

An Akashic Desktop Recorder for Resumption of Works

Tatsuo Matsuura, Tadachika Ozono, and Toramatsu Shintani
Department of Computer Science,
Graduate School of Engineering, Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555 Japan
Email: {tatsuom,ozono,tora}@toralab.org

Abstract—In this study, we developed a system that records the PC user’s past work and enables the user to resume a previous task. Previously, we used screenshots to record work processes. However, this method demands a huge storage capacity. Therefore, in the proposed system, duplicate screenshots are deleted through similar-image determination. In addition, screenshots are transformed into a compressed moving image by introducing the H.264 video coding standard. Moreover, we integrate a backup system into our system to help users reach files based on their queries. The information on the user’s past work is organized in Research Episode Networks, which are graphs representing relationships among events, tasks, and files. Research Episode Networks make it easier for users to search for target files.

I. INTRODUCTION

The present study aims to support users to recall and fall back onto their past work. PC user’s work efficiency is significantly improved through knowledge of work performed in the past. For example, users refer to their previous work when they report on a weekly report, and then, resume their current tasks. However, with time, users forget their past work and must use applications to retrace this work. For example, web browsers record browsing histories and text editing applications record editing history. However, users have trouble looking back on past work because most applications record extensive work history as plain texts only. In addition, users often use multiple applications in parallel on a PC. Existing applications cannot record a whole work process because existing applications do not record application-to-application history.

In a previous study, we recorded user’s past work on PCs by using screen capture[1]. Then, the users could browse screenshots to recall their past work. However, they could not fall back onto their work. Indeed, the implemented recording system maintained a regular work history on the desktop as plain texts and screenshots. The users could look back on the history by browsing the screenshots. However, the required desktop storage capacity for the screenshots was huge, making this approach impractical. To resolve this problem, we propose a filter that will delete screenshots through similar-image determination. In this method, the screenshots are compared and same images are discarded as useless. In addition, our system creates moving images from the screenshots to reduce the image data size. We need to evaluate storage capacity for the screenshots to actually use this system for a long time.

Moreover, in a previous study, we had difficulty searching the work history because we recorded them over long periods. This system cooperates with existing applications and backup

systems, such as the Time Machine backup while Research Episode Networks (RENs) are used to support the search of the work history. These include graphs representing relationships among events, tasks, and files.

In this paper, we describe the Akashic Desktop Recorder, a system that supports user’s work history and backup. The remainder of this paper is organized as follows. Two types of work are examined (Section II). We discuss recall of work and analysis of active history (Section III). We describe details about how to record and resume a user’s past work (Section IV). Furthermore, we explain the system’s architecture and provide an example of execution (Section V). An evaluation experiment is then performed to confirm the required storage capacity of this system (Sections VI and VII). Finally, we conclude our paper (Section VIII).

II. WORK TYPE

There are two types of work performed on PCs: regular work and irregular work. Here, we describe the trend in each type of work and the relative approach employed in this study.

A. Regular Work

Regular work involves tasks performed at regular intervals, for example, preparing a weekly report. Users can efficiently perform similar tasks by referring to their previous work. However, they cannot recall a specific task because they are likely to use similar filenames for regular works. Confirming the web pages users browsed and the files they edited at that time can be an effective strategy. Most existing applications record work history. However, users find recalling their past work difficult because most application records are in the form of plain texts. Therefore, in this study, we maintain a record not only text format but also in the form of images. In our method, users can observe their past work visually.

B. Irregular Work

These tasks are often performed by users during their free time; thus, they are irregular. When users need to perform a similar action in the future, they refer to their past work to make a quick response work. However, there is a high possibility of completely missing records of irregular work, users forget where, when, and what they did a long time ago. In this case, we provide them with a visual record of their past work environment in addition to the text record.

Depending on the specific type of work, different approaches are implemented to aid in recalling and resuming past work.

In this study, we record user’s work history as plain texts and screenshots of the entire desktop. Then, the users can recall their past work by browsing the screenshots. Moreover, this system provides old files related to the task that can be resumed.

III. RELATED WORK

A. Recall of Work

The human brain includes three types of memory: sensory memory, short-term memory, and long-term memory. Sensory memory retains information for a few seconds, whereas information stored in the short-term memory is gradually forgotten with time. However, memories can be maintained in the human brain through repeated exposure to the relevant stimuli. Long-term memory has a large capacity for retaining information for a very long time. Nevertheless, based on the retrieval-failure theory, humans may forget the target information even after maintenance.

Therefore, it is important to determine the method suitable for remembrance in each case. Brunye et al.[2] conducted experimental tests associated with memory. They assigned the participants specific tasks associated with memory and instructions presented in three different formats: only visual, only textual, and in multimedia formats. The results indicated that multimedia has a distinctive advantage for memory preservation, for the following reason. Images provide spatial information, whereas texts facilitate consolidating the respective time series in the memory. Therefore, a combination of image and text data provides optimum results.

Blanc-Brude et al.[3] discussed the recall process of their own documents on a PC. These author’s goal was to identify the attributes and characteristics of the process. To this end, they conducted an experiment to find out what triggered the recall process, with regard to work documents.

B. Analysis of Active History

Gemmell et al.[4] developed MyLifeBits, an application that allows users to store information founded in PCs. MyLifeBits records archival material (e.g., articles and books) and activity history (e.g., keystrokes and mouse clicks). In addition, MyLifeBits can be used to add metadata to files.

In a previous study[5], the desktop activity history was recorded as a whole, regardless of the applications used. In fact, personal information, such as keystroke, mouse operation, screenshots, texts and window operation, was also recorded. Screenshots obtained at ten-second intervals for a month consume 62GB. Therefore, we need to develop a method for reducing storage capacity required.

Jensen et al.[6] used active history to support users in searching information. In particular, these authors analyzed provenance events to identify related documents. In the same study, users verified the effectiveness of using provenance events in recalling previous work on PCs.

Here, we present the Akashic Desktop Recorder, which records work history on PCs. The recorded work history

TABLE I. WORK HISTORY LIST

Type of Record	Information
Time	Acquisition Time
Application Information	Application Name
File Information	File Path, URL
Window Information	(X, Y, Width, Height)
Input Event	Mouth Event, Key Event
Work Annotation	Tag Name
Similarity	Hamming Distance

contains information associated with applications, files, and windows (e.g., an application name, and a file path). In addition, this system captures screenshots and compresses them. Users can browse the screenshots to recall their past work. Moreover, we integrate this system into a backup system to help users reach files they worked on in the past. Furthermore, this system uses RENs, which are graphs representing the relationships among events, tasks, and files, which enable the users to search the target files.

IV. VISUAL WORK HISTORY

The proposed system records useful screenshots and transforms them into a compressed moving image. Moreover, it is integrated into a backup system, such as the Time Machine on OS X, to refer to relative files used in the past. In addition, we use RENs to support users in searching specific files.

A. Screenshot Acquisition

The proposed system records the work process as plain texts and screenshots taken every ten seconds. Table I shows the type of record. However, the image storage capacity becomes bloated after recording screenshots for a long time. To solve this problem, our system deletes identical screenshots recorded continuously, which do not provide any additional information to the user. This method does not cause a lack of information because same screenshots are only deleted.

We chose to use the “Perceptual Hash” (pHash)[7] binary feature representation for each screenshot in order to detect duplicate images. pHash is computed as follows;

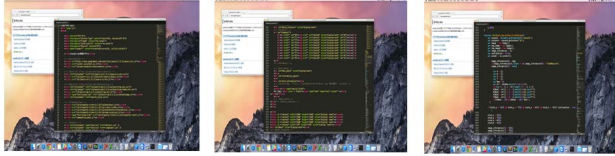
- 1) The screenshots are resized to 32×32 .
- 2) The screenshots are reduced to grayscale.
- 3) The DCT is computed and reduced.
- 4) From the original DCT(32×32), we maintain the top-left part (8×8). The 8×8 DCT represents the lowest frequencies in the screenshots.
- 5) The average value is computed from the 8×8 DCT low-frequency values.
- 6) We set the 64 hash bits to 0 or 1 depending on whether each of the 64 DCT values is above or below the average value.
- 7) The pHash is constructed.

This system compares screenshots of the active window operated by users because this changes frequently. The hamming distance is computed from the pHash of a previous screenshot.

TABLE II. WORK HISTORY

Time	Application Name	File Information	Window Information	Work Annotation
2016-03-05-11-24-40	Sublime Text2	/ESKM2016/main.tex	(613,241,1279,673)	Writing thesis
2016-03-05-11-24-50	Google Chrome	http://www-toralab.ics.nitech.ac.jp	(1098,158,1123,828)	Writing thesis
2016-03-05-11-25-00	Sublime Text2	/ESKM2016/main.tex	(613,241,1279,673)	Writing thesis
2016-03-05-11-25-10	Safari	https://www.npmjs.com	(1439,308,1083,652)	Programming
2016-03-05-11-25-20	Safari	https://www.npmjs.com	(1439,308,1083,652)	Programming

(A) < screenshots >



(B) < inter-frame prediction >

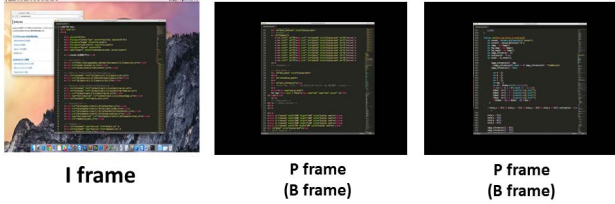


Fig. 1. Inter-frame prediction.

In this study, we use the hamming distance as the two methods. The first method deletes a screenshot that is identified as a duplicate. This system regards the screenshots as same images when the hamming distance equals 0. The second method shows work change by the hamming distance. This system calculates the *ChangeSimilarRate* from the hamming distance by using following equation.

$$ChangeSimilarRate = 1 - \frac{HammingDistance}{64} \quad (1)$$

The *ChangeSimilarRate* is calculated as rate of change of screenshots. Users can see the *ChangeSimilarRate* as rate of work change when they search screenshots. Users browse screenshots the *ChangeSimilarRate* shows large value when they want to understand an outline of past work.

B. Screenshot Compression

In order to reduce the required storage capacity for the several screenshots taken with this process, the proposed system transforms them into a compressed moving image by introducing the H.264 video coding standard[8]. H.264 incorporates an inter-frame prediction function and applies motion compensation. Inter-frame prediction involves anticipating the current frames from previous frames; the difference between the two frames is encoded. Inter-frame prediction involves three frames: I-frame, P-frame, and B-frame. The I-frame contains the image information, the P-frame has the differences with the previous I-frame, and the B-frame has the differences between the previous and the next frame.

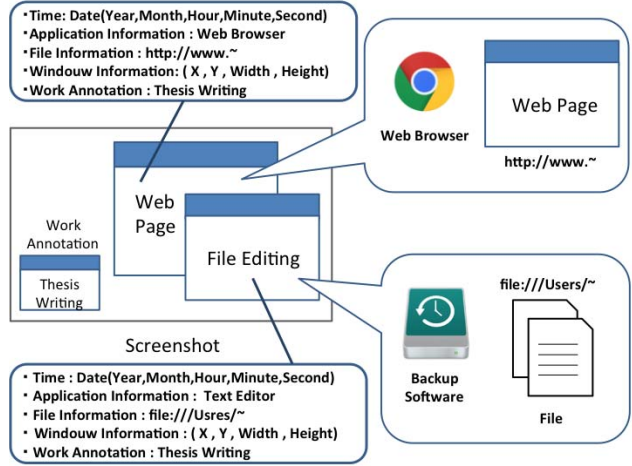


Fig. 2. Screenshot Information.

Moreover, the motion-compensated technique encodes part of the frames' movement.

Fig. 1 shows the three frames of a moving image, in which users open a web browser and use a text editor. Fig. 1(A) shows three frames that are not encoded. The movie size is large because it only contains the I-frame. Fig. 1(B) shows three frames that are encoded through inter-frame prediction and motion compensation. Because users mainly use a text editor, the second and third frames only contain images of the text editor. Therefore, the movie size is small because the three frames are P-frames or B-frames, which contain partial image information. In this study, there are similar screenshots because we record screenshots every ten seconds. Therefore, screenshots are transformed efficiently to a compressed moving image.

C. Collaboration with Existing Applications

When a user wants to resume a past work to facilitate file search, we combine our system with the Time Machine on OS X. The Time Machine saves all files as backup on a hard disk at hourly intervals and creates *Backups.backupdb* files in folders. It then identifies the changed files from the difference in each folder to reduce the required storage capacity. Our system refers to files on the Time Machine based on the acquisition time and the file information. Moreover, our system uses other applications such as Google Chrome in some cases.

Fig. 2 shows the screenshots in this system, which are associated with specific acquisition times, application information, file information, window information, and work annotation.

TABLE III. RESEARCH EPISODE NETWORKS

Time	Event	Deliverable	file1	file2
2012-04-02-10-32	ESKM2012	eskm2012.pdf	/ESKM2012/ESKM2012.tex	/ESKM2012/system2012.eps
2013-03-28-11-50	ESKM2013	eskm2013.pdf	/ESKM2013/main.tex	/ESKM2013/system.eps
2014-04-05-15-13	ESKM2014	eskm2014.pdf	/ESKM2014/eskm2014.tex	/ESKM2014/system.eps
2015-03-10-10-32	ESKM2015	eskm2015.pdf	/ESKM2015/ESKM2015.tex	/ESKM2015/eskmsystem.eps
2016-03-07-13-20	ESKM2016	eskm2016.pdf	/ESKM2016/main.tex	/ESKM2016/system.eps

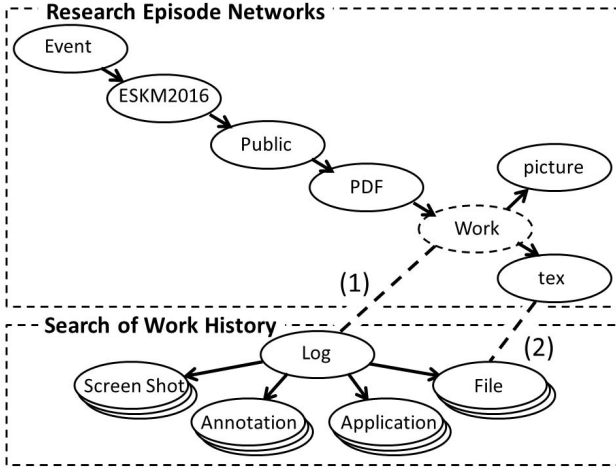


Fig. 3. Research Episode Networks.

Table II shows an example of a work history. This system creates an image map based on the window information. Then, it refers to the respective files in the Time Machine when the users click on the image map. When file information screenshots match files in the Time Machine, our system provides the target files to the users.

D. Research Episode Networks

We introduce RENs to save the relationship between screenshots and files[9]. RENs are graphs representing the relationships among PDF files of papers, presentation materials, and daily reports. In this study, the screenshots have links to files in the Time Machine in RENs. Associating the different information sets effectively helps remind the user of the different aspects of the past work.

In addition, it is important to record the current work process. Recording meta-information as work annotation in a natural language enables users to search for text. In this system, we can always input work in text windows. However, the work burden is large because users record work actively. Moreover, the timing when users input work is different from that when this system records screenshots, and the Time Machine creates backup files. For example, a work description is not always expressed in the screenshots. Here, we assume that users input work in real time.

This system produces RENs at each event. In Fig. 3, we assume that users search for files in ESKM2016. In this system, the “Log” which is the work history, is composed of four nodes: “Screen Shot”, “Annotation”, “Application”, and

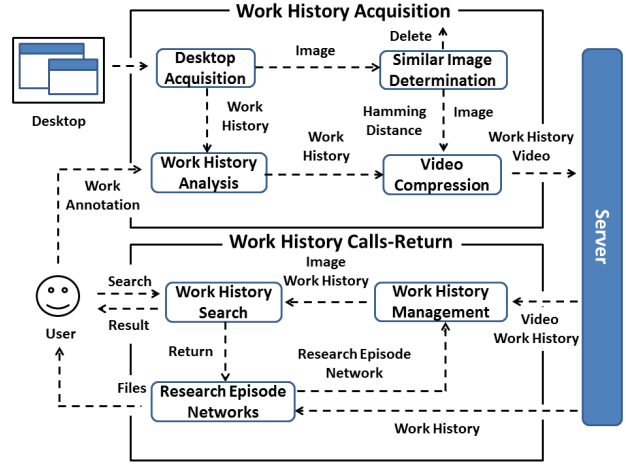


Fig. 4. System Architecture of the Akashic Desktop Recorder.

“Files”. The “Files” node includes the backup files stored in the Time Machine. In these RENs, Event is composed of several subnodes. Table III shows examples of “tex” and “picture”. The tex file includes a paper in ESKM2016. First, this system searches for “ESKM2016” to identify the respective RENs. Then, “Work” node is inserted into RENs of ESKM2016 and linked to the “Log”(1). Finally, the “Files” in the Time Machine are linked to the “tex”(2). This system searches for “tex” by tracing paths from ESKM2016 to the files.

V. AKASHIC DESKTOP RECORDER

A. System Architecture

We implement the proposed system as a web application; the server-side system is implemented on Node.js and the client-side system is implemented on HTML5, JavaScript, and Swift. In addition, we use FFmpeg library to transform the obtained screenshots into moving images. The system works with the Time Machine and RENs to refer to the user’s past files again.

Fig. 4 shows the architecture of the Akashic Desktop Recorder. It consists of two main subsystems: the Work History Acquisition system and the Work History Calls-Return system. The Work History Acquisition system has four modules: a desktop acquisition module, a similar image determination module, a work history analysis module, and a video compression module. The Work History Calls-Return system has three modules: a work history management module, a research episode networks module, and a work history search module.

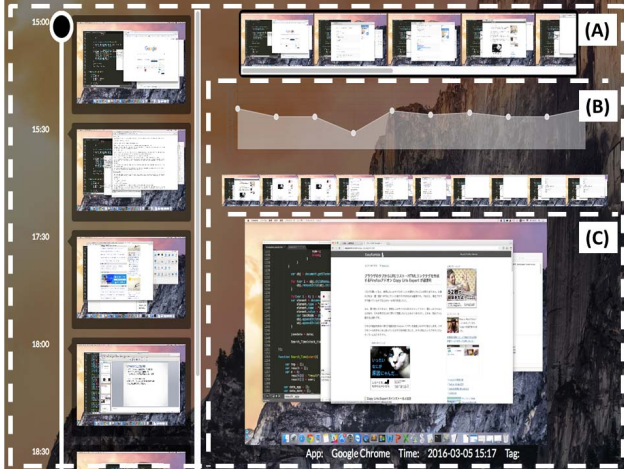


Fig. 5. Search Screenshots.

B. Work History Acquisition System

The Work History Acquisition system records screenshots and analyzes the user's work history. The desktop acquisition module records screenshots and the user's work history every ten seconds. The similar image determination module detects duplicate screenshots by calculating the hamming distance in perceptual hash (Section IV-A). The module compares each screenshot with the previous screenshot. The module deletes the duplicate screenshot and the respective work history. The work history analysis module obtains the work history, including the work annotation input by the users. The video compression module transforms the screenshots into a compressed moving image by introducing the H.264 video coding standard. The module forwards the work history and moving image to the server.

C. Work History Calls-Return System

The Work History Calls-Return system searches the work history and allows the user to resume past work. The work history management module obtains the moving image and the work history from the server. The module acquires a key frame as a screenshot from a video. Moreover, the module displays screenshots in chronological order, calculates the *ChangeSimilarRate* from the hamming distance (Section IV). The module calculates the *ChangeSimilarRate* as it changes in real time, and depicts it graphically. Therefore, the users can search the screenshots while referring to the graph. Moreover, the screenshots are related to the application information, acquisition time, file information and window information to create an interactive image map. Furthermore, the users can search the screenshots and work history in the work history search module, which is connected to the RENs. The research episode networks module creates RENs from the work history, which then collaborate with the Time Machine to allow the users to resume a past work.

D. Searching Work History

Fig. 5 shows an example of the Akashic Desktop Recorder. The users search and browse the screenshots to recall their

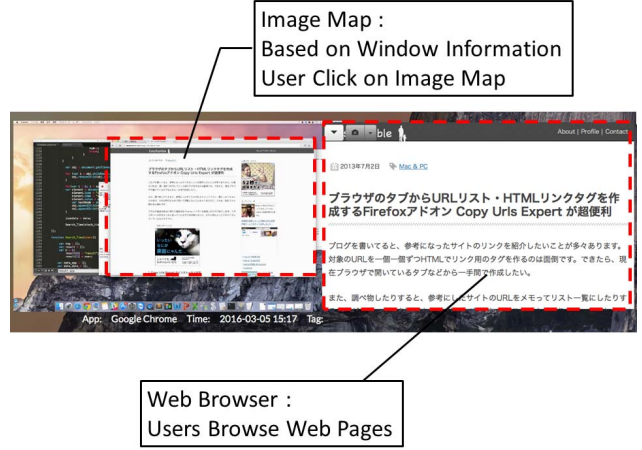


Fig. 6. Resuming Tasks on Image Map.

TABLE IV. STORAGE CAPACITY OF SCREENSHOTS

	Display	Images	JPEG	MP4
A	1,920 × 1,080	2,293	269.1MB	24.5MB
B	1,920 × 1,200	2,087	322.4MB	36.8MB
C	1,920 × 1,200	1,867	258.9MB	29.7MB
D	2,560 × 1,440	1,686	213.7MB	18.9MB
E	2,560 × 1,440	1,851	213.2MB	18.4MB

past work by entering a search query such as an acquisition date, an application name, or work annotations. Fig. 5(A) shows screenshots in chronological order. This system displays thumbnail images of screenshots every 30 min. When the users select a thumbnail image, the system displays the images in detail. Fig. 5(B) shows screenshots displayed along with a graph. The graph values change throughout the work process. The users can use this graph as a reference to browse the screenshots. When the graph indicates a significant change, the users can anticipate a respective change in the work process. Fig. 5(C) shows a selected screenshot. The system displays the screenshot, along with the acquisition time, application name, and work annotation.

Fig. 6 shows an example in which a user browsed a web page and wants to visit it again. The users browse the screenshots containing web pages (Fig. 6 Left), recalling that he/she originally browsed the same web page while working on the text editor. The web browser displays the web page (Fig. 6 Right) when the user clicks on the web browser image on the map. Then, the user can browse the web page again. In addition, the user confirms that he/she was working on the same files on the text editor by clicking the text editor image on the map.

VI. EVALUATION

We conducted an experiment to confirm the storage capacity of the Akashic Desktop Recorder. We recruited five students (A-E) in our laboratory, who use PCs for academic purposes.

The participants used this system for ten hours while working on a PC with no limitation on the work mode and speed. They were allowed, for example, to stop for a meal, coffee break, and visiting the bathroom. However, we did not count the time when the participants left their seats for more than 30 min.

They used the PCs of their laboratory, which had different screen resolutions (A: $1,920 \times 1,080$; B and C: $1,920 \times 1,440$; D and E: $2,560 \times 1,440$). We measured the number of screenshots obtained and the required MP4-format storage capacity in each case. Moreover, we compared the JPEG-format storage capacity with the MP4-format storage capacity.

Table IV shows the number of screenshots, the JPEG storage capacity, and the MP4 storage capacity. The theoretical number of screenshots recorded was 3,600. However, in practice, the average number of screenshots was 1,956.8. Based on these numbers, therefore, this system saved more than 46%. The mean storage capacity for JPEG images was 255.46 MB, and that for MP4 files was 25.66 MB. Thus, this system could reduce the required image capacity by about 90%.

VII. DISCUSSION

The experimental results demonstrate the effectiveness of the Akashic Desktop Recorder based on image compression. First, when the participants browsed a web page and viewed a pdf file, the desktops remained the same. Many screenshots were deleted through by similar image determination because identical screenshots were captured in our system. Thus, this system controlled the storage capacity bloating. However, the size of window and the type of work have effect on similar-image determination. In future studies, we also need to conduct an evaluation experiment of similar-image determination. In addition, the system compressed the screenshots by using the H.264 video coding standard, as described in Section IV-B. The changes in these images were negligible because the participants used multiple windows to work on the desktop. Therefore, this system can be operated for years by recording on an external hard disk drive. In future studies, we plan to use the next generation H.265 video coding standard to compress screenshots more efficiently.

The example of execution in Section IV showed the effectiveness of the Akashic Desktop Recorder based on the backup system and RENs. Usually, users remember the date and application name accurately for work tasks they repeat at regular intervals. Consequently, they can simply search the screenshots based on the acquisition time and application name. On the other hand, users have trouble searching target screenshots because they forget the details of work they perform irregularly. However, inputting work annotation while at work helps them search for the target screenshots in the future. Then, they can resume their past work simply by clicking on the respective screenshots on the image map. This system creates RENs for each event, which link the obtained screenshots to the files in the Time Machine. In this way, the users can search the target files visually by browsing the screenshots. In future studies, we also need to conduct an evaluation experiment to confirm whether this way is effective or not.

VIII. CONCLUSION

In this study, we present the theory behind the Akashic Desktop Recorder and its applications. This system records desktop computer work regularly in the form of plain texts and screenshots. Screenshots are transformed to a compressed moving image using the H.264 video coding standard. Furthermore, this system is connected to a backup system, such as the Time Machine, to allow the users resume their past work. Moreover, RENs correlate the obtained screenshots to enhance the work history and support users in searching through the screenshots.

The experiment and example of execution showed the effectiveness of the Akashic Desktop Recorder. This system significantly reduces the number of screenshots, through similar-image determination, by 46%. In addition, by transforming the screenshots into a moving image, it reduces the required storage capacity for the screenshots by 90%. Moreover, the users can resume their past work by using the Time Machine and RENs. In our future studies, we plan to conduct an experiment to confirm recollection and work return. However, we need to acquire more information on a long-term basis.

In our system, users search screenshots obtained within a few days. Searching a larger amount of data would be much more difficult. Therefore, we need to find a way of determining which information from within the user's work history record is indeed useful.

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI Grant Number 15K00422, 16K00420.

REFERENCES

- [1] T. Matsuura, Y. Niwa, T. Ozono, and T. Shintani, "Implementation of a Visual Work History Management System using Similarity of Screenshot," Proceedings of the 78th National Convention of IPSJ, Vol. 78, No. 4, pp.177-178, 2016.
- [2] T. Brunye, H. A. Taylor, D. N. Rapp, and A. B. Spiro, "Learning procedures: the role of working memory in multimedia learning experiences," APPLIED COGNITIVE PSYCHOLOGY, Vol. 20, pp.917-940, 2006.
- [3] T. Blanc-Brude and D. L. Scapin, "What Do People Recall about Their Documents?: Implications for Desktop Search Tools," In Proceedings of the 12th International Conference on Intelligent User Interfaces, pp.102-111, 2007.
- [4] J. Gemmell, G. Bell, and R. Lueder, "MyLifeBits: A personal database for everything," Comm. ACM, Vol. 49, No. 1, pp.89-95, 2006.
- [5] H. Kondo, H. Koide, and Y. Miyake, "The Basic Design for a System to Utilize the History of Activity and Findings to Develop it Progressively," IPSJ Journal, Vol. 48, No. 12, pp.19-27, 2007.
- [6] C. Jensen, H. Lonsdale, E. Wynn, J. Cao, M. Slater, and T. G. Dietterich, "The Life and Times of Files and Information: A Study of Desktop Provenance," In Proceedings of the 2010 ACM International Conference on Human Factors in Computing Systems, pp.767-776, 2010.
- [7] V. Bajaja, S. Keluskar, R. Jaisawala, and R. Sawant, "Plagiarism Detection of Images," International Journal of Innovative and Emerging Research in Engineering, Vol. 2, pp.140-144, 2015.
- [8] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, pp.560-576, 2003.
- [9] T. Iwasa, Y. Kato, S. Shiramatsu, T. Ozono, and T. Shintani, "Linked Data-based Slide Repository: The Episodic Slide Retrieval Using the Episodic Keyword Networks," Journal of Control Science and Engineering, Vol. 4, No. 1, pp. 36-49, 2016.