral changes. Designing new user studies to test the above values is a future plan. Thirdly, I did not consider effective methods to help people protect their personal information and save local disks' space. Privacy problem and disk space requirements are two main problems that participants mentioned in the user study.

#### **1.4 Thesis Outline**

In Chapter 2, I introduce background knowledge and related work about previous studies. This thesis provides a literature review corresponding to the thesis topic and associated questions. In Chapter 3, I propose design considerations for ScreenTrack. In Chapter 4, this part discusses the designing and development of ScreenTrack. In Chapter 5, this thesis provides how I implemented the pilot study and the user study. In Chapter 6, I analyze the results from the user study and provide the validation of ScreenTrack. Based on this result, I discuss the limitations that existed in ScreenTrack and user study. In Chapter 7, I state conclusion and talk about future work for ScreenTrack.

## Chapter 2: Related Work

This thesis draws ideas from various previous works done in HCI. To introduce these previous studies, I provide an overview of the related work in self-tracking(self-monitoring), self-reflection, mental context reconstruction, interruptions in working and studying, and visual histories in this chapter. The study of this knowledge can help us understand more about why I would design this system, how to develop this system, and what should I do about implement this system, ScreenTrack.

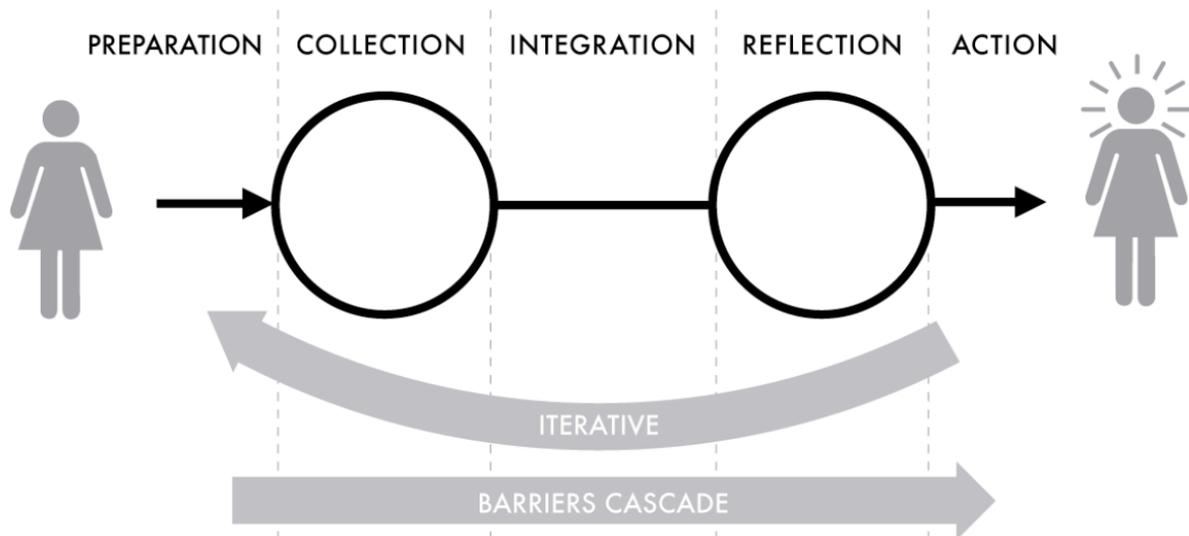
### 2.1 Self-Tracking for Behavioral Changes

Self-tracking (or self-monitoring) is the procedure of recording one's not only activities, behaviors, but also thoughts. Even though the idea of using technology and devices to monitor people's own activities is a quite new concept for most people. This concept could be traced back to 1970 from the area of psychology (Kopp 1988). At that time, Doctors and researchers used personal data collected by self-tracking into therapy and medical experiments. To be more specific, self-tracking served an assessment function and treatment function within behavior therapy. (Kopp 1988; Korotitsch and Nelson-Gray, 1999). As time goes by, people have gradually increased awareness of their own behaviors and activities. In addition, self-tracking technology has been widely designed in sensing and monitoring applications. Self-tracking - also known as the quantified self (QS) movement - is not only used in these medical fields but has gradually become a part of people's daily life (Choe and Lee, 2015). People are able to use methods and tools to track numerous types of personal information and data, e.g., health (Fitbit), physical activities (Apple Health), and web surfing activities (RescueTime).

Personal informatics, or collecting and reflecting on personal information and data, has become increasingly prevalent. In the self-tracking, personal informatics can serve various goal-driven purposes, for example, tracking weight loss, recording physical activity, monitoring sleep quality and tracking personal payments (Epstein et al., 2015; Rooksby et al., 2014). In addition, using collected personal information have been explored in applications in order to improve various aspects of an individual life of users, especially in the application for health and fitness, physical exercises (Hoyle et al., 2014), dietary monitoring (Choe et al., 2014), taking a break (Luo et al., 2018), fertility care (Schroeder et al., 2018), etc. Due to the prevalent of self-tracking technology and requirements of personal goals, researchers have studied how people use personal information in their daily life.

The first and most commonly used model for understanding how people use personal informatics tools is Li et al.'s, five-stage model. This is a model focused on describing how individuals use personal informatics thoroughly, which called the five-stage model (Li et al., 2010) (Figure 2). This model characterizes how people transfer their behaviors between five stages and then iterate them again. This module consists of five phases: Preparation, Collection, Integration, Reflection, and Action. Besides, this model emphasizes the iterative nature of these stages and potential barriers. Among these five phases, the collection phase is data gathering. The integration phase is the step that uses prepared data for future to reflect upon, and the reflection stage is the step that users examine and explore their personal collected data. In this thesis paper, my software:

ScreenTrack is focused on the three phases of this model: Collection, Integration, and Reflection phases.



**Figure 2: Li et al.'s five stage model of personal informatics. This model includes the process of barriers cascade, iterative stages, user and/or system driven stages, and uni- or multi-faceted.**

Based on Li et al.'s five-stage model, Epstein et al. provided a new informatics model shown in Figure 3 (Epstein et al., 2015) aiming at personal informatics. This lived informatics model includes five processes: 1) deciding to track, 2) selecting methods, 3) tracking and acting, as an overview of collection, integration, and reflection, 4) lapsing of self-tracking and 5) resuming from lapsing. This model integrated Collection, Integration, and Reflection phases together and brought up with a new idea of lapsing and resuming. Unlike the model built by Li et al., this model focused on reasons why people lapse in their tracking and why they stop tracking entirely. In addition, resuming phase was a new step compared with Li et al.'s five-stage model above. For resuming step, this model showed that resuming previous contexts is essential after being interrupted or a long period of time. Based on personal informatics models in previous research, I can understand how people use their personal information. Therefore, ScreenTrack should focus on tracking and acting phase to help people use their personal data effectively.



**Figure 3: Epstein et al., Lived informatic model of personal informatics. This model includes the process of deciding, selecting, tracking & acting, and lapsing.**

Researchers have studied the motivations of self-tracking activities from experiments and surveys. In Choe et al. research, the motivation of self-tracking activities can be categorized into three main classes: (1) to improve their health, (2) to improve other aspects of life, and (3) to find new experience of life (Choe et al., 2014). For the first category, people track one or more health-related items with an aim to improve aspects of health, like, keeping fitness and weight control. To be more specific, in our daily life, a number of mobile applications and software aim at helping people know their bodies. In this self-tracking procedure, people usually monitor data like calorie consumption, walking distance, and sleeping quality. Self-tracking is an approach to a better life, not only a data collecting method. For the second category of improving other aspects of life, the paper mentioned that people who are either software engineers or students wanted to find ways to optimize their work and maximize learning. Hence, people want to maximize work performance and to be mindful through the process of self-tracking. For example, some people would like to “Track time to know the current use of time and ways to be more efficient” (Choe et al., 2014). About the third class, this is a complex category with various motivations. People in this category often had no specific goals in mind when deciding to start to self-tracking. According to this fact, these individuals might quickly discover interesting patterns from data that led to data collection becoming habitual. In the paper, the authors gave an

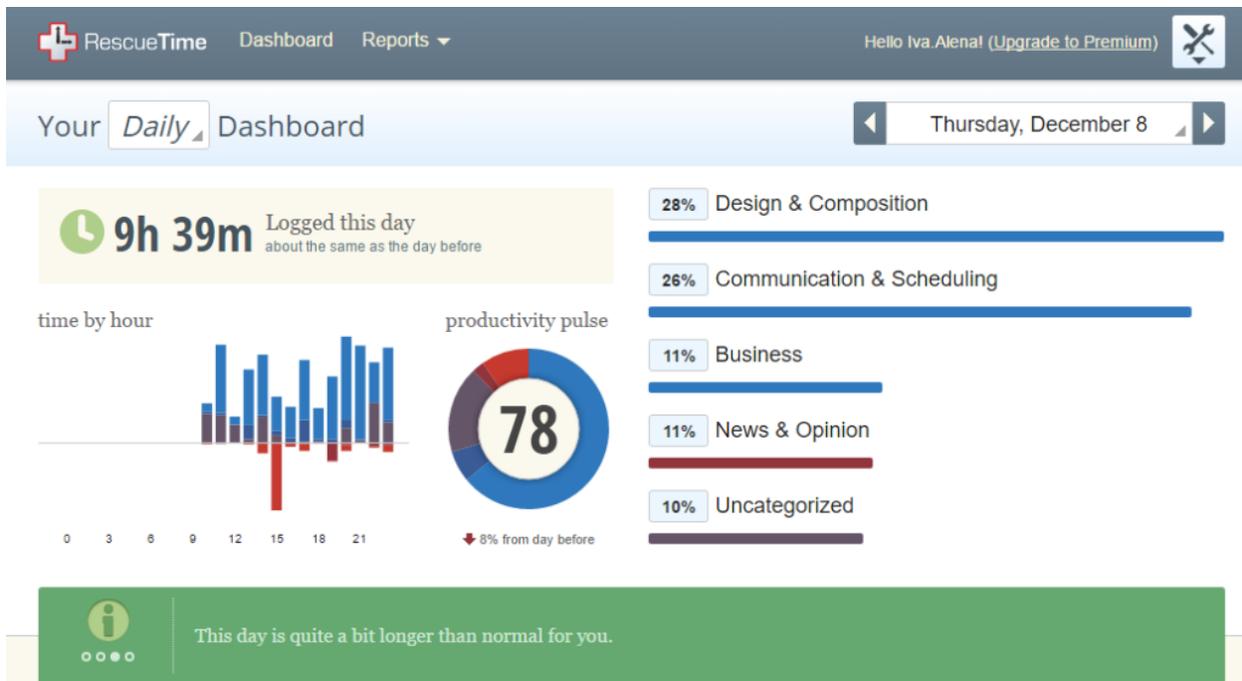
example of logging walked streets. People can go somewhere else new according to their self-recorded information. Knowing people's different motivations can help me design software aiming at specific kinds of people. ScreenTrack should be designed to help track their computer activities and improve their productivity during the procedure of working and studying.

Nowadays, there are many existing systems, tools, applications, and devices that can help people do self-tracking, and can help them in the collecting phase. Each tool and method has its own advantages and disadvantages. Based on the different kinds of collected data and information, each system can show users various results or reports, which could let people do self-reflection based on their own purposes and expectations (Elsden and Kirk 2016). Fitbit is a widely used wearable device for people to track their daily physical activities. Fitbit has many kinds of wearable devices, like wristbands and watches. This kind of devices uses chips, algorithms, and technologies to monitor users various of activities in daily life. Fitbit devices could record users' sporting activities, walking distance, calorie burning, heart beating, and sleep quality. The report of self-monitoring activities can be later reviewed on both devices and application. Another widely used self-tracking device is GoPro, which is a wearable sports camera (Figure 4). This wearable camera allows people to record their daily life in video format, like recording Vlogs, logging memorable moments (GoPro). Self-tracking hardware devices perform outstandingly in recording physical activities. Lens and sensors are implemented in these hardware devices to pursue accurate recording quality (Quantified Self Homepage). But wearable devices have shortcomings. Firstly, these devices are not good at recording virtual information, like computer screen activities. Secondly, one wearable device cannot be used to record various personal information. For example, monitoring body information, like heart rate, is impossible for portable cameras.



**Figure 4: Wearable sports camera: GoPro. Users can self-track their physical activities and create a time-lapse video with it.**

There are lots of applications designed for self-tracking. Apple Health as a mobile application, can track users' daily activities and monitor user's real-time data. This software provides users a health record to assist them in achieving their health goals. In addition, RescueTime is a self-tracking software (Figure 5) installed on the computer monitoring computer activities, such as which website you have browsed, which application you have launched, and how much time you spent on each website and application (RescueTime). Also, this software can classify different websites and application into many categories, like social networking, business, entertainments, and productivity. Based on the collected data, RescueTime can show users different visualized graphs, like bar chart and pie chart, which can indicate users how they spend their past time on each web page and software. Individuals can adjust their future plans and schedules based on this result. However, people cannot know exactly what they have done in the past. People do not know which websites they have visited or which files they have edited. RescueTime cannot help users go back to previous working contexts with collected personal data. Except for the above examples, more and more software and applications for people to self-monitor themselves are designed and developed. In different domains and areas, people can find the right products which meet their personal needs.



**Figure 5: Self-tracking software: RescueTime. This software can track users’ computer activities on computers and provides summary reports.**

Self-tracking is not a simple task that just records people’s activities. We can use self-tracking technology and collected personal information to achieve various personal goals, e.g., controlling weight, retrieving unfinished works, and planning future schedules. According to previous work, researchers found that even a simple form of self-monitoring feedback constitutes self-reflection and contributes to behavior change (Choe et al., 2015). This result means that users can easily change their behaviors and form new habits based on the result of self-tracking. In my thesis, I implemented self-tracking technology to record users’ computers activities in real-time. According to collected personal information, ScreenTrack can help users understand their past computer activities and retrieve previous working context in a short time.

## 2.2 Self-Tracking to Facilitate Self-reflection

After the process of collection and integration, the next phase of the personal informatics model (five-stage model) is the reflection (Li et al., 2010). How to use collected personal data efficiently and effectively is important for this phase: reflection. Personal data can be used for self-reflection to help people become more aware of their own behaviors (Carver and Scheier, 2001), make better decisions (Endsley 1997), and change own behavior in different domain and areas (DiClemente et al., 2001).

Motivated by the Transtheoretical Model of Behavior Change (TTM) (Prochaska and Velicer, 1997), Li et al. later identify two phases of reflection, 1) discovery, and 2) maintenance. In this paper, researchers noted that people should ask themselves different types of questions in each phase (Li et al., 2011). For maintenance phase, people should ask questions about status and discrepancy. In the discovery phase, people should ask questions about histories, goals, context,

and factors. The difference between these two phases is whether participants know their own goals for self-reflection or not. Some people start self-tracking practice without specific clear aims. This fact decreases the efficiency of information gathering until people know exactly what they want, like reasonable goals. After setting personal goals, people would transfer to the maintenance phase from the discovery phase. For people who already have own purposes, they would stay in the maintenance phase until they discover any other new aims.

Individuals can benefit lots from the processing of self-reflection. On the academic side, computer science researches in the human-computer interaction (HCI) field have already done many kinds of works to study the advantages of self-reflection (The Center for the Healing Arts & Sciences, 2015). Following are the benefits of the self-reflection:

- 1) Self-reflection can help individuals build emotional self-awareness. By taking additional time to look back about yourself, users can gain a better understanding of their emotions, strengths, weakness, and behaviors.
- 2) Self-reflection can help users detect and change their behaviors. According to a previous study, the result shows that even a simple form of self-tracking feedback constitutes self-reflection and later contributes to behavior change (Choe et al., 2015).
- 3) Integrity increase when people take time to reflect on core values. People need to have a clear understanding of which values are most important for them. Putting these values at the forefront of minds is essential. Taking time to think allows for new ideas to emerge and counters the risks of becoming stagnant.

Our lives often become a cycle; people are always acting and moving from one thing to the next. Hence, they often forget to stop and look back. Self-reflection provides users with a process to dive deeply into their past personal data. This procedure allows people to know themselves better than before.

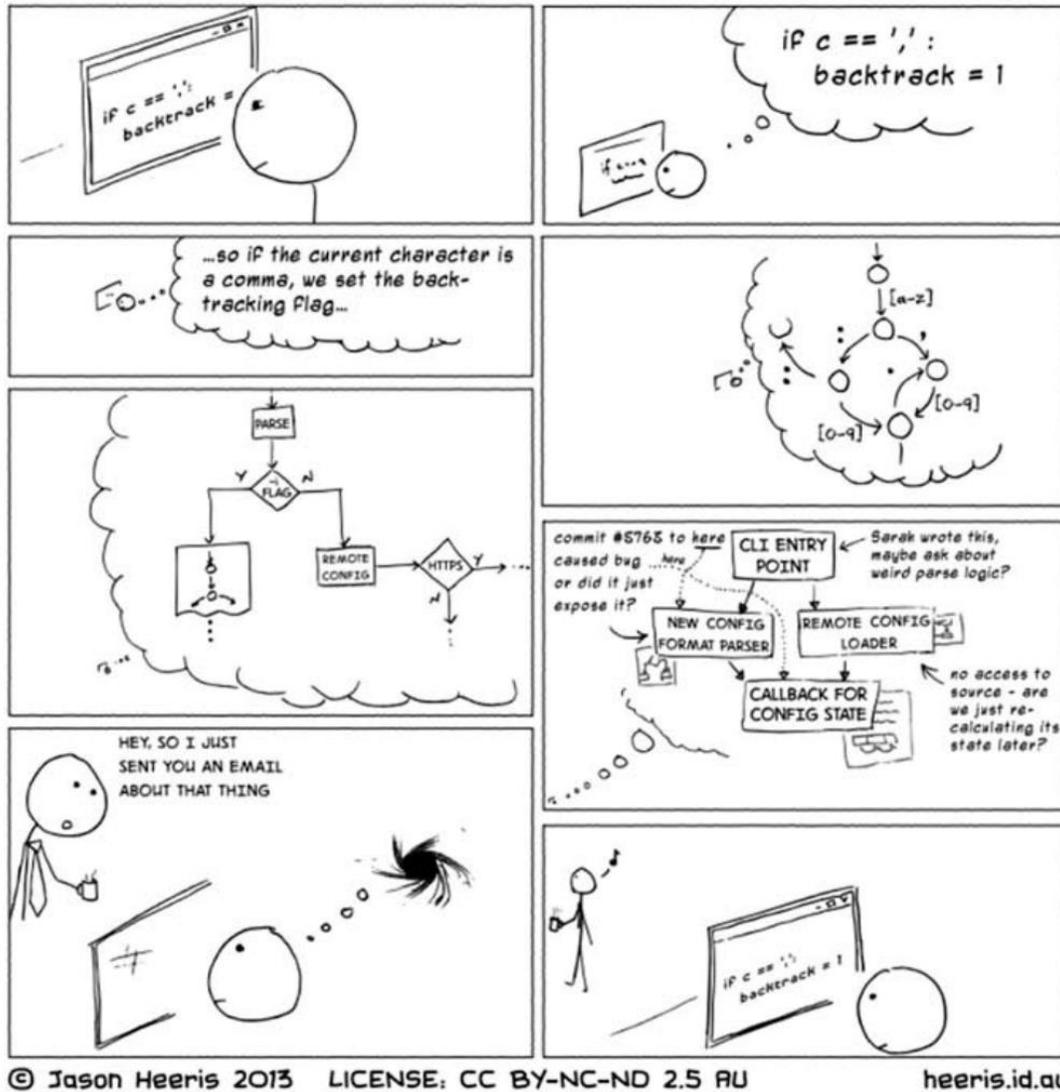
### **2.3 Mental Context Reconstruction for Task Resumption and Reflection**

People need to continue their previous interrupted, suspended or unfinished activities by recovering the physical or digital resources. Mental context reconstruction is a process that people re-encodes the information and task environment which necessary to perform the previous tasks (Salvucci, 2010). However, restoring a previous mental context could be difficult. For this problem, we currently have various methods to help people reconstruct their precious mental context in daily life. For example, some people prefer to use sticky notes to write down messages and cues (Parnin and DeLine, 2010), which could help individuals go back to their unfinished or suspended past work quickly.

However, there are still some difficulties for mental context reconstruction. Taking notes is not an efficient method for computer activities when there are lots of tasks across different software and documents. Writing down all the information seems impossible. In addition, computer activities always contain creating, changing, and manipulating complex mental states that fade quickly after inevitable interruptions and distractions. Due to these reasons, many studies have

already focused on this notorious topic to enhance the effectiveness of productivity for computer activities. In previous research, Chris Parnin's et al. found that programmers routinely use both suspension strategies (Parnin and DeLine 2010) (e.g., taking hand-writing notes and making cues) and resumption strategies (e.g., seeking cues and reviewing systematic information) that involve a wide range of information sources (e.g., program files, comments, notes, compile results, browsed websites and to do list) to preserve mental context after interruptions (Parnin and Rugaber 2011). For programmers, people leave comments and writing notes between code lines to help them resume previous working context after interruptions. In addition, checking compile results can also help them locate the place that they are looking for quickly. In Figure 6, it shows that reconstructing working context is essential but difficult. In another similar study, Kersten et al. found that programmers would like to use the software's unique built-in features to reconstruct past mental context (Kersten et al., 2006). For example, the file tree feature in software Eclipse. Such methods are valuable when people want to resume their previous working context, especially when the number of tasks is increasing.

## THIS IS WHY YOU SHOULDN'T INTERRUPT A PROGRAMMER



**Figure 6: Do not interrupt a programmer (Heeris, 2013). This cartoon well-explain why mental context reconstruction is needed for resuming task, which is challenging.**

Researchers are finding more methods and designing more tools to help individuals reconstruct past contexts. As previous researches pointed out, resuming suspended activities is fundamental to knowledge work. As activities grow in complexity or fade from human beings' memory, these activities become increasingly difficult to cognitively resume (Iqbal et al., 2007). Researchers have studied many methods to help users improve productivity in mental context reconstruction, e.g., programming. Texts, images, and videos are several main ideas aiming for efficiently mental content recovery (Laming and Newman, 1992; Mayer and Bederson 2003; Rule et al., 2017). One study also pointed out that text can be richer in cues than images, which is in contradiction to conventional wisdom (Levy et al., 2016). Using metadata is a common method of reconstructing. Texts contain more details in limited space. However, searching for useful messages in these texts can sometimes be troublesome and time-consuming (Rule et al., 2017).

Instead of videos, images are chosen by most researchers used in mental content recovery studies. With the action of zooming in and out, editing size and color, users could gain more useful information about what they have done in the past. What is more, the software, files, websites which are recorded in visual history, can also help user shorten the time spend on recalling. Static information about past computer activities is not the only data that researchers are looking at. Also, in the study of lifelogging technology, combining location information with recorded personal data together can actually allow people to better situate past activities in context (Kalnikaite et al., 2010). Hence, mental context reconstruction is important for people to resume previous tasks.

## **2.4 Interruptions in Working and Studying**

Interruption is one barrier for people to finish their work in one time. The interruption concept has been defined many times from past to now. Interruptions were studied as early as the 1920s when Zeigarnik stopped people during some simple tasks and allowed them to complete others (Zeigarnik, 1938). Later, more and more studies have focused on the topic of interruption. Coraggio defined interruption as “an externally-generated, randomly occurring, discrete event that breaks the continuity of cognitive focus on a primary task” (Coraggio 1990). McFarlane provided a comprehensive review of the conceptualization of interruption and claimed it was usually defined as a change or disturbance in the process of people’s activities (McFarlane 2002). More specifically for HCI in computer science, McFarlane defined interruption as “the process of coordinating task switching in the human activity of multitasking”. Similarly, Mark et al defined work fragmentation as a break in continuous work activity (Mark et al., 2005).

Interruption can be classified into two main categories, external and internal (Miyata and Norman, 1986; Mark et al., 2005). External interruption drives situational changes in the workplaces, e.g., a phone ringing, a notification jumped on a computer screen, and a co-worker’s disturbing. Internal interruption is a personal initiative to alter the work environment, e.g., taking a break, distraction, and thinking about other matters. Interruptions come at a cost and may result in negative consequences. Dabbish et al. found that self-interruption occurs more frequently when people who work in open office environments where they are located in a central work area (Dabbish et al., 2011). Both external interruption and internal interruptions may affect whether and how fast a previous working context is resumed. Gonzalez and Mark’s research on the degree of fragmentation of work revealed that people are as likely to disrupt themselves as they are to be disrupted by external sources (González 2004).

Nobody likes interruptions. But unfortunately, interruptions are inevitable and anywhere in our daily life. When people are working or studying, interruptions can happen at any time anywhere, for example, distractions, unexpectable emails, taking a break, and seeking help from other co-workers. Interruptions are disruptive to the completion of routine tasks in two ways. The first one is taking extra time to recover from an interruption, in terms of the resumption lag needed to re-engage with the primary task. Second, interruptions increase the likelihood of errors being made on the primary task. Because people may either repeat previously finished work or miss essential information from previous tasks (Brumby et al., 2013).

Dario categorized the effects of interruption into two categories, best timing and worst timing (Salvucci 2010). In this study, the researcher determined the “best” and “worst” in terms of whether the interruption occurred between higher-level subtasks (best) or within lower-level subtasks (worst). For the worst interruption timing, people need to spend much more time and extra actions to reconstruct their previous tasks than the best interruption timing. Individuals have to resume more details and information about their previous suspended work when interruptions happen in worst timing. It is clear that nobody can control the timing of interruptions in daily life. Resuming a previous working context from interruptions is a challenge for individuals.

According to all the above previous studies, we can find that interruptions are inevitable and burdensome in our daily life. We can hardly do anything to avoid interruptions. The cost of resuming from interruptions is not less, as time and energy. In addition, we cannot expect that every interruption is the “best interruption” occurred between two finished tasks. Hence, for a short time term interruption, we need to try our best to go back to our previous work as quickly as possible in case of forgetting what I have done in the past. For longtime term interruption, relying only upon our memory and brain is not a wise strategy. Individuals need to depend on external help, like sticky notes, images, cue information, and related software. One of my goals of the software is focusing on the influence after interruptions and helping individuals resume their previous working contexts in a short time after interruptions.

## **2.5 Utilizing Visual Histories for Information Retrieval and Mental Reconstruction**

One promising approach to resuming mental context lies in cuing users their own memories with images and visual information from their past. Humans have remarkably discriminative visual memory (Rule et al., 2017). In one previous research, participants were shown a series of 2560 images. Each image was shown for about 10 seconds on average. Three tested days after initial presentation, participants were still able to select which of two images they had seen before with 90% correct rate (Standing et al., 1970). This study has shown that human beings can remember the information contained in images correctly for a few days.

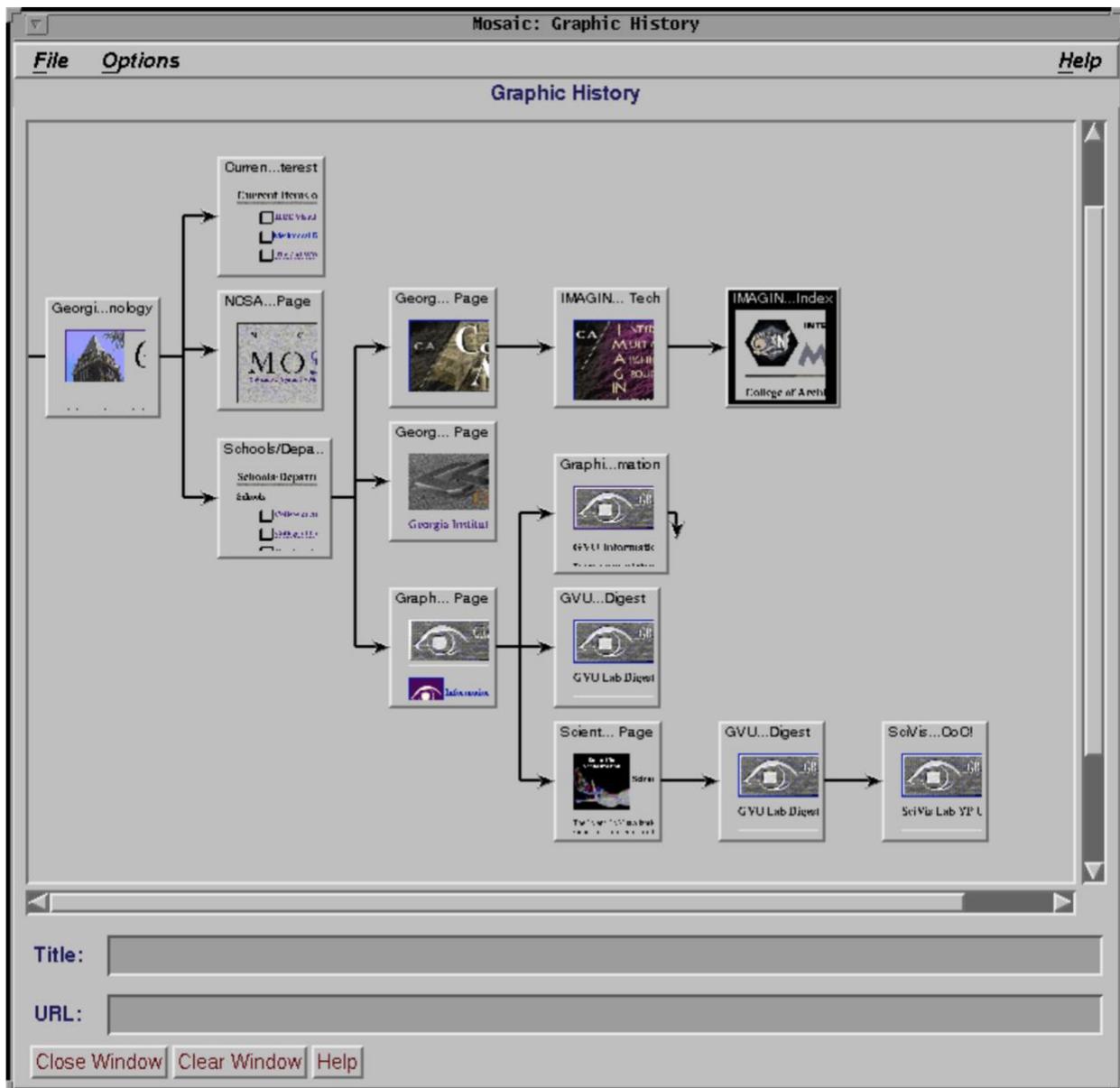
Moreover, a similar study found that people were able to accurately identify the images they had seen before 87% of the time. Even though the only difference between displayed images was the pose of an object (Brady et al., 2008). For human beings, visual long-term memory has a massive storage capacity. People can remember much details of objects contained in visual history. In addition, this finding suggests that people might be able to discriminate similar images which recorded their past activities.

Researchers have found that combining multiple information formats together can help people recall previous contexts. Ding et al. found that video including both text and imagery are more effective than either modality alone (Ding et al., 1999). Researchers stated that the combination of text and visual methods together can significantly improve both recall and comprehension, which would be better than using solely text or images (Michael et al., 2002). Text phrases coupled with imagery produced a useful cue for individuals about their previous contexts. Christel et al. designed and developed a software which combined text, images, and videos

together into one user interface. These additional messages allowed users to browse information directly and to reveal further details at the same time (Christel et al., 2002). In addition, the presentation of previous suspended events can efficiently remind and update users' memory about suspended or unfinished tasks (Czerwinski and Horvitz, 2002). Another research found that using visual histories (thumbnails of the computer screen) can help users reconstruct mental context (Rule et al., 2017). Desktop thumbnails work better for rich mental context reconstruction than text cues. Using visual history can actually improve the process of recalling and can cue users with more details about their past activities and contexts.

Not only studies as mentioned above but using image information to cue previous context memory is also supported by applied research. Researchers argued that considering the fact of autobiographical memory could help develop methods and tools assist people in restoring previous mental context and past activities (Lamming and Newman, 1992). In this research, researchers also found that people could remember more about their daily life while reviewing time-lapse videos than simply relying on text summaries or based on their own memory. In particular, researcher built a hardware device that can write biographies with icons and text messages to help with everyday memory problems (Lamming and Flynn, 1994). This method provided an alternative way of retrieving information that was once known but has been forgotten later. In addition, using abstract textual messages is less efficiently help people reconstruct previous mental context than using images and time-lapse videos. To be more specific, later studies focused on corroborating this finding with videos which recorded people's activities in their working environment (Czerwinski and Horvitz, 2002). A study used images that were taken with various wearable devices and software showed that visual history support people to recall past activities (Sellen et al., 2007). In the field of coding and programming, researchers have found that showing snippets of recently edited code and comments could help programmers resume their past tasks quickly and even a more accurate identification of previous working context (Safer and Murphy, 2007). These studies demonstrated that individuals can recognize their past activities in images correctly and quickly, even for long-term memory.

Researchers have done lots of studies in visual history helping people recall previous contexts while surfing the Internet. Based on the history of web browsing, many researches implemented the technology of visual history to help individuals recall past surfed web pages. Researchers enhanced the history-keeping facility of the browser by providing a two-dimensional view of the web pages that a user has visited during a past session (Ayers and Stasko, 1995). Figure 7 is an overview of the graphic history view in browser. Visited websites have been stored in a tree data structure with images and metadata to provide cues for users when they want to revisit these pages again. It is hoped that the addition of the graphic history view can encourage exploratory behavior and help users navigate past web pages easily in general. Mayer and Bederson developed Browser Icon to help users visualized their past web history. This software can dynamically draw animated graphs of the user's paths through the previous web pages they have browsed (Mayer and Bederson, 2003). Not only storing metadata like URLs and web titles in browsers but visual information at the same time help individuals understand their past surfing activities and revisit previous closed web pages quickly.



**Figure 7: Graphic History View of Web Browsing (Ayers and Stasko, 1995). This history view demonstrates how a user navigate web pages over time and what web page they visited from which webpage.**

These above studies and researches have shown that people can recognize their previous working context quickly and correctly under the help of visual history, like image recording and time-lapse video. These images and video information can assist users in identifying their previous working tasks and restoring past activities accurately. Individuals are able to remember more details about past events when viewing images of past events compared with textual summaries and their own memory (Hodges et al., 2006; Lee and Dey, 2008). In ScreenTrack, I am going to implement visual history feature which combines text, images, and time-lapse video together helping users recall their past working contexts and computer activities.

## 2.6 Conclusion and Thoughts

After reading these related papers and studying existed systems, this chapter has provided literature reviews about self-tracking (2.1), self-reflection (2.2), interruption in working and studying (2.3), mental context reconstruction (2.4), and visual history (2.5). This background knowledge is the fundamentals of designing and implementing in the future. Only based on this solid information and knowledge, can I continue my next step of designing the system.

According to the above background knowledge, the process of self-tracking is the primary phase of ScreenTrack. Self-tracking is not only about recording personal information, but also about using this collected data. Based on personal informatics model, the first step I need to do is collecting personal data and information, which is computer screen activities. This is the fundamental knowledge of the design of ScreenTrack. Reconstructing previous contexts is a problem for computer users. Due to interruptions, recalling previous working context is a question for many people. Only relying on memory is not a good method, especially when there is lots of information need to be remembered. Images can cue people much more details about past works. Motivated by the research which focused on visual history, I plan to implement visual history feature in ScreenTrack to help people understand their precious tasks and recall interrupted or unfinished tasks in a short period of time.

Additionally, I found that even though there are already many types of research focused on self-tracking, interruption resumption, and visual history. Most of them separately studied these areas without building valid connections between each part. Or studies mainly focused on some specific fields. For example, researchers mainly studied self-tracking in health topic, and mental context reconstruction is mainly discussed for programming situation. There is no research studied on how to help people reconstruct previous working contexts in general conditions using self-tracking method. According to these facts, ScreenTrack should build connections among self-tracking, visual history, and working context reconstruction. Based on this goal, this software should be designed according to the technology of self-tracking. This software should provide the user with the function of mental context reconstruction helping user recall their past works in a short period of time using visual history feature. I discuss design goals of ScreenTrack in the next chapter in further detail.

## Chapter 3: Design Goals

In this chapter, I state design goals for ScreenTrack. The design of this system was informed by prior studies on the methods of self-tracking, a set of benefits grounded in recall from interruption, and reducing time to recover previous task context. All these design goals are aiming to enhance productivity based on the method of self-tracking and mental context reconstruction.

### 3.1 Create a Replayable Visual History of Computer Screens that Works Across Programs.

The computer is effective and efficient at archiving activities history. But each software has its own history. For example, Google Chrome can record the history of web pages that users have browsed in recent 90 days. Microsoft Word can show the list of recently opened files and documents. However, the history of each software is independent, not interleaved with usage history from other applications. Users cannot have one single history of all the computer activities of using different software and applications. While a user can learn a history of opened documents or web pages within an application, users cannot find information unless they know which application they used for a particular activity - for example, a pdf document can be opened in a web browser or a pdf viewer application. In addition, working context in a computer can be composed of multiple applications, not single one (e.g., for writing an academic paper - word processor app, dictionary, bibliography software, and web browsers). These situations increase the complexity of retrieving relevant working context and the need for maintaining one integrated history of computer activities.

In addition, existing methods of recording history stored in different software are primarily textual information. People can get information only from names, URLs, and titles. Sometimes it is challenging to tell whether the file or URL is the one that they would like to retrieve just by reading the title or the file name of it. Especially, in case that the file names are similar or not informative, users are most likely to determine if the document is the one that they are looking for by opening it individually. For the browser, some web pages have vague titles, e.g. research papers and untitled pages. Some titles are the combination of meaningless numbers and words. Based on prior studies, using the visual history of past activities could help users recall previous working context quickly (Ayers and Stasko, 1995; Michael et al., 2002; Rule et al., 2017). According to these studies, images of past events can assist people in remembering additional details.

Hence, creating a visual history of computer screens that work across programs can be a solution for users to enhance productivity and improve their effectiveness. ScreenTrack can create a visual history of computer screen activities across different applications. Using screenshots and textual metadata in a time-lapse video together can help users see their previous working context in chronological order.

### 3.2 Capture a Screenshot with Metadata Available and Display the Visual History with Metadata.

Based on a prior study, displaying textual information with images together can significantly improve both recalling past working context and comprehending details of previous tasks (Large et al., 1995). Text can directly show users metadata involved Images can display rich information about past working content. During the process of self-tracking, only storing one kind of information is limited in cannot helping users remember their past working contexts and retrieve previous messages either. Hence, showing both text and images to users requires ScreenTrack to record this information at the same time. Due to the above reasons, I designed ScreenTrack to capture screenshots with metadata available at the same time.

One of the design goals for ScreenTrack is saving metadata available while capturing screenshots in order to help retrieve working context in the future. When a screen activity is detected, the application captures not only computer screen image, but also other information including the frontmost running software name and taken time. Then, based on the name of this frontmost running software, ScreenTrack classifies it into different categories which I defined in the software. There are four categories: Productivity, Browser, Coding, and Others. In addition, for certain category of software, it stores additional information including file title, file path, URL, website title, software window size and position at the same time. After collecting all these available metadata, ScreenTrack store this information into a JSON file into a local folder that corresponds to a session. Capturing as much as available data is essential for showing users rich details about previous working context.

### **3.3 Display Metadata and Screenshots Together.**

Implementing visual history function in ScreenTrack is a primary way to help people retrieve their past activities from a computer screen image. From a prior study, researchers found that the combination of text and visuals in the software would perform than using solely text or images (Ding and Soergel, 1999). This study showed the presentation of text message with pictures could significantly improve both recall and comprehension compared to either picture or text message alone (Large et al., 1995). Individuals could reveal additional details through images which were recorded from past activities. Based on the above findings, one of the design goals of ScreenTrack is displaying screenshots and related metadata together in the user interface. Combination of textual information and screenshots together can help users retrieve their previous working context in a short time and hence improve productivity.

In ScreenTrack, screenshots are displaying with the corresponding metadata. Metadata that was stored while taking screenshots are shown on the right next to the screen captures. The metadata includes related information of displaying images, such as software name, taken time, and web page path. This arrangement is similar to other online video players which most people are familiar with, like YouTube player. While using the software to review their past activities, people can browse the images continuously on the left and can also check the related metadata information on the right at the same time.

### **3.4 Provide Various Ways to Capture and Navigate Temporal History of a Computer Screen.**

One of the design goals for this system is providing various options to take screenshots and navigate the visual history of computer screen activities. This design consideration can help users self-track their computer activities and retrieve past information tasks conveniently and quickly in a customizable way that they want.

ScreenTrack provides users with various ways to capture computer screen activities. Taking screenshots with a fixed time interval cannot capture all meaningful incidents while users are tracking their computer activities. It may miss essential information due to the interval. In the meantime, capturing all possible snapshots are not a feasible solution. To solve this problem, I added an option to capture the screen whenever an application changes. It is a checkbox labeled “Detect switching software” in the menu bar user interface. Before starting to record computer screen activities, users can select this option. During the process of recording, ScreenTrack can capture a screenshot whenever the frontmost running application is switched. According to a previous study, people switch software frequently while using computers (Jin and Dabbish, 2009).

ScreenTrack allows a user to navigate their past visual history of computer activities on various levels. This system provides four different ways for users to navigate their visual histories over time: 1) Replay Time-lapse video, Timeline Slider, Buttons, Playing speed. Users can use different methods based on their own purpose and how delicately they want to locate a screenshot. For example, a user can simply replay the video to see what they did in a time window that they are interested in to get a temporal summary of what they have done. On the other hand, users can drag the slider to help them locate a specific screenshot. Clicking previous and next buttons can help users review screenshots one by one. Users can also go to the latest image and the oldest image in the selected time window with only one click. Lastly, users can change the playing speed at any time while they are watching the time-lapse video. Providing such different methods facilitate flexible navigation of visual history.

### **3.5 Make the Disk-space Requirement Manageable and Address Privacy Concerns.**

Disk-space requirement and privacy problem are two other considerations in designing ScreenTrack. These are two important problems for every people. Privacy problem in this thesis is about protecting users’ personal information which is stored during the process of self-tracking. The disk-space requirement is about considering the storage of disk which is used to save screen captures and metadata in local disk.

Firstly, ScreenTrack takes screen captures automatically during the recording process and then saves them into local disk. In this process, we gave a user a few options to use the disk-space under certain limits (image compression, automatic deletion of old files) and securely stored files for privacy (no connectivity, password protected folders). Storing these screenshots can take much disk space. More computer memory is needed, especially when a user runs this system more than 8 hours every day and leave it always-on, or set a short time interval for taking screenshots (e.g., 5 seconds). To solve these above problems, designing the goal of making the disk-space requirement manageable is set. I added a slider named “Compression Rate” in the menu bar interface. The user can drag this slider to choose the compression rate of the

screenshot. Instead of saving the original size of screenshots, the system can compress the screenshots into a smaller size and take less memory of disk while saving them. In addition, the format of the screenshot is “.jpg” which takes less space than other image formats. Besides, an additional feature is planned to delete session data that is older than a week automatically. ScreenTrack is primarily for a short time window (up to several days).

Secondly, I considered the privacy problem about how to protect users’ personal information. Privacy consideration is one of the most critical concerns for every people. No one wants their personal information to be leaked to others, especially their computer screen activities. This kind of information may contain many details of private information (e.g. credit card information). For the design goal of protecting privacy, I choose not to include any networking function in ScreenTrack and keep any stored information in a local disk as opposed to posting to a remote server. Users do not need to connect their computers with networks to run this system. All the information and data can be saved locally without uploading to the network. In this case, only the user who is able to access this computer can check his self-tracked personal information and data. In addition, in ScreenTrack, a checkbox named “Play sound when taking screenshot” is designed in the menu bar interface. A user can be alarmed with a sound whenever a screenshot is captured. This feature will help a user to be aware of what kinds of information are captured so that they can later delete files if they do not like them. I also plan to design a visual indicator (with countdown) to make it less intrusive than sound alarming.

## Chapter 4: Software Design

This chapter includes descriptions about the designing and development of ScreenTrack. I have designed and implemented the ScreenTrack software on Apple's Mac operating system: Mac OS. Among many operating systems and programming languages, I selected Mac OS and Swift (a programming language) for software development. This development environment is superior and stable. Mac OS is one of the widely used operating systems and ScreenTrack developed in one system is enough to answer the two research questions (RQ1 and RQ2) that I plan to answer from the thesis. Swift is a new programming language which is being enhanced and encouraged to be used for more and more developers. Hence, using Swift as the programming language makes the software be maintained and improved in the future easily.

My design considerations have been discussed in chapter 4. With those in mind, I started to design user interfaces of the system according to the knowledge of HCI and existing software that users are already familiar with (e.g. video player). ScreenTrack is designed to be able to run all the time and record screen activities while people are using their computers. For most cases, this recording process may keep continuously running for several hours. Hence, this system should not occupy too much memory and CPU. What is more, when my software is running and recording, it should not interrupt users at any time. In order to accomplish these ideas, I decided to design this system as a menu bar software. This kind of software only appears on the top of the screen in the menu bar (In Figure 8, the top bar in gray is the menu bar.) For the menu bar user interface, I tried to simplify options and features in this user interface. Users do not want to be boarded by too many options and selections at the beginning of the self-tracking.

Considering the software main feature, visual history, I placed a video player interface on the left while metadata information is shown on the right. The combination of text, images, and video together can help users recall their previous work quickly than only one kind of message. Based on the time-lapse video visualization, users can figure out their past computer activities and retrieve previous working contexts in a short time.

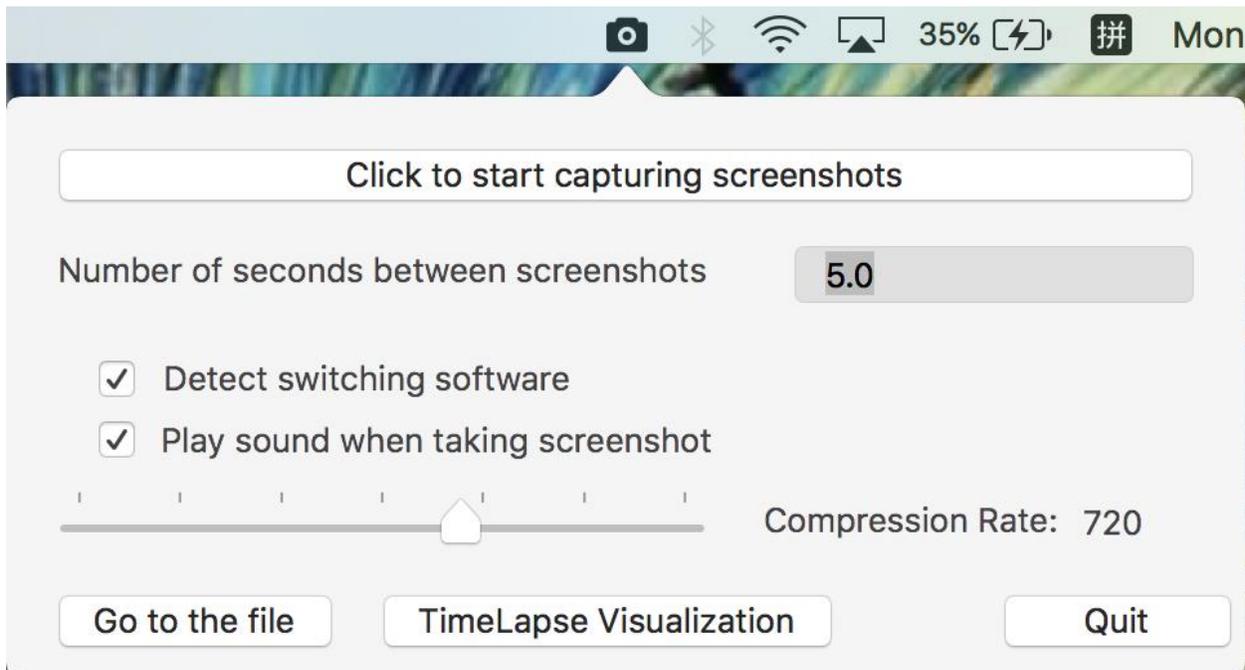
Additionally, a function that can help users reopen those past files, documents, websites, and software is implemented. Instead of simply telling users what they have done in the past, ScreenTrack provides users with this method to go back to these previous files and software. Only offering static information and screenshots cannot help people retrieving previous working contexts without actually reopening them.

### 4.1 Recording Configuration User Interface

As stated in the chapter overview and design goals above, ScreenTrack is supposed to be running for consecutive hours per day while a user is working on his computer. Hence, this software should not distract users' concentration and interrupt users at any time. For the Mac OS, if some software is opened or running, their icons are displayed on the dock. If some software is minimized, thumbnails of this software are shown on the dock too. Icons and thumbnails can be one of the sources of distraction and interruption. Items on the dock can increase the difficulty for users to find software they are looking for. This situation can increase the possibility of interruption and decrease the working productivity. According to these reasons, after observing

and discussions, I decided to design ScreenTrack as a menu bar software. While running this software, only a small icon is displayed on the menu bar at the top of the screen, which occupied less area (Figure 8).

For the design of the menu bar user interface, I tried to simplify options in this user interface. I place as less as checkboxes, sliders, and buttons on this user interface. Menu bar user interface is the first phase of ScreenTrack, Users do not want to be boarded by too many options and selections at the beginning of the journey of self-tracking and mental context reconstruction. Based on this thought, I provide only seven functions in this user interface: the recording button, time interval setting, sound display checkbox, detecting software switching checkbox, compression ratio slider, time-lapse visualization button, go to folder button and quit button.



**Figure 8: Recording Configuration User Interface. A user can set default values and start self-tracking to record computer screen activities.**

The uppermost button is the recording button. Users can click this button to start recording screen activities and taking screenshots. Also, users can click this button again to stop the recording process. After each click, the title on this button changes to “Recording! Click to stop automatic screenshot”. Messages shown on this button inform users whether it is recording currently or not. In this user interface, users can input their own preferred time interval to take screenshots. Once the time interval is set, this value cannot be changed when users start recording the screen. People are prevented from changing the time interval casually during the process of recording.

The checkbox of detecting switching software is the function that can take a screenshot automatically when the software detects that the frontmost running software has changed. For most cases, fixing the time interval for taking screenshots is not flexible. People often switch current software to another one for additional information. In this case, due to a longtime

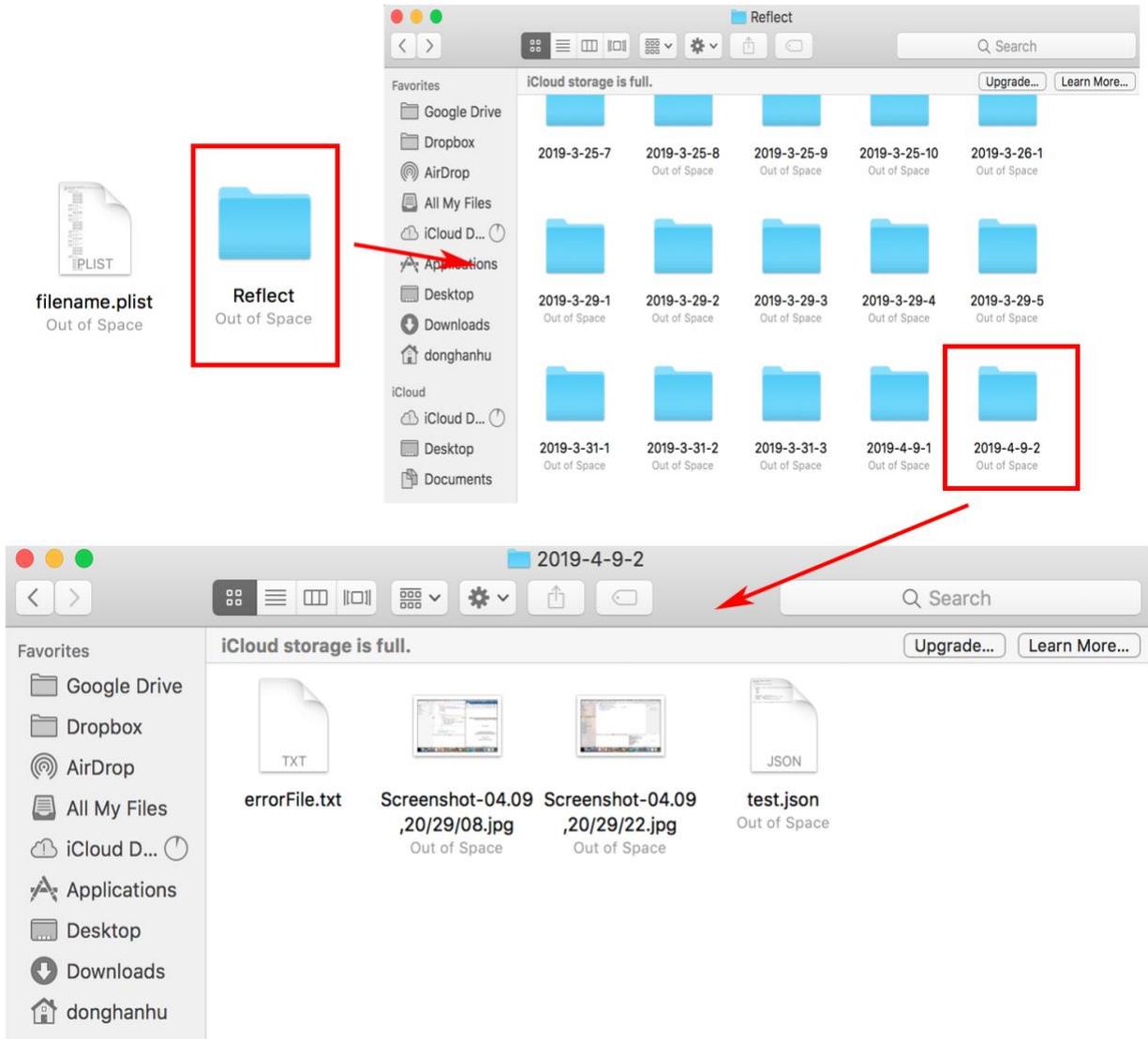
interval, ScreenTrack may omit some essential screen activities. The second checkbox is the “Playing sound” option. If users select this checkbox before clicking the recording button, ScreenTrack plays a capture sound while taking a screenshot. This function is designed to alert users that ScreenTrack is recording the screen activities currently.

The following slider is the compression ratio slider. ScreenTrack is supposed to take screenshots for continuous working hours. Hence, the recording process can last several hours per day. Saving all these original screen capture images definitely occupy lots of memory for users’ computers. By moving this slider, ScreenTrack can automatically compress screenshots into a specific size, which is less than the original size. After this compression procedure, these operated screen captures take less memory for local disks. Hence, users do not have to worry about using several gigabytes memory to save these pictures.

The “Go to file” button is a function helping users go the least folder which stores screenshots for the previous session. The “Time-Lapse Visualization” button is designed to open a new window showing the time-lapse video for displaying visual history. Finally, the “Quit” button is designed to close the whole software.

## **4.2 ScreenTrack Captures and Stores Screenshots in a Local Folder.**

ScreenTrack is designed and implemented for users to self-tracking and mental context reconstruction. Based on these aims, this system can help people enhance productivity and improve the effectiveness of using computers. Focused on these basic goals, the first task for ScreenTrack is self-tracking. To be more specific, in my research, self-tracking is a procedure that can record computer screen activities as much as clearly and then save recorded data locally for future usage. In ScreenTrack, the main self-tracking method is taking screenshots which contains screen activities.



**Figure 9: Screenshots folder hierarchy. These folders show how ScreenTrack saves screenshots and metadata into local disk.**

To achieve the goal of screen capturing, I used the OS built-in screen capture function. After launching ScreenTrack, this software detects automatically whether the default folder has already existed in the disk or not. This default folder is used to save screenshots. If not, ScreenTrack can create a default folder automatically in the disk (Figure 9). The path of this folder is hardcoded for Mac OS, and the path for this default folder is “/Users/xxxxx/Documents/Reflect”. “xxxxx” in this folder path is the username of this computer. This default folder is only used to save all screenshot session folders. After creating this default folder and clicking the start recording button, ScreenTrack is able to take a screenshot for the whole computer screen. Then, after the operation of compression, this screenshot is stored into the corresponding session folder automatically.

Additionally, for most cases, it is impossible that users keep recording their computer screens for several hours without pausing. People need time to have a break, eat food, or even shut down computers. For privacy consideration, people like to stop monitoring when they are typing personal information. Due to the above different reasons, users always divide the whole recording process into several parts. For each of these separate recording sessions, ScreenTrack creates a local folder and saves these images distinguishingly with previous folders. Users can also later check these folders for screenshots manually.

### **4.3 Frontmost Running Software Identification and Data Collection**

Only taking screenshots for a current computer screen is far from enough. There are still several drawbacks existed in this current monitoring method. Firstly, based on only screenshots, users still need extra time to check out what kind of information has been recorded in this picture. For example, users require to review images carefully to figure out the software that has been running in the past time. What is more, users may also need additional time to figure out files, documents, projects, and websites recorded in this screenshot. Secondly, after recognizing the software and the contained information, users still require to spend extra time finding these specific files and websites. The process of finding past working contexts can be complicated and burdensome. Spending time on restoring this information is a waste. According to this, one of my fundamental goals is to shorten this procedure as much as possible to enhance productivity and to improve working effectiveness.

Therefore, I added a function to identify the frontmost running software while taking the screenshot at the same time. Additionally, after identifying the frontmost running software, ScreenTrack then classifies this frontmost running software into different categories. Based on different categories, ScreenTrack collects related information for this software at the same time. For each launched software, collected data is different from other software. This relevant information is stored in a JSON file, which is created automatically at the beginning of the recording in the session folder.

However, it is impossible to classify every software clearly into the corresponding category. There are thousands of different software for Mac OS. In addition, each user has his own preference for choosing different software. Users like to install various software for personal purposes. Some of this software is not widely used for other people. Hence, according to the above reasons, I only classified software that Mac user used most into four main categories: 1) Productivity, 2) Browser, 3) Programming, and 4) Others. Applications that belong to Productivity category include “Previews”, “Pages”, “Numbers”, “Keynote”, “Microsoft Word”, “Microsoft Excel”, “Microsoft PowerPoint”, “Acrobat Reader” and “TextEdit”. For the Browser category, there are “Safari” and “Google Chrome”. For the Programming category, there are “Xcode” and “Eclipse”. The fourth category, “Other” is for the software except for the software as mentioned above (Table 1). These categories are not comprehensive by any means and should be refined as more applications should be covered. In addition, people may want to get different kinds of metadata per software so the category information should be customizable ideally. This remains as a future work.

	Software Name	Recorded Time	Category	File Path/ Website URL	File Name/ Website Title
Productivity	✓	✓	✓	✓ (File Path)	✓ (File Name)
Coding	✓	✓	✓	✓ (File Path)	✓ (File Name)
Browser	✓	✓	✓	✓ (Website URL)	✓ (Website Title)
Other	✓	✓	✓	X	X

**Table 1: Related information collected based on software classification.**

As shown in Table 1, ScreenTrack collects different kinds of information based on different categories. As for the “Other” category, ScreenTrack only identifies its localized software name instead of collecting related information. All rest information is set as “Null” in the JSON file. For some social networking software, no additional information can be stored to help people retrieve their previous working contexts. In the future, more software for each category will be added into the software, and the number of the category will increase.

The reason that I create JSON files to store data and information is that storing various data into a JSON file is easy with an organizational hierarchy. ScreenTrack can write data into the JSON file and then fetch it out easily without worrying about crushing. In addition, JSON files occupy less memory for a local disk. For the design of a JSON file, ScreenTrack saves several following essential data into it. 1) The folder path of this session folder. ScreenTrack fetches information and screenshots based on the path of this session folder. 2) Start time and end time of this recording session. This time information is used to figure out the time of recording. 3) The number of screenshots saved in this session folder. 4) A dictionary array to store related information of each screenshot. Each dictionary data structure contains photo name, software name, the window size of this software, and the category for this corresponding software (Figure 10, 11, 12, 13). All this data and information is used later in the time-lapse visualization window view for visual history guidance.

```

10  {
11    "PhotoName" : "\\Screenshot-04.25,11:16:54.jpg",
12    "FrontmostPageUrl" : "https://www.amazon.com/s?k=delonghi+coffee+pump&ref=nb_sb_noss_2",
13    "SoftwareName" : "Google Chrome",
14    "bound" : [
15      "0",
16      "0",
17      "1440",
18      "900"
19    ],
20    "FrontmostPageTitle" : "Amazon.com: delonghi coffee pump",
21    "category" : "Browser"
22  },

```

**Figure 10: Data structure stored in a JSON file for the “Browser” category.**

```

10 {
11   "PhotoName" : "\Screenshot-04.25,11:16:54.jpg",
12   "FrontmostPageUrl" : "https://www.amazon.com/s?k=delonghi+coffee+pump&ref=nb_sb_noss_2",
13   "SoftwareName" : "Google Chrome",
14   "bound" : [
15     "0",
16     "0",
17     "1440",
18     "900"
19   ],
20   "FrontmostPageTitle" : "Amazon.com: delonghi coffee pump",
21   "category" : "Browser"
22 },

```

**Figure 11: Data structure stored in a JSON file for the “Productivity” category.**

```

10 {
11   "PhotoName" : "\Screenshot-04.25,11:16:54.jpg",
12   "FrontmostPageUrl" : "https://www.amazon.com/s?k=delonghi+coffee+pump&ref=nb_sb_noss_2",
13   "SoftwareName" : "Google Chrome",
14   "bound" : [
15     "0",
16     "0",
17     "1440",
18     "900"
19   ],
20   "FrontmostPageTitle" : "Amazon.com: delonghi coffee pump",
21   "category" : "Browser"
22 },

```

**Figure 12: Data structure stored in a JSON file for the “Programming” category.**

```

10 {
11   "PhotoName" : "\Screenshot-04.25,11:16:54.jpg",
12   "FrontmostPageUrl" : "https://www.amazon.com/s?k=delonghi+coffee+pump&ref=nb_sb_noss_2",
13   "SoftwareName" : "Google Chrome",
14   "bound" : [
15     "0",
16     "0",
17     "1440",
18     "900"
19   ],
20   "FrontmostPageTitle" : "Amazon.com: delonghi coffee pump",
21   "category" : "Browser"
22 },

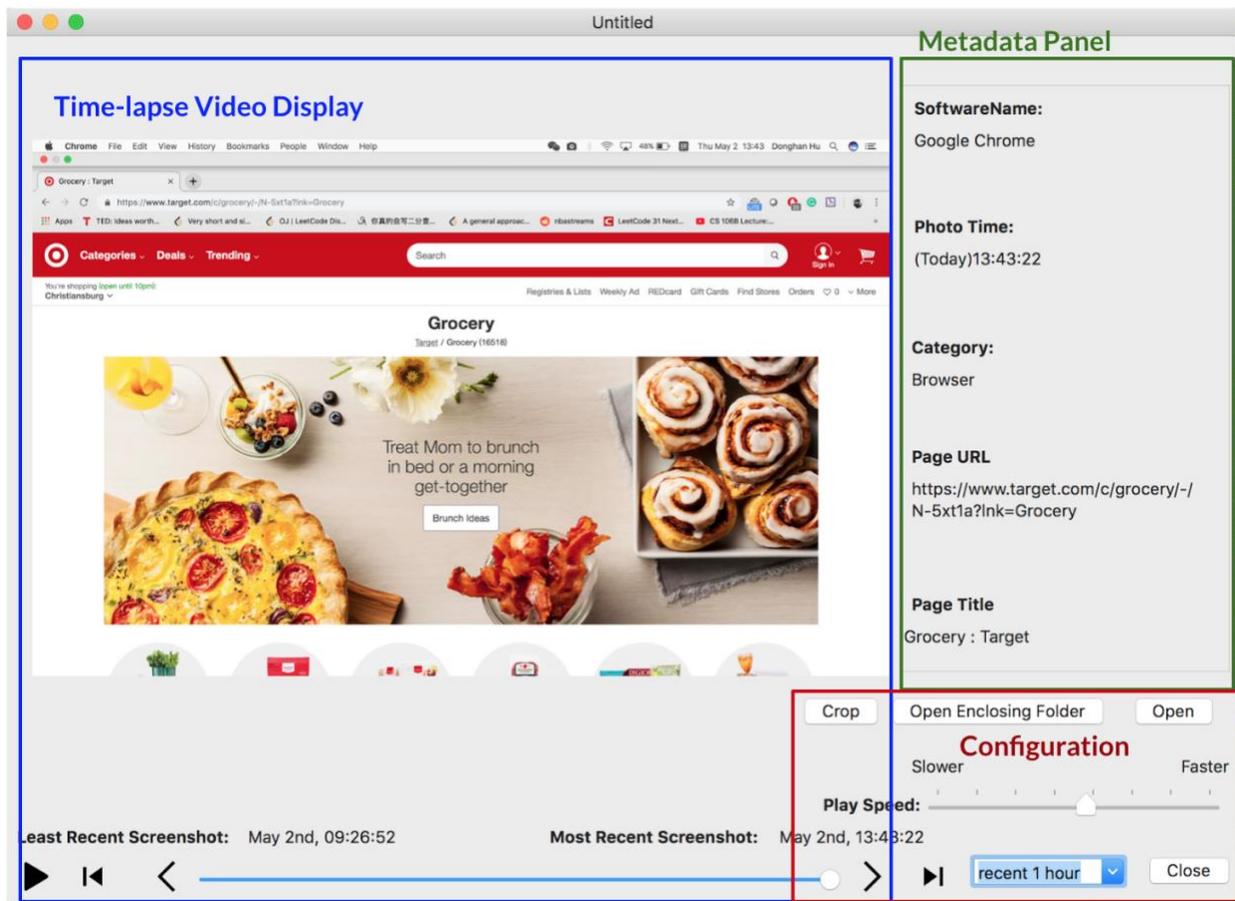
```

**Figure 13: Data structure stored in a JSON file for the “Other” category.**

#### 4.4 Time-Lapse Visualization Window

After clicking the “Time-Lapse Visualization” button in the menu bar interface, ScreenTrack can open a new window of “Time-Lapse Visualization”. This window appears in the middle of the screen. The window of time-lapse visualization is mainly designed to display the previously saved screenshots and related information, provide backtrack functions, and let people reconstruct previous mental context.

As for the arrange of this view, the window is divided into three parts (Figure 14), 1) Time-lapse Video Display, 2) Metadata Panel, and 3) Configuration.



**Figure 14: Time-Lapse Visualization Window. This window has three parts: Time-lapse Video Display, Metadata Panel, and Configuration.**

For the Time-lapse Video Display, this part consists of one image display area, one player progress bar, five player buttons, and two text labels. This part looks like other regular video players, which are familiar with most users, like YouTube player. However, there are still several differences between this visual history and other normal video players.

Instead of playing normal videos, after clicking the “Play” button, this image display area can display previous screenshots one by one, like a time-lapse video. The speed of displaying images can be set on the right. This time-lapse video is not made of continuous frames, but screenshots taken in the last period of time.

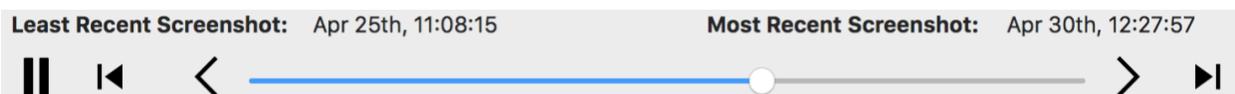
Below the image display area, there are two text labels “Least Recent Screenshot” and “Most Recent Screenshot”. The first message shows users about the time when the least recent screenshot was taken in this period of time. The second message tells users the date of the most recent screenshot. These two static metadata fields are designed as guidance for users.

Time information can helpfully inform users that which period of time that they are reviewing now. Based on these time messages, people cannot get lost in their past computer activities, especially for a long period of time.

Under two labels of time information, there are five buttons and a player progress bar. These five buttons from left to right are “Play/Stop” button, “First Image” button, “Previous Image” button, “Next Image” button, and “Last Image” button. I chose patterns of these five buttons, which are familiar with most users. These patterns can be seen from other widely used video players, which consist of triangles and vertical lines. Hence, users can effortlessly know what these buttons are used for and manipulate easily with these buttons even for the first time.

The “Play/Stop” button is used to start or stop a time-lapse video. If a time-lapse video is not playing currently, a user can click the play button to start playing his visual history at any time. When images are displaying as a time-lapse video, this button’s pattern changes to a pause pattern from the start pattern indicating the new status of time-lapse video. A user can also click this button again to stop playing a time-lapse video at any time he wants. The second button is the “First Image” button, which lets users go to the first image with only one click. Next one is the “Previous image” button. By clicking this button, users can browse previous image one by one. The fourth button is the “Next image” button. This button has the same function as the “Previous Image” button. Using this button, users can review the next screenshot one by one with each click. The last one is the “Last Image” button. After clicking this button, the last recorded screenshot in this period of time is displayed immediately.

These five buttons help users review their previous screenshots of computer screen activities quickly with high effectiveness. The player progress bar is an item that shows users where the time-lapse video is going now. Users can also drag the slider to anywhere of the time-lapse video they want, such as the beginning of a video, or the middle of a video. This bar makes mental context reconstruction tasks much more flexible.



**Figure 15: User Interface Region for Navigating Time of Time-lapse video.**

In this Time-lapse Video Display part (Figure 15), users can review their past computer activities and reconstruct previous mental context with both time-lapse video method and image browsing method. This time-lapse video method lets users review their past works on computers quickly and saves lots of time which might be spent on searching. This image browsing method can help users to locate a specific screenshot accurately.

The second part of this Time-Lapse Visualization window is the Metadata Panel. This area is mainly designed to show related information in textual format. Metadata is collected when ScreenTrack is taking screenshots. To be more specific, related data is based on the category of each recorded software.

In this Metadata Panel, there are five text labels with five corresponding text displaying fields. They are “SoftwareName”, “Photo Time”, “Category”, “File Path/URL Path” and “PageTitle/FileName”. Because ScreenTrack is aimed at enhancing productivity and improving the effectiveness of working and studying, I mainly focused on the categories of Productivity, Browser, and Coding. This above five information is related to the currently displaying screenshot. Firstly, the “SoftwareName” is the localized name of the software which is identified

as the frontmost running software recorded in the screenshot. In Mac OS, each software must have a localized name. Hence, this information cannot be empty. The second metadata is “Photo Time”. This message can tell users about the creation time of this screenshot. People can know when they were using this software intuitively. The third information is “Category” message. This information informs users about the category of this recorded software. Following metadata is “File Path/URL Path”. If the recorded software belongs to the category of “Productivity” or “Programming”, this text field shows the file path of this recorded file or document. If the recorded software belongs to the category of “Browser”, this text field shows the URL of this recorded website. In addition, if the recorded software belongs to the “Other” category, this field shows the word “null” which means there is no related file or website address either. The last text field is “PageTitle/FileName”, which is kind of the same as the fourth one, “File Path/URL Path”. If the recorded software belongs to the category of “Productivity” or “Programming”, this text field shows the file name. If the recorded software belongs to the category of “Browser”, this text field tells users the title(name) of this recorded website. If the recorded software belongs to the “Other” category, this field shows “nil” which means there is no validate file name or website title either. In addition, ScreenTrack shows a reminding word “null” to indicate users there was no related data recorded in the past.

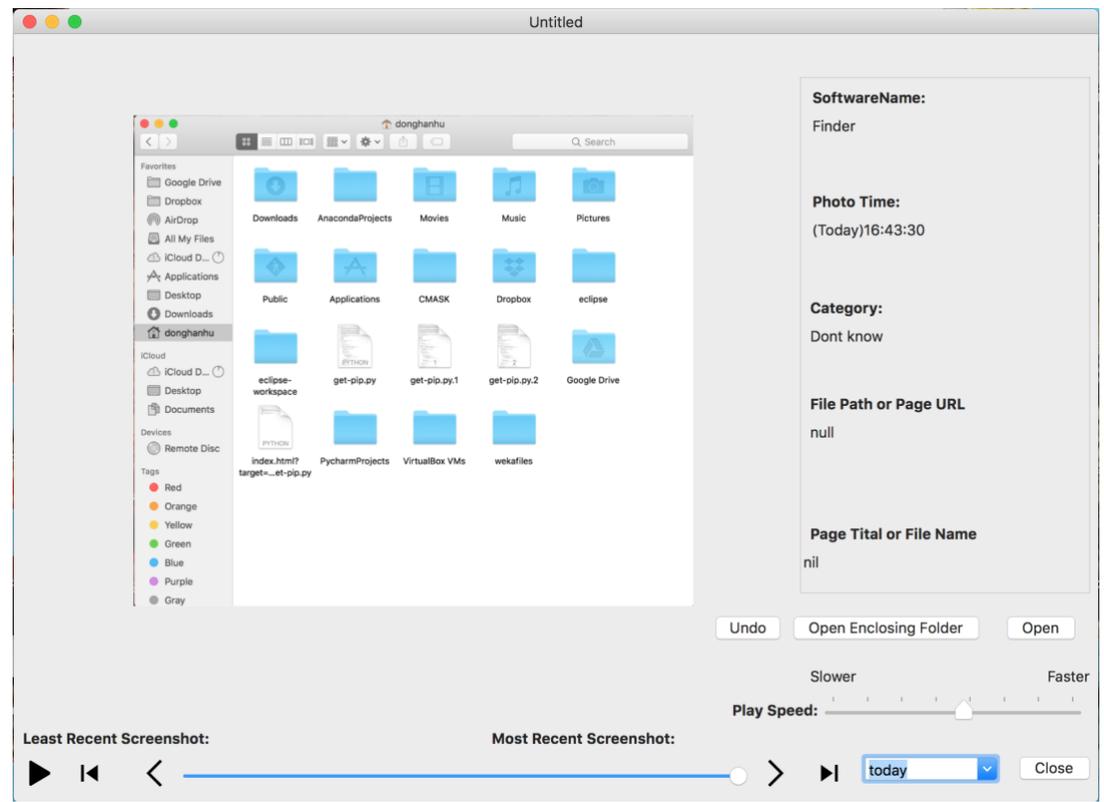
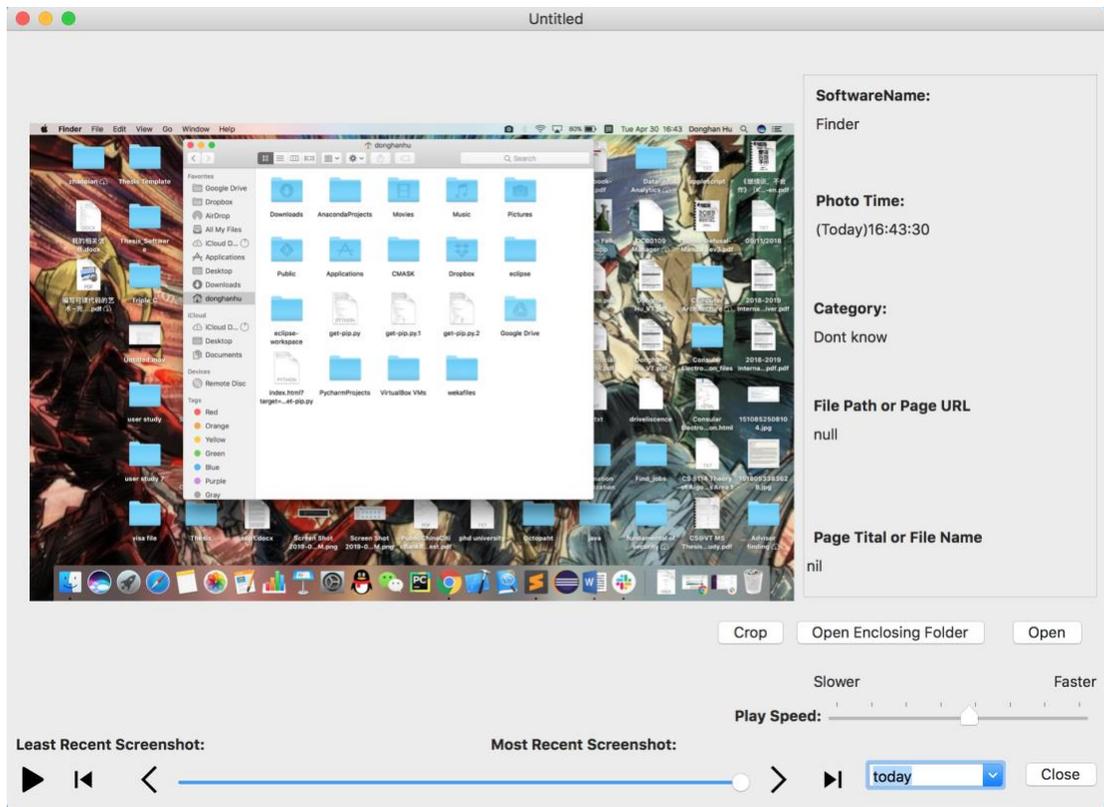
According to this metadata part, users can get useful information about currently displayed screenshot easily and intuitively. This text format method contains more information with less occupied space. Reading text is more familiar with users, especially concise words and nouns. The information of software name, capture time, and related information written in short words can actually help users recall what they were doing in the past quickly and effortlessly.

The third part in this “Time-Lapse visualization” window is the “Configuration” part at the lower right corner of the window. In generally speaking, this area provides users with several functions that can control other parts and helps users to restore their precious working contexts effectively. In this part, there are only four buttons, one slider, and a combo box. The arrangement of these items helps users control the whole window easily.

At the first line of this area, there are three buttons, “Crop” button, “Open Enclosing Folder” button, and “Open” button. The “Crop” button is designed for users to zoom in screenshots. If the software recorded in the current displaying screenshot is not in full-screen mode, which means there is an irrelevant background behind the software, users can click this button to zoom the software into full-screen mode. When users are looking at screenshots for details, the background behind the recorded software may distract users’ concentration. Due to this reason, removing useless background information is one challenge for ScreenTrack to let users focus only on frontmost running software. After clicking the “Crop” button, title on this button changes to “Undo” (Figure 16). Then, users can click this button again to zoom out the screenshot to its original view.

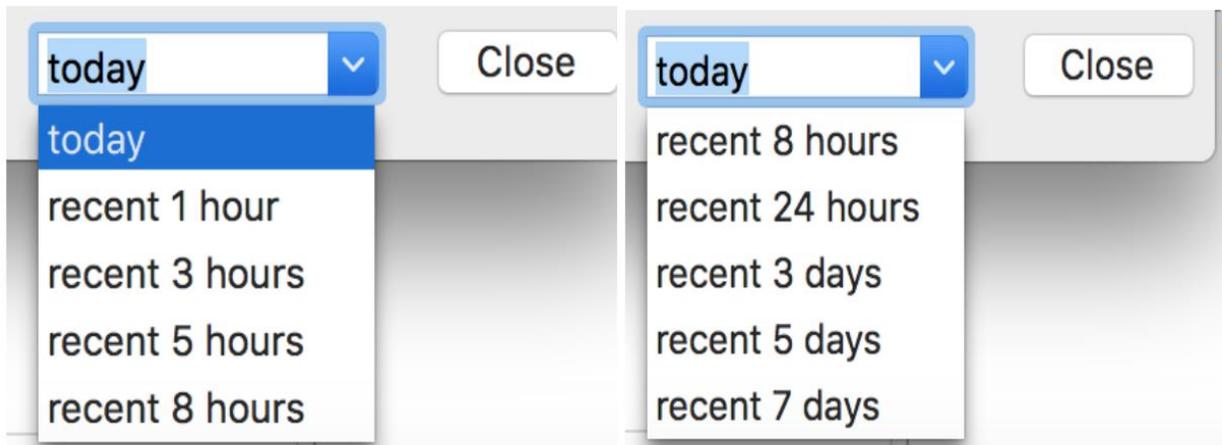
The second button is “Open Enclosing Folder”. This button is specially designed for productivity and coding software. If the recorded software in current displaying screenshot belongs to the category of Productivity or Coding, users can click this button to open the folder which contains the related file, document, and project showed in screenshot. For this button, ScreenTrack does not open this file automatically. Only the Finder application can be opened. If this file,

document, or project has been deleted or renamed, ScreenTrack may not be able to open the corresponding folder. At the same time, an alert window jumps out telling users that no file is found. The third button is the “Open” button. Once users click this button, ScreenTrack can automatically launch the corresponding software which is recorded in the current screenshot. In addition, this launched software appears as the frontmost software on a computer screen. What is more, if this opened software is a browser, this browser can be relaunched and the same website recorded in the screenshot can be opened at the same time. If this opened software belongs to the category of productivity, the file and document which was edited in the past can be reopened meanwhile. These three buttons can help users to figure out their previous activities and go back to past working contexts quickly.



**Figure 16: Screenshots before cropping (Top) and after cropping (Bottom). The example application in this image is Finder.**

Below these three buttons is a slider to control the playing speed of the time-lapse video. To be more specific, the slowest speed of playing screenshots is one image per second. The fastest speed of playing screenshots is twelve images per second. As two labels showing above this slider, users can move this slider to the left for a slower speed and to the right for a faster speed. The default value of playing speed is six images per second. At the lower right corner of the window, there is a combo box that allows users to select a different time period. ScreenTrack offers nine options in this combo box. They are “Today”, “Recent 1 hour”, “Recent 3 hours”, “Recent 5 hours”, “Recent 8 hours”, “Recent 24 hours”, “Recent 3 days”, “Recent 5 days”, and “recent 7 days” (See Figure 17). After choosing a different period of time, ScreenTrack loads different screenshots taken during this period of time. If there is no screenshot has been taken in the period of time that a user has selected, a default image showing “No Image Available” appears on the left. The last button is the “Close” button. Users can click this “Close” button to shut down the current window safely.



**Figure 17: Expanded drop down menu for different options of time period.**

## **Chapter 5: User Study**

In this chapter, I designed the pilot experiment and recruited four participants to finish my pilot experiment. Based on the result of the pilot study, I found out the issues that existed in the pilot study. Later, I scheduled the order of each part of the study and re-designed a new user study. In this chapter, I introduce my pilot experiment, user study, the result of the study, and the analysis of the result in this chapter.

### **5.1 Pilot Experiment and Implementation**

I describe thoroughly the following parts in my pilot experiment design and implement: 1) participants, 2) designing and implementation of the pilot experiment, and 3) issues.

#### **5.1.1 Participants**

I recruited four participants (ages 22-26, two male and two female) to finish the pilot experiment. All of the participants are college students from the Computer Science Department who use their computers every day. They all have high digital literacy and use the computer for different purposes.

#### **5.1.2 Process of the Pilot Experiment**

At the beginning of the pilot experiment, I provided the participants with a consent form. Once the participant agreed to be part of the study, the participant asked to complete two different online tasks while ScreenTrack is on in the laptop that the participant is provided with. After submitting an online survey about demographic questions, the participant was told to start two web-surfing tasks. Each task was limited in 25 minutes. During the web-surfing task, the participant was asked to browse shopping websites and choose furniture for their own rooms. After two web-surfing tasks, I asked participants questions related to their previous web-surfing tasks. Then the last part of the pilot study was a post-task interview. In this part, questions were asked about how these participants felt about ScreenTrack.

For web-surfing tasks, there were two types of room: 1) Bedroom, 2) Living room, and two types of condition for two tasks: 1) Control condition, 2) Experimental condition. Under control condition, I did not run ScreenTrack to track users' screen activities during web-surfing task. Under the experimental condition, ScreenTrack was run to record the whole process of the task of choosing furniture.

To cancel the learning effect, I combined these two factors into four different sequences: 1) Bedroom (Control) + Living room (Experimental), 2) Bedroom (Experimental) + Living room (Control), 3) Living room (Control) + Bedroom (Experimental), 4) Living room (Experimental) + Bedroom (Control). Each participant was assigned one of these four sequences.

#### **5.1.3 Findings in Pilot Experiment**

After finishing the pilot experiment, I found there are several drawbacks and disadvantages in my pilot study. 1) The order of each study part. Planning questions right after the web surfing task did not give me great feedback. Due to the short time interval between each part, participants still remained fresh memory about their previous work contexts. The reason is that no task has been placed between the web-surfing task and the following recall question task. Participants were not interrupted by external or internal factors. Switching working context situation did not exist either. 2) The time assigned for each task. Some participants thought that 25 minutes was quite short for choosing more than three categories of furniture. Sometimes, they did not have their own ideas about buying what kind of furniture. Hence, participants usually spent lots of time thinking about the type of furniture that they should search for. Much time has been wasted instead of browsing websites. 3) The websites that the user chose to browse. Participants would like to select the websites that they used frequently in the past. This fact meant they are familiar with the features and interfaces of these shopping websites. People can see what they have viewed in the past easily. However, this situation would not be common in other websites and software, especially for those without history function. According to the above issues, I need to solve these problems and optimize the user study.

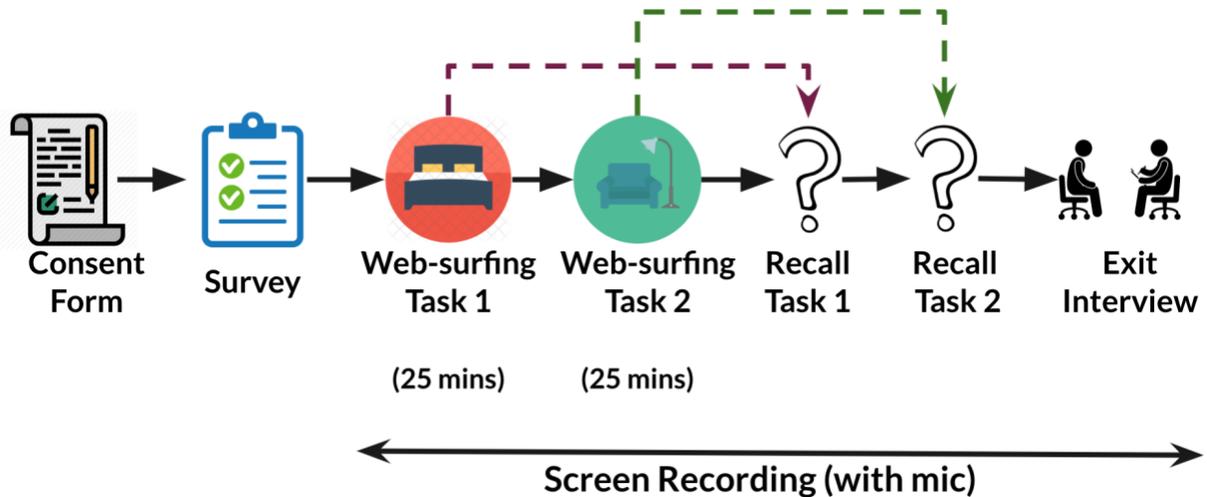
## **5.2 User Study Design and Implement**

### **5.2.1 Participants**

For the participant in my user study, I observed 21 participants (13 male and 8 female) finishing the user study. The average of the participants is 26.2 years old. The range of age is from 22 to 35. All participants have high digital literacy. Most of them surf the Internet more than once a day (18 of 21). The rest of the participant surf the Internet once a day. Hence, all of them are heavy interest users. According to the pre-task survey, most of them use computers in daily life for various purposes. For example, entertainments, communication, working, and studying. After the user study, each participant was paid with a \$15 gift card as compensation.

### **5.2.2 User Study Designed**

Based on the above problems existed in the pilot study, I re-designed the user study, and re-schedule the order of each part in our following user study.



**Figure 18: The procedure of user study. The whole user study contains seven parts: 1) consent Form, 2) Survey, 3) Web-surfing Task 1, 4) Web-surfing Task 2, 5) Recall Task 1, 6) Recall Task 2, and 7) Exit Interview.**

I ran this user study as a within-subject design. This means that one participant completed two web-surfing tasks in both two conditions; one was the control condition, and the other is the experimental condition. The control condition was, during the web-surfing task, ScreenTrack was not turned on to monitor screen activities. No screenshots were taken under this condition. The experimental condition is, during the web-surfing task, ScreenTrack was turned on and record computer screen activities. ScreenTrack captured screen every 5 seconds. Under these two conditions, the participant was not be informed whether ScreenTrack was running or not.

The whole process of the user study is shown in Figure 18. At the beginning of the task, I provided the participant in this user study with a consent form. After reading the consent form carefully, the participant should decide whether he wants to continue with the user study or not. If the participant agrees with the consent form, we would continue with the following steps of the user study. If the participant wants to quit the user study, the participant can stop participating at any time during the user study. The second step of the user study is an online pre-task survey. There were six questions about demographics in this online pre-task survey. Then, the third and fourth steps were two web-surfing tasks. The participant was asked to complete two different web-surfing tasks in the laptop that the participant is provided with. After web-surfing task 1 and task 2, the participant was asked to answer two recall tasks based on the previous web-surfing tasks. Later, the last step of the user study was a post-task interview. I planned to ask the participant about four questions related to this user study. The laptop screen was recorded with the mic turned on starting from the online survey to the post-task interview questions.

### 5.2.3 Web-Surfing Task and Recall Task

The web-surfing task is about browsing specific web pages to simulate the computer activities that involves many different documents. In each web-surfing task, the participant was asked to surf specific online shopping websites for purchasing furniture. They were given a hypothetical

situation where they need to purchase new furniture for their unfurnished house. Websites that participants were only allowed to browse were [www.wayfair.com](http://www.wayfair.com), [www.overstock.com](http://www.overstock.com), and [www.target.com](http://www.target.com), which are chosen based on the variety of choice and the number of products. They were asked to pick one product per type (e.g. one bed, one nightstand, one couch). The participant was also provided with two lists of suggested furniture for two. In the user study, the participant was asked to finish two web-surfing tasks back to back. One task was for a bedroom and the other task was for a living room. The order of two rooms was randomized for each participant. To be more specific, participants were told to do the web surfing task for 25 minutes, with 5 minutes notice. When the time was running out, the participant would be asked to stop browsing websites and move on to the next part of the user study. In addition, for each task, the participants were asked to choose at least three types of furniture per room and no more than five. Flexible numbers of furniture gave participants enough time to choose their favorite furniture without hurry.

In the web-surfing task, there are two factors that I have to randomize and balance to cancel the learning effect in the user study. First one was whether to use ScreenTrack. I randomized this factor in the user study whether participants are using ScreenTrack or not. But participants were not informed about this factor. The second factor was the room type. I have chosen two different types of room: bedroom and living room. Hence, there are four different types of tasks by combining these two factors. Later in the user study, one participant would do one of these two pairs in randomized order, which makes four totally different sequences. Following are four different sequences.

	Bedroom	Living Room
Control Condition (w/o ScreenTrack)	Bedroom + Control (BC)	Living Room + Control (LC)
Experimental Condition (w/ ScreenTrack)	Bedroom + Experimental (BE)	Living Room + Experimental (LE)

**Table 2: Four sequences for user study with two factors.**

- Sequence 1: BC (Task 1) + LE (Task 2)
- Sequence 2: BE (Task 1) + LC (Task 2)
- Sequence 3: LC (Task 1) + BE (Task 2)
- Sequence 4: LE (Task 1) + BC (Task 2)

I have recruited 21 participants for my user study in total. As a result, each sequence has 5 participants on average, one more participant was recruited for a backup and assigned to one of the sequences randomly (Sequence 1).

For the step of the recall task, I asked recalling questions related to the previous web-surfing task to participants. At the beginning of each recall task, participants were asked how many types of furniture have they chosen and what were they in the order of searching. I then asked the participants to complete two following tasks for each furniture type for both the control condition and the experimental condition. Following are two questions asked for each trial in the recall task part:

- Question 1: Could you please describe the detail of the favorite furniture?
- Question 2: Could you open the webpage that has the detail of the product that you have chosen (Using ScreenTrack or using the browser)?

The purpose of asking Question 1 is to for the participant to pick one particular product. Before asking questions for experimental condition, an interactive tutorial was run for participants to show them how to use ScreenTrack. The tutorial was mainly about displaying time-lapse video, locating past screenshots and retrieving previous information. During the tutorial of how to use ScreenTrack, participants would be asked to try different features that can help them retrieve previous work context. I had to make sure that participants were familiar with the ScreenTrack before moving on to the next step. After the tutorial, I requested participants above two questions and then continued to finish the rest of the user study.

## Chapter 6: Result

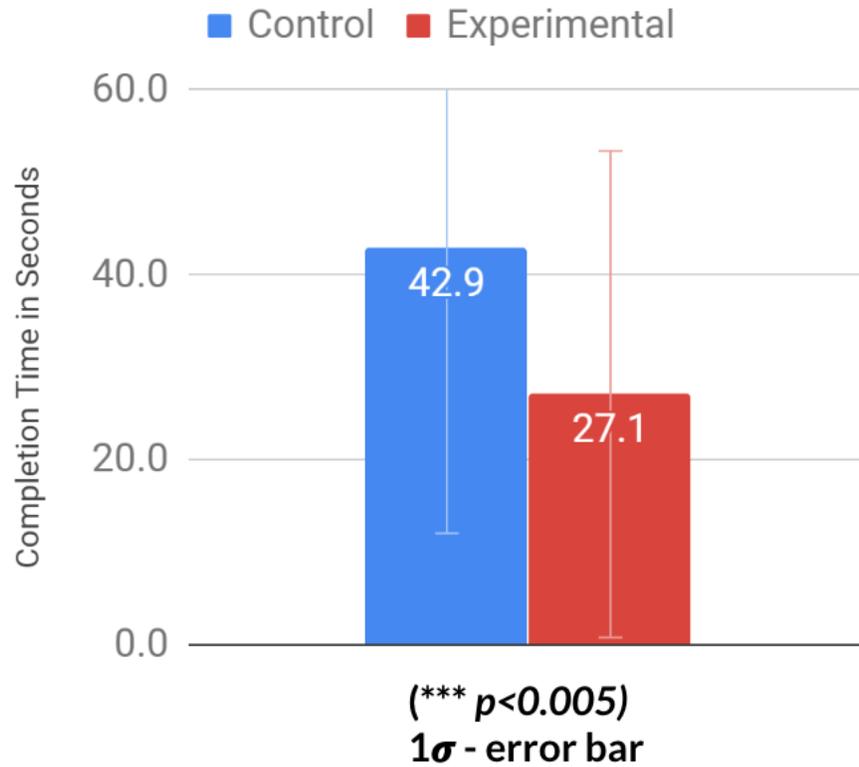
In this chapter, I introduce the validation of ScreenTrack based on the result of the user study. After the user study, I collected the completion time for participants to retrieve each closed website. Based on the recording videos for all participants, I transcribed the whole interview and ran a thematic analysis. Then, I give answers for two research questions that have been mentioned in the introduction chapter.

### 6.1 Result and Analysis for Time Efficiency of Working Context Retrieval

***RQ1: Does using the visual history of a computer help users retrieve relevant information more quickly than other existing methods?***

With 21 participants, I have collected completion time data for 63 trials for the control condition and 62 for the experimental condition. Each participant completed three tasks per condition ( $3 \times 21 = 63$  trials). One data point is missing in the user study because a participant picked the exact same furniture from the other condition, which made it not necessary to retrieve the web page again. For the time interval of the control condition, the beginning of the time interval is when a participant launches the browser. The end of the time interval is when a participant opened the web page that they were looking for. For the experimental condition with ScreenTrack, the beginning of the time interval is when a participant opened the time-lapse video interface. The end of the time interval is the same as the control condition. I considered these time intervals are the time that participants spend to retrieve the previous closed web pages. For each user study, I recorded the time interval from three retrieving web page trials for each condition. If a participant chose more than three types of furniture, only the first three products were asked to retrieve in the user study.

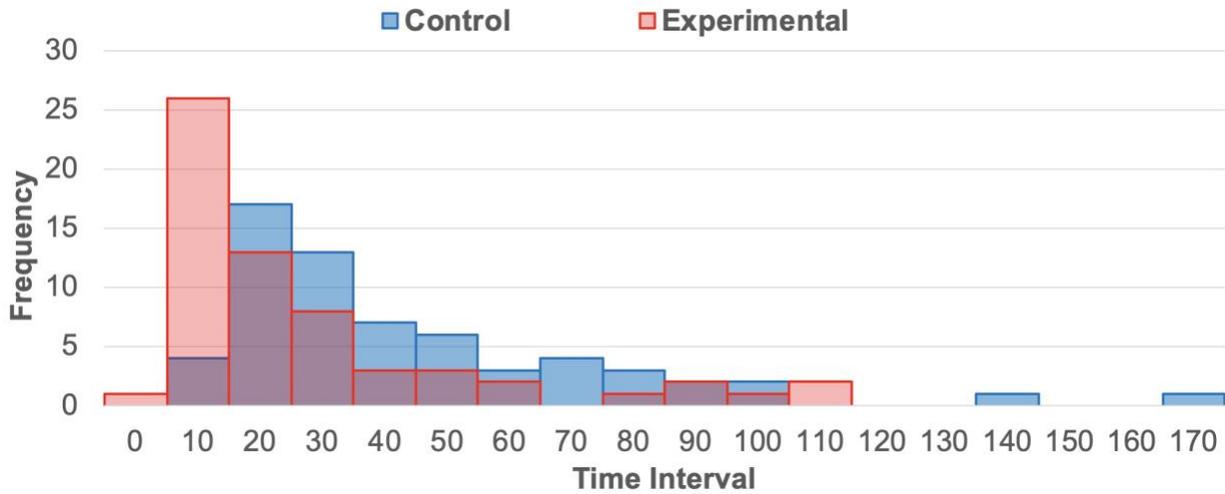
The result indicated that ScreenTrack allows the participants to retrieve a web page of interest quicker on average. In the control condition, the average time spent to retrieve a web page is 42.9 seconds ( $\sigma=30.8$ ). In the experimental condition, the average time spent to retrieve a web page is 27.1 seconds ( $\sigma=26.3$ ). Participants in the experimental condition retrieved a web page is 16.8 seconds faster than the control condition on average. As a result, retrieving a web page is 37% quicker when using ScreenTrack than without using it. The result has been shown in the following bar chart (Figure 19).



**Figure 19: Average time spends on retrieving a previous closed website on the control condition and experimental condition.**

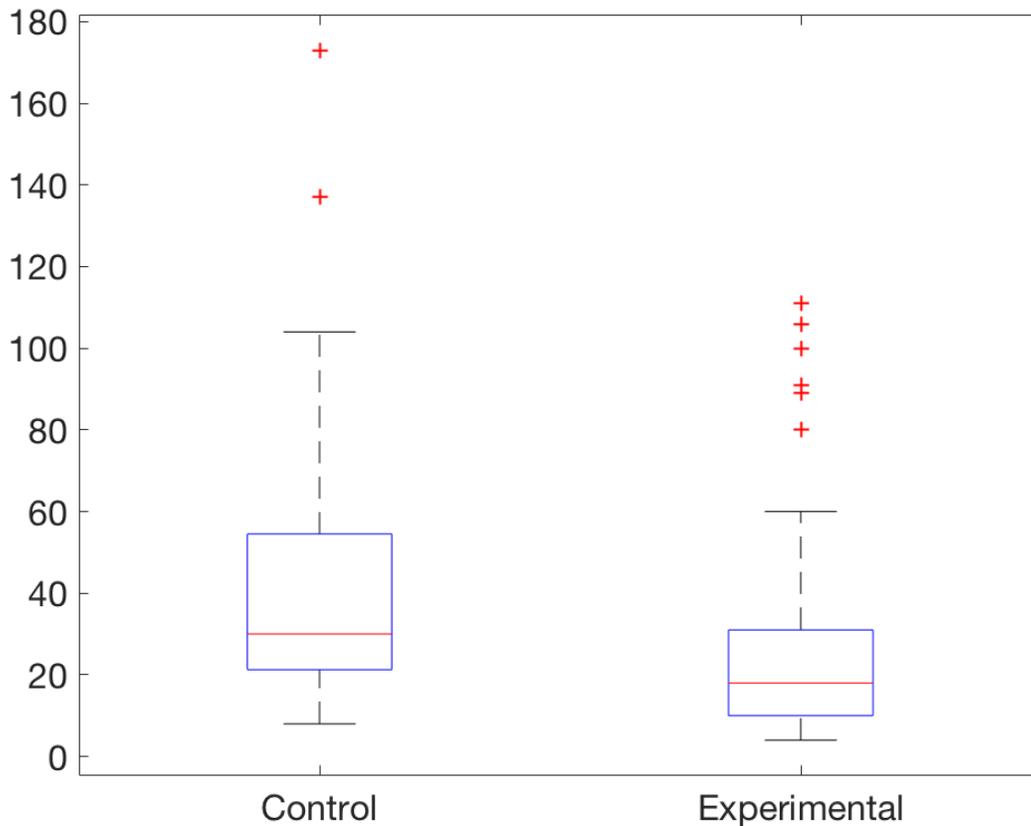
In the above figure of the bar chart, the values of standard deviation are wild for both control condition and experimental condition. What is more, error bars are large for these two conditions. Based on this fact, I looked further into the raw data and made two charts: the distribution chart (Figure 20) and the boxplot chart (Figure 21).

In the distribution chart, I found that in the experimental condition, 56 trials of time interval data are from 0 seconds to 70 seconds. 6 data points are from 80 seconds to 110 seconds. While in the control condition, there are 61 data sets from 0 seconds to 110 seconds which is greater than the experimental condition. 2 data points are greater than 140 seconds which are also higher than the maximum value of experimental condition data. I found that data points in the experimental condition are distributed more densely than the control condition. Both two conditions have data points which are far from each average condition value. Due to the small number of time samples, few data value can influence the whole result of each condition. These data distributed far from the average value can slightly explain why the error bars in Figure 19 are quite wide, and the standard deviations are large.



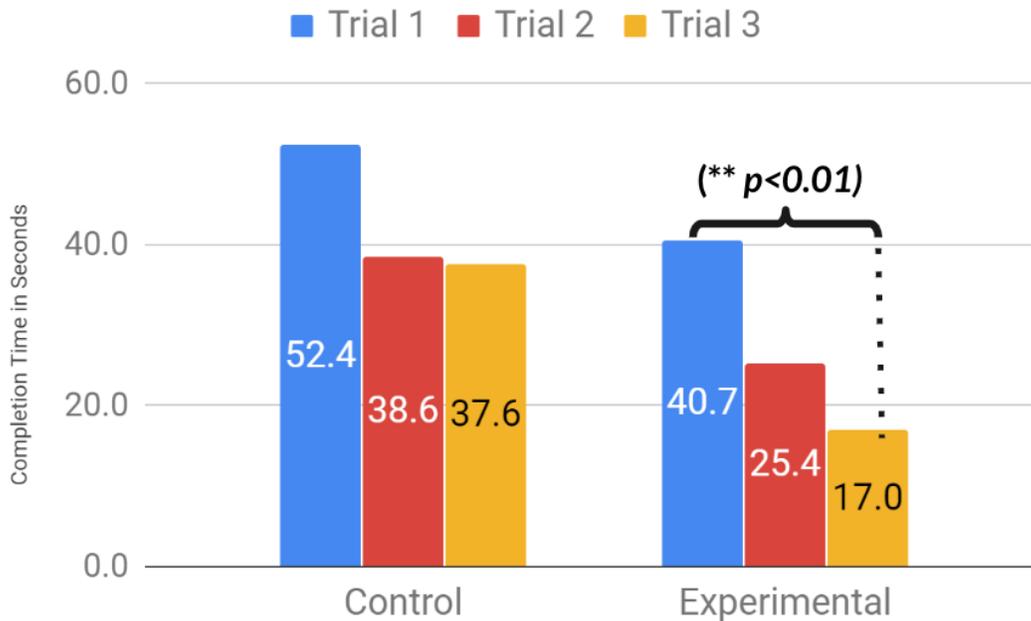
**Figure 20: Distribution of data points for control condition and experimental condition.**

Based on the boxplot chart (Figure 21), I found that there are 2 outliers out of the boxplot in the control condition. There are 6 data outliers out of the boxplot in the experimental condition. I only have 63 data points for the control condition and 62 data points for the experimental condition, which are quite a small number of samples. These outliers can influence the standard deviation and error bar for each condition. Hence, the outliers pointed out of the boxplot can also explain why the error bar in Figure 19 is quite wide, and the standard deviations are large.



**Figure 21: Boxplot for data points in control condition and experimental condition.**

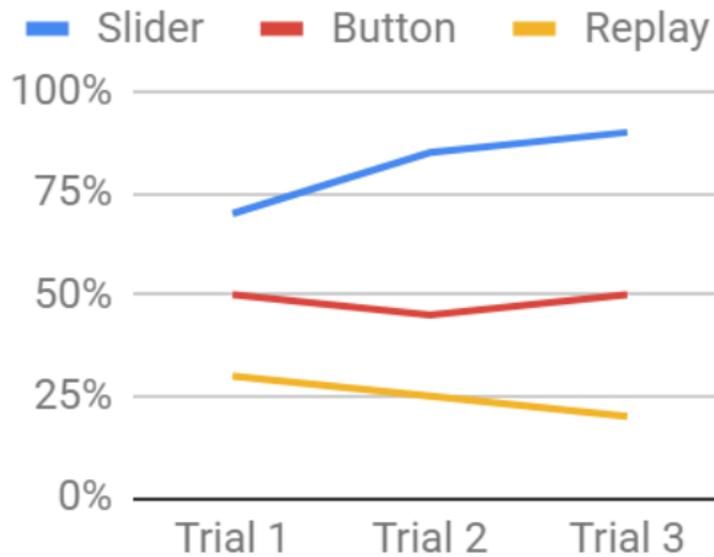
Whether learning effects exist in my user study or not is another question. For this question, I looked at further details on how participants performed in each trial for the control condition and the experimental condition. In the control condition, the average time spent on retrieving a website for trial one is 52.4 seconds. For trial two, the average time is 38.6 seconds. The average time for trial three is 37.6 seconds. Participants spent about less than 40 seconds for trial two and trial three. According to this result, in the control condition, the difference was not statistically significant between each trial ( $p > 0.05$ , two-tailed t-test unequal variance). On the other hand, in the experimental condition, the reductions are clearer between these three trials. For the first trial, the average time spent on retrieving the web page is 40.7 seconds. For the second trial, the average time is 25.4 seconds. The average time for trial three is 17.0 seconds, which is much smaller than the first trial. The difference between trial one and trial three is statistically significant ( $p < 0.01$ , two-tailed t-test unequal variance). According to the data and bar chart (Figure 22), I found that learning effects exist in the experimental condition. Participants spent less time on finding previous closed web pages gradually. Participants were getting better in locating the websites that they were looking for under the assist of ScreenTrack.



**Figure 22: Average time spends on retrieving a previous closed website for each trial on the control condition and experimental condition.**

How participants locate websites under experimental condition? Why there is time decrease between each in experimental condition? One potential reason to explain this improvement is the change in UI elements usage over time in the experimental condition is the ways in which participant navigated time differently over trials. In Figure 23, there is an increasing trend of using slider feature from 70% to 82%. The trend of using replaying feature is reduced from 30% to 20%. Participants did not only use one method to help them retrieve previous information. In trial one, some participants chose to play the video from the beginning to help them locate the websites. While in trial three, more people decided to use a slider to help them find the right screenshots in my software. As a result, in the experimental condition, participants preferred using slider function (82%) and button functions (48%) to replaying the time-lapse video (25%) to locate the right screenshots and open the websites. Just like participant 1 (P1) mentioned as follows:

*P1: "I was searching for the couch first it took me longer time. Because I was just going through the entire video but for the second I knew like I had a rough idea whether relative location was. So, it seems like much quicker actually."*



**Figure 23: In experimental condition, the usage of three features in ScreenTrack to help participants retrieve previous closed websites.**

I have made following further observation based on the result of the user study. On average, participants searched for 3.8 types of furniture for both a bedroom and a living room. In the control condition, participants preferred using history function in a browser (73.3%) to researching the product again (15%). There was no significant difference ( $p > 0.05$ , two-tailed t-test unequal variance) for room types (Bedroom vs Living room) or the task order (Task 1 vs Task 2).

Based on the above charts and analysis, I can make a conclusion for research question one. Using the visual history of a computer helps users retrieve relevant information more quickly than other existing methods for the participants. While it will need a larger-scale user study, it seems promising that ScreenTrack can help people retrieve their past working context (web pages) more quickly than existing textual browsing history method.

## 6.2 Result and Analysis for Perceiving the Values of Using ScreenTrack

*RQ2: How do users perceive the values of using ScreenTrack in their computer usage?*

To answer this question, I designed a post-task interview with four questions in this user study. In this post-task interview part, I asked four following question to participants which are related to the user study and ScreenTrack.

- Question 1: What did you like about the system? What did you NOT like about the system?
- Question 2: Would you like to use this kind of software (or something similar) on your computer?
  - If yes, how would you use it?

- If no, what makes you not to use such software?
- Question 3: Do you have any concern in using the system?
- Question 4: Do you have any suggestion on how we can improve the system?

After finishing all these 21 user studies, I transcribed the whole interview and then ran a thematic analysis. Based on the thematic analysis, I got the following results from 21 participants' interview answers.

### **6.2.1 ScreenTrack is Easy to Use and Intuitive.**

In the interview, 15 participants (out of 21) said that the user interface of ScreenTrack is easy to use and intuitive. Participants were favorable to the interface because it is easy to use as watching a video has already become a familiar concept. Instead of reading metadata on the traditional history list, watching a time-lapse video or visual history are easy for users to find what they are looking for. Also, watching a video is already a familiar concept for most people, especially for people who use a computer as a daily tool. People do not need to spend extra effort to learn how to use it. After a user finding the screenshot that he is looking for, the user can reopen previous websites and working contexts with only one click, which is easy and intuitive. Following are what participants said in the interview part.

*P4: "it's easier to use compared to the web browser history where I have to actively search for the keyword. (...) yeah it follows a chronological order that can even further help me back to chase what I did and what webpage I clicked. of course, it can direct me right to the webpage which is pretty cool and easy."*

*P21: "I mean it's easy enough to operate which was kind of nice it was super intuitive and it made sense."*

### **6.2.2 Single Visual History with Screenshots**

According to the interview, participants (10/21) liked the way that they can retrieve the previous working context from a time-lapse video of screenshots. Not only providing users with static metadata and URL messages but using visual history feature can help users recall more details about previous working context. Users can actually see what they were doing in the past through screenshots and visual history. People can check more details than only metadata. This visual history helps people recall previous working context quickly and more comprehensively. Also, according to interview responses, using visual history guidance is a great design for them while recalling past activities. Following are what participants said in the interview part.

*P3: "I already forgot the things that are selected in the past and visualizing them in one place is good."*

*P9: "I would say an easier version of the history and you see you have to go again all those things, but this way you can directly switch on and you can see the visualization so rather than links and all."*

*P8: "So I like how it can automatically detect the URL and direct to the page. it has the time with lapse to see all the histories which is nice."*

In addition, many participants also said this software would help them in other situations: web browsing, online shopping, document writing, and demonstration. They would like to use ScreenTrack for their daily computer tasks. As I can see in these responses, participants have high expectations for ScreenTrack to be used in their everyday life.

### **6.2.3 Participants Could Imagine Using ScreenTrack.**

Participants mentioned that they would like to use this software in their computers for various situations. Firstly, 9 participants indicated that they would like to use ScreenTrack while doing literature review and research tasks. Participants said this software could help them especially record research papers which are hard to find in the later work. Some participants noted that currently, they do not have efficient ways to track the paper that they have read before. Reopening papers that have been read before is a difficult task for them. Following are what some participants said in the interview part.

*P3: "If here's the thing I love to have something like this when I do the Google Scholar thing. Because I follow and go through lots of pages and most of the time you don't track them."*

*P20: "Another thing might be doing I mean for literature search. Definitely like going from paper to paper and forget which one is which."*

*P17: "I can fall through these things other than that even in research like anywhere you want to understand what you did in the past."*

In addition, many participants also said this software would help them in other situations: web browsing, online shopping, document writing, and demonstration. They would like to use ScreenTrack for their daily computer tasks. As I can see in these responses, participants have high expectations for ScreenTrack to be used in their everyday life.

### **6.2.4 Participants Were Concerned with Privacy Issue.**

Participants pointed out their several concerns while using the software in interview questions. First of all, 18 participants mentioned that privacy issue is what they care most during the procedure of using ScreenTrack. Firstly, participants worried that this software would record personal information and might leak this information to someone else, like credit card information, bank account, and usernames of shopping websites. Personal information is private. Participants would not like to record and save this information on their computer. They were also not sure whether this software is trustworthy or not. Secondly, participants worried that personal recording information on the computer without protections is not a good idea. Others who can access their computers can also access these personal recordings easily. There are no protections like passwords to protect this data in my current software. Following are what participants said regarding privacy in the interview part.

*P2: "It raises privacy concerns that whatever what if you want to do something like entering credit card information, any personal information, anything personal. then because a lot of times while doing the search tasks you do branch out into another task which is more of a personal maybe reading emails. For that matter so how would it protect against sometimes question."*

*P10: "I mean I feel like there's always the concern that some people have that as this may get into other people's hands. Or the data gets stored in ways it gets sent to anybody else."*

*P16: "I mean it's recording me, so that's so concern, that so is that on because of this data leak or it goes into somewhere then it might be in trouble, that's a problem."*

### **6.2.5 Eating up Computational Resources (Disks / CPU)**

The other concern that participants cared is eating up resources for both disk and CPU. ScreenTrack is recording the computer screen activities all the time. It needs much space in the local drive to store screenshots, especially when the time interval of taking a screenshot is short. Recording a computer screen for a long time can generate lots of screenshots. Hence, participants worried that using this software might take lots of disk space and CPU memory of their own computers. What is more, one participant also worried that keeping reading data and writing data may slow down his computer. Following are what participants said in the interview part.

*P4: "Maybe instead of taking the screenshot every constant time taking the screenshot at or respond to a certain action, like when I open a web page or when I scroll the mouse things like that would trigger a taking picture function, instead of taking constantly."*

*P15: "it stores it on the actual hard drive of the computer in which case you are saving all of these images onto your computer it's gonna slow it down."*

*P18: "Data size. I don't know how large the data files are."*

Based on the responses from participants in my user study, I ran a thematic analysis to summarize these answers. According to the thematic analysis, participants stated that ScreenTrack was easy to use and intuitive to first time using. Using visual history feature in ScreenTrack helped users check more details about their past activities and also assisted them in retrieving previous working context quickly. Besides, participants mentioned that they would like to use ScreenTrack in their daily life for various goals, like literature reviewing, web surfing, and document writing. What is more, participants indicated their concerns while using ScreenTrack to record their computer activities. Privacy problem and disk requirements are two main concerns that many users care about.

### 6.3 Limitations

I designed and implemented the software ScreenTrack help people self-track their computer screen activities and retrieve past working contexts. Based on screenshots, ScreenTrack provides users with a visual history of previous computer activities to help people reconstruct past working context quickly. After the development of user study, there are still several potential problems and limitations that existed in ScreenTrack and user study.

Firstly, my user study lacks validation study for mental reconstruction and interrupted task resumptions. In my current user study, I did not interrupt participants during the user study. In addition, participants were not asked to reconstruct their mental context and to resume previous work in a period of time either as confirmed in other studies (Rule et al., 2017; Parnin and DeLine, 2010). To address this limitation, a longitudinal study is desirable to validate the effect of ScreenTrack about mental reconstruction or a new user study that is composed of two in-lab study with longer time interval (e.g. one week interval) The new user should focuses on interruptions, mental reconstruction, and task resumptions to test the above values. For example, I can add unexpected interruptions during the user study and then ask users to recall their previous work with or without ScreenTrack.

Secondly, my user study in the current stage was implemented in a short time window (about 75 minutes). Unlike other previous user studies continued for a few days, in-lab user study finished in hours cannot provide sufficient proof that ScreenTrack can help user recall previous working context after a longer period of time. Based on this limitation, user study needs a different representation for a longer time scale, such as designing a new user study for a long period of time. A new user study may continue for days, weeks, and even months. Thirdly, I have received some reflections for ScreenTrack from the participants' responses in the user study.

Thirdly, privacy problem is always the most concern for most participants. For this sort of self-tracking software, people care about their personal information especially. Lacking effective protecting methods is one limitation of the privacy aspect. For this problem, participants suggested that ScreenTrack can add a password feature to protect personal recordings. Another advice is blocking recording function automatically for some specific websites and software, like bank websites.

Fourthly, participants mentioned that they think local memory is another potential problem. Taking screenshots and saving them into local disk can take lots of disk memory for personal computers. For this concern, participants suggested that deleting outdated files automatically for more disk space is one example. If ScreenTrack keeps taking and storing more images without deleting them automatically, this software would definitely occupy too much disk space for local disks. About the usage of ScreenTrack, more cues can be added into the time-lapse video interface. This is another suggestion from the user study interview. People like to gain more kinds of information while browsing their past computer activities. For example, showing time information and thumbnails above the slider is one suggestion.

## Chapter 7: Conclusion and Future Work

### 7.1 Conclusion

This thesis presents a research of self-tracking computer activities for retrieving working context. I tested this idea with the development of ScreenTrack. This software can help users record their daily computer activities and retrieve previous working contexts from a time-lapse video. Given users frequently switch context and are interrupted or distracted while working, retrieving working context can be facilitated with the visual history provided by the software. ScreenTrack, provides users with a new method of self-tracking to record computer screen activities and a visual history feature to help people recall previous tasks. The software was created based on the lifecycle of software development: literature review, problem definition, requirements design, architecture design, software components design, and development. The whole process was integrated verification and validation by the following user study.

The user study confirmed the effects of ScreenTrack in facilitating an efficient retrieval of working context. According to the result of the user study, it was found that participants were able to retrieve information that they were asked more quickly with ScreenTrack than the control condition with statistical significance ( $p < 0.005$ ). Using ScreenTrack could decrease the time of retrieving a closed website for 15.8 seconds on average. Learning effects existed in the user study, which indicates that the effects of ScreenTrack can be bigger as users use this more consistently. I also explored how the participants find the value of using ScreenTrack during a follow-up interview. In addition, participants gave positive feedback on the software that they would like to use such software in their computers in various context and expressed potential concerns of using such software for privacy and computer storage problems. As a conclusion, ScreenTrack can contribute to enhancing productivity for people while they are using their computers in their daily life.

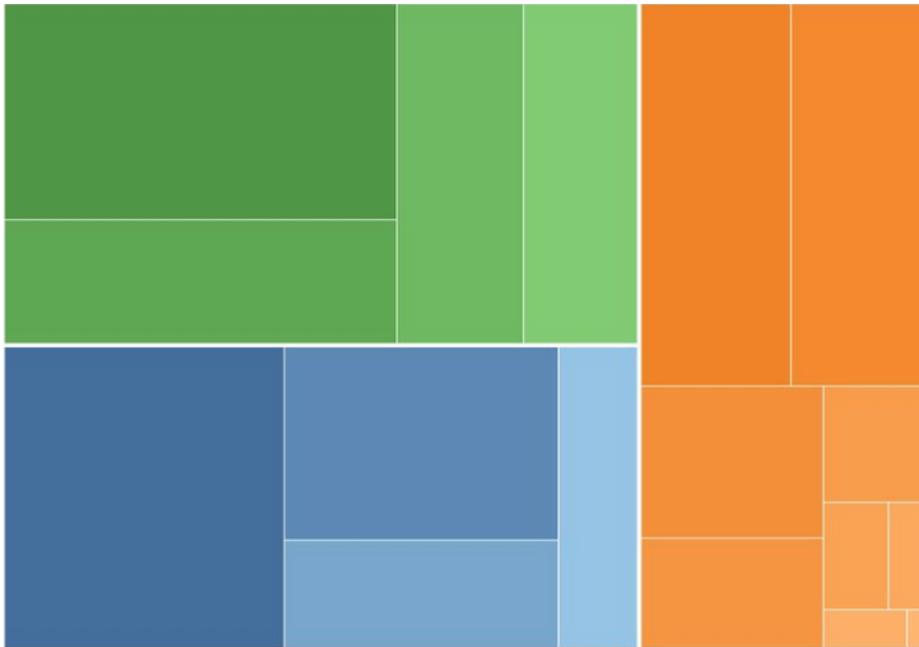
### 7.2 Future Work

I strived to develop ScreenTrack with a focus on helping people self-track their computer activities and using a visual history of previous work to assist users in reconstructing mental context quickly. While I was focusing on the part of self-tracking and mental context reconstruction, there are several other topics that can be studied and implemented to create a feature-rich software, like security features. Based on the responses from the participants and design considerations, I found that there are three main aspects which should be improved and developed in future work. Firstly, more attention should be paid to the privacy problem. In the current method, ScreenTrack did not provide effective methods to alleviate users' concerns on potential leaks of personal data during the procedure of recording the computer screen activities. Storing personal information safely in computers has not been considered thoroughly either. Showing more protection features to users can let them trust ScreenTrack. In future work, I plan to add a password feature to protect local files and time-lapse video function. People who do not know the password cannot access to these screenshots folders and cannot open time-lapse video window either. In addition, I plan to add a blocking feature that can stop recording activity automatically when users launch some specific software or browse some specific websites, like bank webpage. For these personal situations, users do not have to stop and restart recording

manually. Meanwhile, users do not have to worry about saving privacy information into their computers.

Secondly, disk-space is another concern that I should consider more in my future work. Based on this consideration, I plan to apply intelligent algorithms to ScreenTrack when it is taking screenshots. The software should detect users' physical activities, like mouse movement and keyboard activities in order to skip capturing redundant screenshots. In the current method, ScreenTrack takes a screenshot for a fixed time interval or after detecting frontmost running software switch. If a user leaves his computer without stopping recording, the software can take lots of same screenshots, which are unnecessary. These images even heavy the workload of displaying visual history in the future. Hence, for this situation, ScreenTrack should stop taking screenshots automatically if there is no additional mouse activities or keyboard activities. In addition, this software should detect not only the software's switching but also the switch of websites which are opened in a browser. If a user browses a web page more than a specific time, the software should not keep taking screenshots for this website. To save more disk space, ScreenTrack should delete old outdated history folders automatically. For example, the software should be able to remove screenshots folders which were created one month ago. If ScreenTrack keeps creating new folders and saving screenshots without deleting function, these files may occupy too much disk space for computers.

Thirdly, ScreenTrack is mainly designed for self-tracking and mental context reconstruction in the current stage. I need to implement more features help users self-reflect themselves under the help of ScreenTrack. The transition from self-tracking to self-reflection is needed. Instead of the time-lapse video feature, a new representation for a longer time window is required for the self-reflection feature. Using collage view to give users an overview of how much time they have spent on each software or application in a longer time scale (weeks and months) is one plan. Figure 24 present some ideas on how we can represent longer time period in a summarized form; different colors show different categories of software. For example, green is for productivity, and blue is entertainment. The size of each rectangle shows how much time they have spent on this. Bigger square means more time. This collage view can help users understand how they spent their past time and do self-reflection task.



**Figure 24: College view for self-reflection during long time window.**

## References

- Ayers, E. Z., & Stasko, J. T. (1995). Using graphic history in browsing the World Wide Web. Georgia Institute of Technology.
- Brumby, D. P., Cox, A. L., Back, J., & Gould, S. J. (2013). Recovering from an interruption: Investigating speed– accuracy trade-offs in task resumption behavior. *Journal of Experimental Psychology: Applied*, 19(2), 95.
- Carver, C. S., & Scheier, M. F. (2001). *On the self-regulation of behavior*. Cambridge University Press.
- Choe, E. K., Lee, N. B., Lee, B., Pratt, W., & Kientz, J. A. (2014, April). Understanding quantified-selfers' practices in collecting and exploring personal data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1143-1152). ACM.
- Choe, E. K., & Lee, B. (2015). Characterizing visualization insights from quantified selfers' personal data presentations. *IEEE computer graphics and applications*, 35(4), 28-37.
- Christel, M. G., Hauptmann, A. G., Wactlar, H. D., & Ng, T. D. (2002, December). Collages as dynamic summaries for news video. In *Proceedings of the tenth ACM international conference on Multimedia* (pp. 561-569). ACM.
- Cuttone, A., Petersen, M. K., & Larsen, J. E. (2014, June). Four data visualization heuristics to facilitate reflection in personal informatics. In *International Conference on Universal Access in Human-Computer Interaction* (pp. 541-552). Springer, Cham.
- Coraggio, L. (1990). Deleterious effects of intermittent interruptions on the task performance of knowledge workers: A laboratory investigation.
- Czerwinski, M., & Horvitz, E. (2002). An investigation of memory for daily computing events. In *People and computers XVI-memorable yet invisible* (pp. 229-245). Springer, London.
- Czerwinski, M., Horvitz, E., & Wilhite, S. (2004, April). A diary study of task switching and interruptions. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 175-182). ACM.
- Dabbish, L., Mark, G., & González, V. M. (2011, May). Why do i keep interrupting myself?: environment, habit and self-interruption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 3127-3130). ACM.
- DiClemente, C. C., Marinilli, A. S., Singh, M., & Bellino, L. E. (2001). The role of feedback in the process of health behavior change. *American journal of health behavior*, 25(3), 217-227.
- Ding, W., Marchionini, G., & Soergel, D. (1999). Multimodal surrogates for video browsing. *arXiv preprint cs/9902019*.
- Doherty, A. R., Caprani, N., Conaire, C. Ó., Kalnikaite, V., Gurrin, C., Smeaton, A. F., & O'Connor, N. E. (2011). Passively recognising human activities through lifelogging. *Computers in Human Behavior*, 27(5), 1948-1958.
- Elsden, C., Kirk, D. S., & Durrant, A. C. (2016). A quantified past: Toward design for remembering with personal informatics. *Human–Computer Interaction*, 31(6), 518-557.
- Endsley, M. R. (1997). The role of situation awareness in naturalistic decision making. *Naturalistic decision making*, 269, 284.

- Epstein, D. A., Ping, A., Fogarty, J., & Munson, S. A. (2015, September). A lived informatics model of personal informatics. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (pp. 731-742). ACM.
- GoPro | The world's most versatile action cameras. (n.d.). Retrieved from <https://gopro.com/>
- González, V. M., & Mark, G. (2004, April). Constant, constant, multi-tasking craziness: managing multiple working spheres. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 113-120). ACM.
- Heeris, J. (2013, October 16). The Slightly Disgruntled Scientist. This Is Why You Shouldn't Interrupt a Programmer - The Slightly Disgruntled Scientist. Retrieved from <http://heeris.id.au/2013/this-is-why-you-shouldnt-interrupt-a-programmer/>
- Hollan, J. D. (2012). Activity-enriched computing: capturing and mining activity histories. *Computer*, 45(10), 84-87.
- Hoyle, R., Templeman, R., Armes, S., Anthony, D., Crandall, D., & Kapadia, A. (2014, September). Privacy behaviors of lifeloggers using wearable cameras. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (pp. 571-582). ACM.
- Iqbal, S. T., & Horvitz, E. (2007, May). Disruption and recovery of computing tasks: field study, analysis, and directions. In CHI (Vol. 7, pp. 677-686).
- Jeuris, S., & Bardram, J. E. (2016). Dedicated workspaces: Faster resumption times and reduced cognitive load in sequential multitasking. *Computers in Human Behavior*, 62, 404-414.
- Jin, J., & Dabbish, L. A. (2009, April). Self-interruption on the computer: a typology of discretionary task interleaving. In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 1799-1808). ACM.
- Kalnikaite, V., Sellen, A., Whittaker, S., & Kirk, D. (2010, April). Now let me see where i was: understanding how lifelogs mediate memory. In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 2045-2054). ACM.
- Kersten, M., Elves, R., & Murphy, G. C. (2006). WYSIWYN: Using task focus to ease collaboration. Supporting the social side of large scale software development, 19.
- Kopp, J. (1988, December). Self-monitoring: A literature review of research and practice. In *Social Work Research and Abstracts* (Vol. 24, No. 4, pp. 8-20). Oxford University Press.
- Korotitsch, W. J., & Nelson-Gray, R. O. (1999). An overview of self-monitoring research in assessment and treatment. *Psychological Assessment*, 11(4), 415.
- Lamming, M. G., & Newman, W. M. (1992, February). Activity-based Information Retrieval: Technology in Support of Personal Memory. In IFIP Congress (3) (Vol. 14, pp. 68-81).
- Lamming, M., & Flynn, M. (1994, February). Forget-me-not: Intimate computing in support of human memory. In Proc. FRIEND21, 1994 Int. Symp. on Next Generation Human Interface (p. 4).
- Large, A., Beheshti, J., Breuleux, A., & Renaud, A. (1995). Multimedia and comprehension: The relationship among text, animation, and captions. *Journal of the American society for information science*, 46(5), 340-347.
- Levy, E. C., Rafaeli, S., & Ariel, Y. (2016). The effect of online interruptions on the quality of cognitive performance. *Telematics and Informatics*, 33(4), 1014-1021.
- Li, I., Dey, A., & Forlizzi, J. (2010, April). A stage-based model of personal informatics systems. In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 557-566). ACM.

- Li, I., Dey, A. K., & Forlizzi, J. (2011, September). Understanding my data, myself: supporting self-reflection with ubicomp technologies. In Proceedings of the 13th international conference on Ubiquitous computing (pp. 405-414). ACM.
- Luo, Y., Lee, B., Wohn, D. Y., Rebar, A. L., Conroy, D. E., & Choe, E. K. (2018, April). Time for break: Understanding information workers' sedentary behavior through a break prompting system. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (p. 127). ACM.
- Mark, G., Gonzalez, V. M., & Harris, J. (2005, April). No task left behind?: examining the nature of fragmented work. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 321-330). ACM.
- Mayer, M., & Bederson, B. B. (2003). Browsing icons: a task-based approach for a visual Web history.
- McFarlane, D. C. (2002). Comparison of four primary methods for coordinating the interruption of people in human-computer interaction. *Human-Computer Interaction*, 17(1), 63-139.
- Miyata, Y., & Norman, D. A. (1986). Psychological issues in support of multiple activities. *User centered system design: New perspectives on human-computer interaction*, 265-284.
- Nahum, M., Nelken, I., & Ahissar, M. (2008). Low-level information and high-level perception: the case of speech in noise. *PLoS biology*, 6(5), e126.
- Parnin, C., & DeLine, R. (2010, April). Evaluating cues for resuming interrupted programming tasks. In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 93-102). ACM.
- Parnin, C., & Rugaber, S. (2011). Resumption strategies for interrupted programming tasks. *Software Quality Journal*, 19(1), 5-34.
- Parnin, C., & Rugaber, S. (2012, June). Programmer information needs after memory failure. In 2012 20th IEEE International Conference on Program Comprehension (ICPC) (pp. 123-132). IEEE.
- Prochaska, J. O., & Velicer, W. F. (1997). The transtheoretical model of health behavior change. *American journal of health promotion*, 12(1), 38-48.
- Quantified Self Homepage. (July). Retrieved from <https://quantifiedself.com>
- RescueTime: Time management software for staying productive and happy in the modern workplace. (n.d.). Retrieved from <https://www.rescuetime.com/>
- Rooksby, J., Rost, M., Morrison, A., & Chalmers, M. (2014, April). Personal tracking as lived informatics. In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 1163-1172). ACM.
- Rule, A., Tabard, A., Boyd, K., & Hollan, J. (2015, July). Restoring the context of interrupted work with desktop thumbnails. In 37th Annual Meeting of the Cognitive Science Society.
- Rule, A., Tabard, A., & Hollan, J. (2017). Using visual histories to reconstruct the mental context of suspended activities. *Human-Computer Interaction*, 32(5-6), 511-558.
- Salvucci, D. D. (2010, April). On reconstruction of task context after interruption. In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 89-92). ACM.
- Schroeder, J., Chung, C. F., Epstein, D. A., Karkar, R., Parsons, A., Murinova, N., ... & Munson, S. A. (2018, June). Examining self-tracking by people with migraine: goals, needs, and opportunities in a chronic health condition. In Proceedings of the 2018 on Designing Interactive Systems Conference 2018 (pp. 135-148). ACM.

- Sellen, A. J., Fogg, A., Aitken, M., Hodges, S., Rother, C., & Wood, K. (2007, April). Do life-logging technologies support memory for the past?: an experimental study using sensecam. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 81-90). ACM.
- The Center for the Healing Arts & Sciences. (2015, November 9). The Importance of Self Reflection. Retrieved from <http://www.thecenterforhas.com/the-importance-of-self-reflection/>
- Trafton, J. G., Altmann, E. M., Brock, D. P., & Mintz, F. E. (2003). Preparing to resume an interrupted task: Effects of prospective goal encoding and retrospective rehearsal. *International Journal of Human-Computer Studies*, 58(5), 583-603.
- Zeigarnik, B. (1938). On finished and unfinished tasks. *A source book of Gestalt psychology*, 1, 300-314.

## Appendix

### Appendix A - Pre-task survey

Q1 What is your age?

---

Q2 What is your gender?

- Male
- Female
- Other

Q3 What is the highest level of education you have completed?

- Less than High School
- High School graduate
- Some college
- 2-year degree
- 4-year degree
- Professional degree
- Doctorate

Q4 What would you rate your digital literacy?

	Poor	Below Average	Average	Above Average	Excellent
I would rate my digital literacy as:	<input type="radio"/>				

Q5 How often do you surf the Internet?

- More than once a day
- About once a day
- A few times a week
- A few times a month

Q6 What is your purpose of using the computer?

- Entertainment
- Studying
- Social Interaction
- Working
- Others \_\_\_\_\_

## Appendix B - Study Procedure

Unique Identifier of the participant (e.g. P01):

### STEP 0: Pre-study prep

- Print out one copy of consent form
- Make sure that the touch bar is changed.
- Bring pen and timer (stopwatch)
- Print one copy of this document
- Handout the specific page for the participant([link](#))
- Make sure that all the previous time lapse videos are stored in a separate folder. (However, you will need some example so that you have something to demonstrate with when you show time-lapse video editor) Make sure that demonstration video is same across the participants.
- Enter the spreadsheet with the participant information as much as possible ([link](#))

### STEP 1: Consent form

- READ the content aloud so that they can listen to it.
- Get a signature in the consent.

### STEP 2: Study Prep

- Bring the laptop.
- Start recording the screen with QuickTime Player (VERY IMPORTANT)
- Open the pre-task survey website. ([link](#))

### STEP 3: Pre-task survey:

- Read this: “Before starting the actual task, I would like you to fill out this survey.” (The survey should have been opened in the laptop already.)
- Make sure that the survey is submitted.

### STEP 4: Task Explanation

(Read this and give them a printed hand out that contains the information.)

You will be asked to surf online shopping websites specifically for purchasing new furniture.

Imagine that you are moving into the place that you are now living in and suppose that room and house is unfurnished so that you have to buy the whole set of furniture. Followings are the websites I will ask to use.

[www.wayfair.com](http://www.wayfair.com)

[www.overstock.com](http://www.overstock.com)

<https://www.target.com/c/furniture/-/N-5xtnr>

You will be asked to do this task twice, one for Bedroom and one for Living Room. For each task, use the following list. Do NOT choose non-furniture products e.g. pillow, bedding, slipcover, comforter, mattress, anything for decoration, cushion.

- Bedroom/Office (25 minutes)
  - Bed frame
  - Nightstand
  - Office desk
  - Office chair
  - Dresser / Wardrobes
  - Lamp
- Living Room (25 minutes)
  - Couch/Sofa
  - Coffee tables / Side Table
  - Media stand (TV)
  - Standing Lamp
  - Console/Sofa Tables
  - Recliner/Chair
  - Bookshelves
  - Lamp
- Notes.
  - Please choose at least THREE types of furniture per room for 25 minutes. (NO more than FIVE).
  - For each product type, browse until you know what you would like to purchase. Once you know what you would like to purchase, you can move on to the next furniture type (Bed → Nightstand). However, make sure that you compare your final choice with other products before you move on. You do not have to memorize your final choices or order them.
  - Check the details (dimensions, materials, colors, price, warranty, shipment date, rating/reviews, video) if that will fit in the place that you are living in and if you are willing to spend that money for the product. The purpose is to make this task as realistic as possible.
  - Note that the screen of the laptop (with sound) will be recorded to make sure that you do not enter any of your own information.

**STEP 5: [TASK 1]**

- Circle the condition and the task
 

(    )	Bedroom,	(    )	Living room
(    )	System,	(    )	Control
- If this is the system condition, please start the ScreenTrack (5 seconds interval, maximum width)
- “For the first task, please browse furniture for the [Bedroom, Living room]”
- Open a web browser and maximize the browser.
- “Okay, now you can start for (Bedroom, Living Room) Task.”
- Set the timer to 25 minutes.

## STEP 6: [Between TASK Prep]

- (Give the participant 5-minute notice) - “We are going to wrap-up the first task in five minutes.”
- (Tell them to stop) - “Okay, you can stop the task now.”
- If Task 1 was system condition, stop ScreenTrack Recording.
- Close all the browser and reopen a web browser

## STEP 7. [TASK 2]

- Circle the condition and the task. (Fill this out beforehand)  
(     )     Bedroom,                   (     )     Living room  
(     )     System,                   (     )     Control
- If this is the system condition, please start the ScreenTrack (5 seconds interval, maximum width)
- “For the second task, please shop furniture for the [Bedroom, Living room]”
- Open a web browser and maximize the browser.
- “Okay, now you can start for (Bedroom, Living Room) Task.”
- Set the timer to 25 minutes.

## STEP 8. [ End of Task 2]

- (Give the participant 5 minutes notice) - “We are going to wrap-up the second task in five minutes.”
- (Tell them to stop) - “Okay, you can stop the task now.”
- If Task 2 was system condition, stop ScreenTrack Recording.
- Close all the browser.

## Step 9. [Task 1 question]

- Q1: For task 1 - (Bedroom, Living Room) how many types of furniture have you chosen? What are they? (WITHOUT BROWSING THE BROWSER OR SCREENTRACK)
- (Tutorial for time-lapse video if TASK 1 was system condition)
  - This is a software that captures your screen every N second. It will let you go back and see what you did.
  - (Timeline) you can scrub the timeline to see the screenshots
  - (Five buttons) - Play - next, previous, - go to the beginning, go to the end.
  - (Speed) You can change speed while you are playing.
  - (Open URL) when you click this button, it will open the URL that you were using at the time the snapshot was taken.
  - Give them some time to interact with the time-lapse video during the tutorial. However, please make sure that they do not see their own videos.
  - Make sure that they used the slider (scrubbing the timeline)
  - So for you to get familiar with the software, I am going to ask you to do a simple task. The user in this video browsed Amazon.com and the user was going to purchase an office desk. They searched the desk, and browsed Amazon and

- picked one desk among them. Could you open the webpage that has the detail of the desk that this person chose, using the time-lapse video UI and open button?
- Q2: For [product 1], could you tell me the detail of the product 1 as much as possible (name of the product, how much it was, what color it was, the style of it, if the review was great, etc)? (WITHOUT BROWSING THE BROWSER OR SCREENTRACK)
- (Open Web Browser or ScreenTrack)
- Q3-1 (ScreenTrack): Using the time-lapse video, could you open the webpage that has the details of the product that you chose?
- Q3-2 (Chrome): Could you open the product webpage that has the details of the product that you chose? Feel free to use browser history if you would like
- (DO NOT INTERRUPT THEM UNTIL THEY OPEN THE WEBPAGE)
- Close the browser (or ScreenTrack)
- Repeat Q2, Q3 for product 2 and product 3.

### Step 10. [Task 2 question]

- Close the browser.
- Q1: For task 2 - (Bedroom, Living Room) how many types of furniture have you chosen? What are they? (WITHOUT BROWSING THE BROWSER OR SCREENTRACK)
- (Tutorial for time-lapse video if TASK 1 was system condition)
  - This is a software that captures your screen every N second. It will let you go back and see what you did.
  - (Timeline) you can scrub the timeline to see the screenshots
  - (Five buttons) - Play - next, previous, - go to the beginning, go to the end.
  - (Speed) You can change speed while you are playing.
  - (Open URL) when you click this button, it will open the URL that you were using at the time the snapshot was taken.
  - Give them some time to interact with the time-lapse video during the tutorial. However, please make sure that they do not see their own videos.
  - Make sure that they used the slider (scrubbing the timeline)
  - So for you to get familiar with the software, I am going to ask you to do a simple task. The user in this video browsed Amazon.com and the user was going to purchase an office desk. They searched the desk, and browsed Amazon and picked one desk among them. Could you open the webpage that has the detail of the desk that this person chose, using the time-lapse video UI and open button?
- Q2: For [product 1], could you tell me the detail of the product 1 as much as possible (name of the product, how much it was, what color it was, the style of it, if the review was great, etc)? (WITHOUT BROWSING THE BROWSER OR SCREENTRACK)
- (Open Web Browser or ScreenTrack)
- (Say the following if TASK 1 was system condition)
  - THIS TIME, I are going to ask you to use the browser to do the same task.
  - Feel free to use the browser history if you would like to.
- Q3-1 (ScreenTrack): Using the time-lapse video, could you open the webpage that has the details of the product that you chose?
- Q3-2 (Chrome): Could you open the product webpage that has the details of the product that you chose? Feel free to use browser history if you would like
- (DO NOT INTERRUPT THEM UNTIL THEY OPEN THE WEBPAGE)

- Close the browser
- Repeat Q2, Q3 for product 2 and product 3.

### **Step 11. Follow-up questions.**

- Ask the following questions.
  - Q1: What did you like about the system? what did you not like about the system?
  - Q2: Would you like to use this kind of software (or something similar) in your computer?
    - If yes, how would you use it? In which situation (tasks), would you use such software? (other than the online shopping I used today)
    - If no, why makes you not to use such software? (not useful? concerned?)
  - Q3: Do you have any concern in using the system?
  - Q4: Do you have any suggestion on how I can improve the system?

### **Step 12. Compensation**

- Make sure that you have the person's email.
- A \$15 e-gift card will be sent to your email within a week.

### **Step 13. After the study**

- UPLOAD the videos.
- Inform the professor for the compensation
- Measure timestamps and enter into the spreadsheet.