Thumbnail Summarization Techniques for Web Archives

Ahmed AlSum and Michael L. Nelson

Computer Science Department, Old Dominion University, Norfolk VA, USA
{aalsum,mln}@cs.odu.edu

Abstract. Thumbnails of archived web pages as they appear in common browsers such as Firefox or Chrome can be useful to convey the nature of a web page and how it has changed over time. However, creating thumbnails for all archived web pages is not feasible for large collections, both in terms of time to create the thumbnails and space to store them. Furthermore, at least for the purposes of initial exploration and collection understanding, people will likely only need a few dozen thumbnails and not thousands. In this paper, we develop different algorithms to optimize the thumbnail creation procedure for web archives based on information retrieval techniques. We study different features based on HTML text that correlate with changes in rendered thumbnails so we can know in advance which archived pages to use for thumbnails. We find that SimHash correlates with changes in the thumbnails ($\rho = 0.59, p < 0.005$). We propose different algorithms for thumbnail creation suitable for different applications, reducing the number of thumbnails to be generated to 9% – 27% of the total size.

1 Introduction

A thumbnail is a small image that represents a web page as it is rendered in a web browser such as Firefox or Chrome. Representing web pages with thumbnails has been used in various research such as: the visualization of the web search results [1–3], the visualization of recommended and similar pages such as SimilarWeb.com, and helping in revisitation and remembrance of the web page [4, 5]. Using thumbnails in web archives and temporal data has been studied [6, 7], but the creation cost of the thumbnails in the web archive has not yet been studied.

There are currently a few web archives that create thumbnails for mementos (archived versions of web pages). For example, the UK Web Archive and Archive.is provide a partial list of thumbnails per URI. Archive-It enables the partners to generate thumbnails for quality assurance purposes.

Figure 1 shows two examples of a TimeMap (a list of all the available mementos for a URI) for VisitWales.com. The HTML bubble interface in the Internet Archive[1] is in the back with the red border while in the front is the thumbnail view of the same URI in the UK Web Archive[2]. The thumbnail view gives the user an idea about the

[1] http://web.archive.org/web/*/http://visitwales.com/

[2] http://www.webarchive.org.uk/ukwa/target/56197146/source/subject

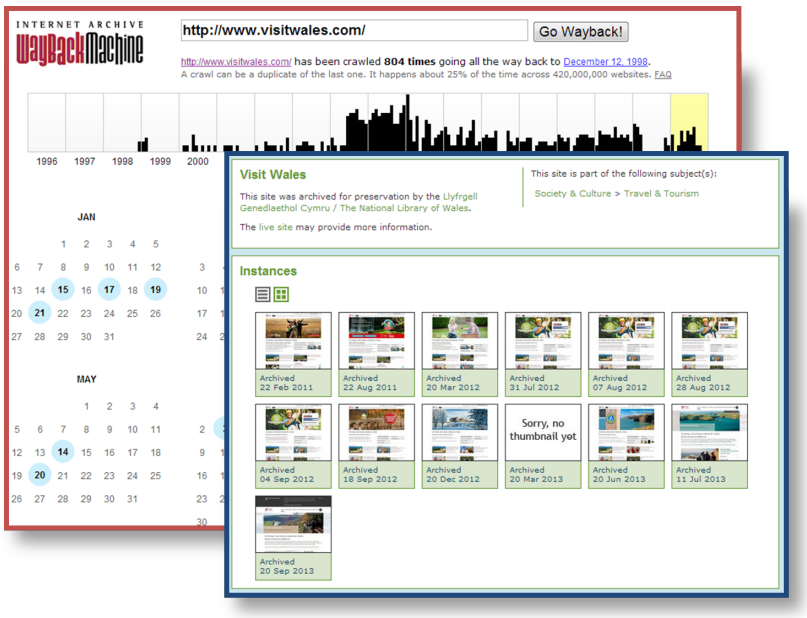


Fig. 1. Different representations for VisitWales.com TimeMap.

website layout and its evolution through time. Also, it helps the user in searching and revisitation for specific mementos of this URI. For example, `VisitWales.com` did not change from Jul 2012 to Sep 2012; the website layout was the same from Feb 2011 to Jun 2013 while it had a slight change for the rest of the timeline. This visual representation of the website through the time helps the user to select the desired mementos in the TimeMap. The thumbnail with the text "Sorry, no thumbnail yet" could have been generated for several reasons, but a likely explanation is that the HTML was not successfully processed by the thumbnail generator.

Creating the thumbnails in web archive has a number of challenges. First, scalability in time, the computation of the thumbnail requires the rendering of the web page including retrieving all the images, style sheets, and running any javascript. Second, scalability in space, the thumbnail (as metadata information about the web page) requires storage and our experiment shows that the average size for thumbnail with 64x64 pixels is 3KB and with 600x600 pixels is 133KB while the average size of HTML text is 39KB. Based on the Internet Archive estimated size of 240 billions mementos [8], they would need an additional 335 TB to store a 64x64 thumbnail for each memento, it will reach 14.5 PB to store a 600x600 thumbnail for each memento. Finally, page quality, the construction of the archived web page may not be successful on the time of building the thumbnail due to the missing embedded resources.

Using the thumbnails also has some limitations. Some TimeMaps have too many mementos for browsing. For example, the Internet Archive has over 17,000+ mementos for `cnn.com` over a 14 year span. So even if the Internet Archive has thumbnails for

all the mementos, some form of sampling or partitioning will be necessary because the cognitive load of processing all the mementos will be beyond what the user can handle [9]. In a personal digital library, Graham et al. [10] suggested 25 images per view panel because displaying all images did not help in conveying an overview of the album.

In this paper, we use the HTML text of the mementos and the crawler log information to predict the visual change in the URI through the time in order to select the significant set of mementos that can summarize the TimeMap. We explore various features that could be extracted from the HTML pages and select the most relevant to the visual effect, and then we propose different techniques for online and offline thumbnail creation.

The rest of this paper is organized as follows. Section 2 describes the related work. Section 3 describes the dataset that we use in our experiment. Section 4 explores the different features that extracted from HTML text or crawler log. Section 5 discusses the selection algorithms that is used to estimate the representative thumbnails for TimeMap. Section 6 concludes with a summary and future work for this study.

2 Related Work

Few web archives provide a thumbnail view for its content. The UK Web Archive uses thumbnails in timeline and 3D wall visual browsing [11]. Reed archives[3], National Taiwan University Web Archiving System (NTUWAS) [12], and Archive.is provide either a partial or full list of thumbnails for each memento. Archive-It generates on-demand thumbnails as part of the quality assurance process but it is only accessible for their partners. Web Archive Service at California Digital Library[4] uses thumbnails to represent the collections. The usage of thumbnails in the web archives has been studied in few applications. Soman et al. [13] developed ArcSpread which took a query from the user, extracted related information from the Stanford WebBase [14] using a Hadoop cluster, and displayed the results in spread sheet style. ArcSpread used both images and web page thumbnails to express the matched web pages. Jatowt et al. [15] proposed "*Past Web Browser*" to present TimeMap page snapshots as a slideshow of memento thumbnail. It has been extended to *Page History Explorer* [16] for visualizing and comparing the history of web pages. Zoetrope [7] provided a timeline visualization to show the duration and frequency of web pages on specific website. Padia et al. [6] implemented new visualization techniques for Archive-It collections entitled "*image plot with histogram*" in which they represented each page with its thumbnail.

Thumbnails help with the presentation of temporal data. Tsang et al. [17] proposed the concept of "*Temporal Thumbnail*" where they included the time dimension into the space of 3D model. They showed that the Temporal Thumbnails were effective for quickly analyzing large amounts of viewing data. Stoev and Straßer [18] studied the visualization of historical data in existence of time dimension. They studied two models; navigation with 4D, and navigation with one fixed dimension.

Thumbnails have applications on the web. `SimilarWeb.com` and `StumbleUpon.com` have recommendation pages function that uses the page thumbnails to represent

[3] `http://www.reedarchives.com/`
[4] `http://webarchives.cdlib.org/`

the recommended pages to help the user to determine the page relevancy. Janssen [19] used fixed thumbnail per document (called an Icon Document) to visualize the search results from UpLib digital library system[5]. Janssen studied the computation of the icon size and the decoration of the icon with labels (e.g., creation date).

The thumbnail can enhance the users' ability to find the information. Lam and Baudisch [20] proposed a "*Summary Thumbnail*" that generated customized thumbnail for web pages to increase the readability of the text on embedded/small screens devices. They discovered that the preservation of the page layout allowed the users to better detect useful information. Woodruff et al. [1] showed that using a mixture of textual and image representations of the web page increased the user prediction of the page effectiveness.

Aula et al. [21] compared thumbnail and textual summaries by surveying the users between different types of page previews. The study showed that the combination of textual information with the thumbnails increased the user estimation of the page content and usefulness. Also, they found that the recognition improved when the thumbnail size was 200x250 pixels. Kaasten et al. [4] and Teevan et al. [5] discovered that thumbnails of size 208x208 pixels or above are effective to remember an exact page.

The selection of the representative image from a set of images depends on comparison techniques between the images itself. These techniques are different from our proposed techniques as we depend on the HTML text to select the representative thumbnail image. AutoAlbum [22] clustered personal photographs into album using time-based and content-based clustering. Coelho and Ribeiro [23] used image filter to get an abstracted version of the images to reduce the dataset to 10% of the original size. Chu et al. [24] studied the near duplicate detection technique for images to cluster a set of images and select a representative image from them. Kherfi and Ziou [25] used the image clustering to organize a large set of images and to provide the user with an overview of the collection's content. They used probabilistic models based on predefined keywords or visual features. Graham et al. [10] studied a set of images that had an attached timestamp (e.g., images from digital camera). They provided various clustering and summarization techniques such as summarization by time and clustering based on the time and location. They developed a calendar browser with specific 25 photos per panel. They selected the images based on the number of images each month.

3 Dataset

We built a collection of TimeMaps using the homepage URIs for the companies listed in the 2013 Fortune 500 list[6]. For each URI, we retrieved the TimeMap from the Internet Archive Wayback Machine. Of those 500 URIs, we have 488 TimeMaps since 12 are not archived due to robots.txt exclusion. The total number of mementos was 499,540. The mean number of mementos per TimeMap is 1023 and the median is 685. For each memento, we downloaded the HTML text from IA and we used PhantomJS[7] to capture a screenshot for each memento.

[5] http://uplib.parc.com/

[6] http://money.cnn.com/magazines/fortune/fortune500/2013/full_list/

[7] http://phantomjs.org/

4 Exploration of Features

In this section, we explore various features that can be used to predict the change in the visual representation of the web page. The features could be obtained from the crawler log (e.g., CDX[8] file for Heritrix crawler) or from the HTML text for the memento.

4.1 Web page similarity features

There have been several methods proposed for calculating similarity between web pages [26–29]. In this section, we explore various features and techniques to estimate the resulting difference in thumbnail images based on the HTML source of the mementos.

SimHash Similarity SimHash [28] is a fingerprint technique to calculate the near-duplicates between web pages. We used 64-bit SimHash fingerprints with $k = 4$. We calculate the SimHash for different parts of the web page. First, we calculate SimHash for the full HTML text. Then, we use boilerpipe library [30] to extract different parts of the web page such as: the main content from the web page, all the text (even the template text), templates including the text, and the template excluding the text (just HTML structure). For each subsequent mementos in the TimeMap, we computed the Hamming Distance between their SimHash fingerprints.

Levenshtein Distance between HTML DOM Tree Each web page can be expressed in a DOM tree. We can compare between two web pages by calculating the difference between their DOM trees. In this feature, we transform each HTML page into a DOM tree (using jsoup[9]), then we calculate the Levenshtein distance between both trees by calculating the number of operations (insert, delete, and replace) to turn one tree into the other [31].

Embedded Resources The change in the embedded resources that construct the web page affects the visual appearance of the rendered web page. We extract the embedded resources for each memento and calculate the difference between each two pages. We calculate the total number of new resources that have been added and the resources that have been removed. Then, we divide the total number based on the embedded resource type (e.g., image, style sheet, javascript), so we have the specific number of addition and removal for each category. For example, the difference between M_{t1} and M_{t2} could be: addition of 5 resources (2 javascript files and 3 images) and removal of 2 resources (1 javascript and 1 image).

Memento Datetime Memento Datetime is the datetime that the memento was crawled (or observed) by the web archive. This information is already available in the CDX file. The similarity is calculated based on the difference in seconds, a low number indicates high similarity.

[8] `http://archive.org/web/researcher/cdx_file_format.php`
[9] `http://jsoup.org/`

4.2 Experiment

The success criteria for each feature is how well it can predict the visual difference between two mementos. First, we generate a thumbnail for each memento in each TimeMap. Then, we calculate the difference by comparing the number of different pixels between each two thumbnails using SciPy[10]. In order to compare two thumbnails, we resize them into different dimensions: 64x64, 128x128, 256x256, and 600x600. We calculated the Manhattan distance and Zero distance between each pair. Finally, we calculate the correlation between the similarity of the web pages (based on features in section 4.1) and the difference between the thumbnails.

4.3 Results

Figure 2 shows histogram of the correlation (using Pearson's ρ) (on x-axis) between some features and the image difference calculated by Manhattan distance using thumbnail size 600x600 and the number of TimeMap that achieved this level (on y-axis). Generally, all the features showed a positive relationship that range from weak to strong. The best correlation has been found between SimHash fingerprints for the original HTML text showed a positive correlation, 70% of the TimeMap has a correlation $\rho \geq 0.5$ (on average $\rho = 0.59, p < 0.005$). The Levenshtein distance between the HTML DOM tree has a high correlation on average ($\rho = 0.57, p < 0.005$). We use the SimHash similarity because the computation is faster than the HTML DOM tree distance.

Our results show interesting features. First, the calculation of the visual difference between the images is not affected by the thumbnail size. The results are consistent between the different thumbnails sizes. Second, we repeat the SimHash calculation on different parts of the web page, the SimHash similarity between the full HTML text shows the highest correlation over the rest of web page parts. Third, however the datetime and embedded resources show a weak positive correlation, however it can be used with other strong features to get a better prediction.

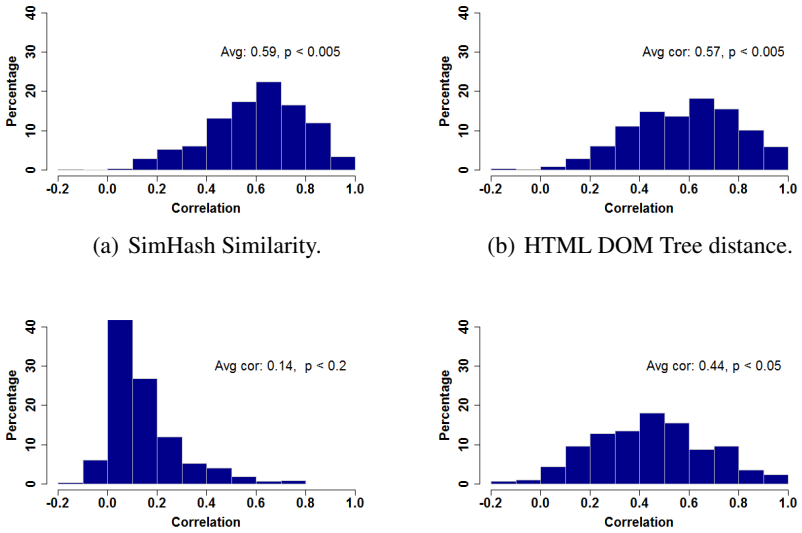
5 Selection Algorithms

In this section, we discuss three algorithms to select a list of representative thumbnails for individual TimeMaps.

5.1 Threshold Grouping

In this algorithm, we use feature f for TimeMap $TM = \{M(t_1), M(t_2), ..M(t_n)\}$. The initial step is to divide the TM into groups G, each group has only two subsequent mementos $M(t_i)$ and $M(t_{i-1})$. For each G, we compute the $diff(f)$ between $M(t_i)$ and $M(t_{i-1})$. If $diff(f, M(t_i), M(t_{i-1})) < \alpha$, we will eliminate one of the mementos M in the group. Then, the new list is sorted by time and we repeat the grouping again. We will continue the process until we reach that for every pair of mementos $diff(f, M(t_i), M(t_{i-1})) > \alpha$. Figure 3 shows the first step of the threshold grouping.

[10] `http://docs.scipy.org/doc/scipy/reference/index.html`



(a) SimHash Similarity. (b) HTML DOM Tree distance.

(c) Memento Datetime (d) Embedded Resources.

Fig. 2. Histogram for the correlation between Thumbnail difference and various features.

The value of α can be configured and it may depend on other factors. Figure 4 shows the relationship between the change in the SimHash threshold (on x-axis) and the reduction of the TimeMap and the loss of image difference (on y-axis). The image loss is calculated as the Manhattan distance between the selected thumbnail and the eliminated thumbnail. We defined the optimum point as the smallest TimeMap size with the least image difference loss. Our empirical study shows that for SimHash similarity, we find that $\alpha = 0.05$ is the optimum value where it decreases the TimeMap to 27% of its original size with the loss of 27% of the image differences.

We notice here that with SimHash threshold at 0, which means removes all the duplicate snapshots only, we still have loss in image differences. It happens because even if the HTML text may be the same, the rendered image may not be visually identical. The reason for that on the live web may be different advertisements, different results from javascript (e.g., picking a random image each time), etc. The web archive environment adds more causes such as: the embedded resources may not be archived, change of embedded resources (e.g., update in style sheet without changing the URI), or the web archive server can not render the page at that time (cf. "Sorry, no thumbnail yet" in figure 1).

5.2 *K* Clustering

In this algorithm, the web archive will select K thumbnails for each TimeMap. K could be absolute, relative, or an expression based on other parameters (e.g., rate of change, crawling frequency). The algorithm will depend on the a set of features F. To get the

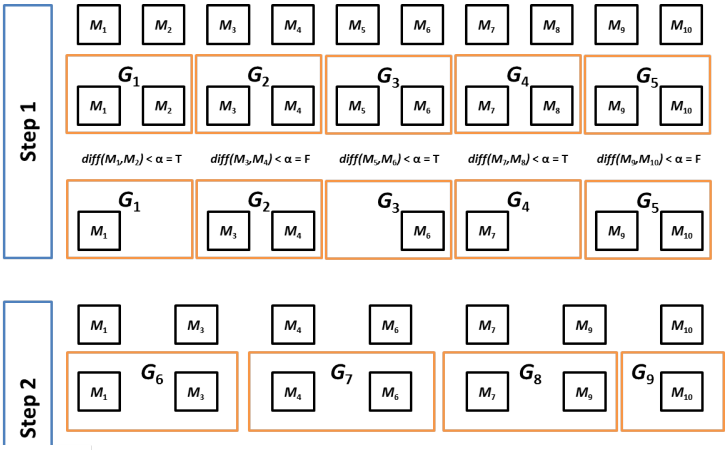


Fig. 3. Threshold Grouping algorithm.

most representative K, we apply the "*K-Medoids*" [32] on each TimeMap to cluster the mementos into K clusters. For each cluster, we select random thumbnail to represent the cluster. We apply the algorithm on two sets of features, $F_1 = \{SimHash\}$ and $F_2 = \{SimHash, Memento - Datetime\}$, we repeat the algorithm on $K \in \{5 : 200\}$. Figure 5 shows the average sum square error and the average image loss in both cases. The red dots describes the average sum square error for each cluster, the red line is the power regression for the sum square error. The blue dots describes the average image loss, and the blue line is its power regression. The black line is a linear cost function based on the time taken to generate the K thumbnails. Notice that the increasing number of clusters decreases the error value but also it increases the cost of creating the thumbnails. Also, the sum square error and image loss error trends are the same which align with our results at section 4.3. Using two features (figure 5(b)) gives lower error rate in both sum square error and image loss which means a better representation of the TimeMap. Based on our empirical study, we find the optimal values for K = 25..40, which decreases the size of the average TimeMap by 9% to 12% of its original size.

5.3 Time Normalization

The drawback of the previous algorithms is that it did not take the time dimension into the consideration. In this algorithm, we will apply normalization per time for the TimeMap. We will divide the TimeMap into fixed time-slots T and select random k thumbnails from each slot. The advantage of this algorithm is that it is easy to implement because it depends only on the CDX files independent from the web pages itself. Figure 6 shows the reduction of each TimeMap after selecting $k = 1$ and $T = 1\ month$. It reduced the TimeMap size on average to 23% of its original size.

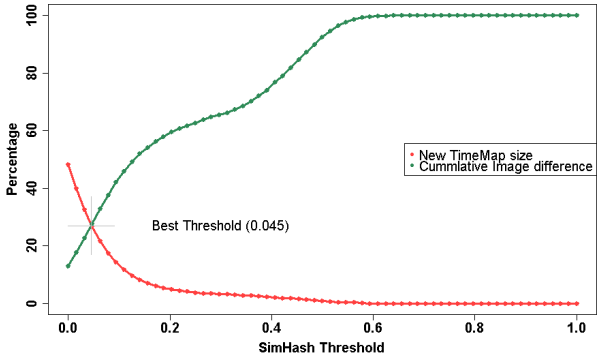
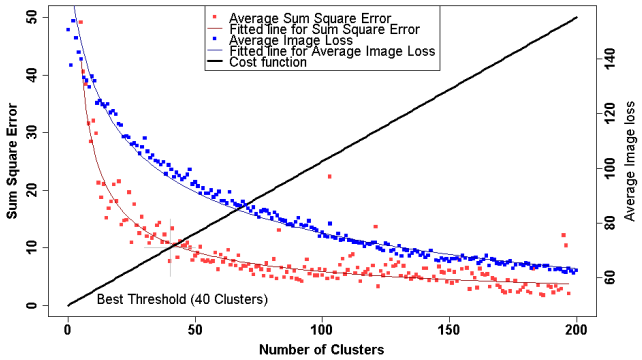
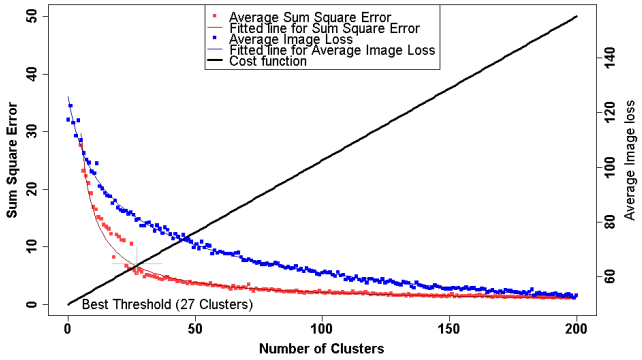


Fig. 4. Optimum SimHash threshold point for Threshold Grouping algorithm



(a) Best K using SimHash feature only.



(b) Best K using SimHash and Memento-Datetime features.

Fig. 5. Best K value for K Clustering algorithm.

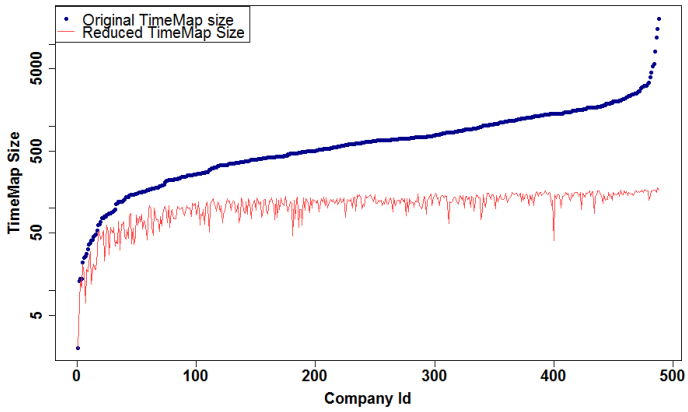


Fig. 6. Reduction of the TimeMap size after applying k thumbnail per time-slot algorithm

5.4 Analysis

Table 1 compares between the three algorithms. Even though the three of them could be used for online and offline processing, we suggest using Threshold grouping for offline because it generates the significant list of thumbnails that represents the TimeMap disregards the application itself. K clustering could be suitable for the online processing because it generates the requested K thumbnails from the application, even the extensive nature of clustering algorithms may affect the application performance. Comparing the average image loss of the selected thumbnails in the Time Normalization technique shows higher error rate than the other algorithms which means it gives a poor representation of the TimeMap, however it requires less computation.

Table 1. Comparison between the selection algorithms.

	Threshold Grouping	K clustering	Time Normalization
TimeMap Reduction	27%	9% to 12%	23%
Image Loss	28	78 - 101	109
# Features	1 feature	1 or more	1 feature
Preprocessing required	Yes	Yes	No
Efficient processing	Medium	Extensive	Light
Incremental	Yes	No	Yes
Online/offline	Both	Both	Both

6 Conclusions

We studied effective methods to generate thumbnails for the web archive corpus. We explored various similarity features, we found that SimHash and Levenshtein distance

between HTML DOM trees had $\rho = 0.59$ and $\rho = 0.57$ correlation with the visual difference between two mementos. We suggest using SimHash as it is efficiently computed from the HTML text. We proposed three algorithms to select K representative thumbnails from the TimeMap. The algorithms decreased the TimeMap size from 9% to 27% of its original size with the minimum loss in thumbnails differences. The techniques could be used with online and offline processing.

Our next step in this research will be to integrate an exploratory TimeMap service in ArcLink that will apply the algorithms defined above to selectively generate thumbnails for TimeMaps to facilitate collection understanding.

References

1. Woodruff, A., Faulring, A., Rosenholtz, R., Morrsion, J., Pirolli, P.: Using thumbnails to search the Web. In: Proceedings of the SIGCHI conference on Human factors in computing systems. CHI '01 (2001) 198–205
2. Kules, B., Wilson, M.L., Shneiderman, B.: From Keyword Search to Exploration: How Result Visualization Aids Discovery on the Web. Technical report, HCIL-2008-06 (2008)
3. Treharne, K., Powers, D.M.W.: Search Engine Result Visualisation: Challenges and Opportunities. In: Proceedings of 13th International Conference on Information Visualisation. (2009) 633–638
4. Kaasten, S., Greenberg, S., Edwards, C.: How People Recognise Previously Seen Web Pages from Titles, URLs and Thumbnails. In: People and Computers XVI - Memorable Yet Invisible SE. Number January. Springer London (2002) 247–265
5. Teevan, J., Cutrell, E., Fisher, D., Drucker, S.M., Ramos, G., André, P., Hu, C.: Visual Snippets: Summarizing Web Pages for Search and Revisitation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '09, ACM (2009) 2023–2032
6. Padia, K., AlNoamany, Y., Weigle, M.C.: Visualizing digital collections at Archive-It. In: Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries. JCDL 12 (2012) 15–18
7. Adar, E., Dontcheva, M., Fogarty, J., Weld, D.S.: Zoetrope: interacting with the ephemeral web. In: Proceedings of the 21st annual ACM symposium on User interface software and technology. UIST '08 (2008) 239–248
8. AlSum, A., Nelson, M.L.: ArcLink: Optimization Techniques to Build and Retrieve the Temporal Web Graph. Technical report, arXiv: 1305.5959 (2013)
9. Mayer, R.E., Moreno, R.: Nine ways to reduce cognitive load in multimedia learning. Educational psychologist **38**(1) (2003) 43–52
10. Graham, A., Garcia-Molina, H., Paepcke, A., Winograd, T.: Time as essence for photo browsing through personal digital libraries. In: Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries. JCDL '02 (2002) 326–335
11. Hockx-Yu, H.: The Past Issue of the Web. In: Proceedings of 3rd International Conference on Web Science. WebSci '11 (2011) 1–8
12. Chen, K., Chen, Y., Ting, P.: Developing National Taiwan University Web Archiving System. In: Proceedings of 8th International Web Archiving Workshop. IWAW '08 (2008)
13. Soman, S., Chhajta, A., Bonomo, A., Paepcke, A.: ArcSpread for Analyzing Web Archives. Technical report, Stanford InfoLab (2012)
14. Cho, J., Garcia-Molina, H., Haveliwala, T., Lam, W., Paepcke, A., Raghavan, S., Wesley, G.: Stanford WebBase Components and Applications. ACM Transactions on Internet Technology **6**(2) (2006)

15. Jatowt, A., Kawai, Y., Nakamura, S., Kidawara, Y., Tanaka, K.: Journey to the past: proposal of a framework for past web browser. In: Proceedings of the 17th conference on Hypertext and hypermedia. HYPERTEXT '06, ACM (2006) 135–144
16. Jatowt, A., Kawai, Y., Tanaka, K.: Page History Explorer: Visualizing and Comparing Page Histories. IEICE Transactions on Information and Systems **E94-D**(3) (2011) 564–577
17. Tsang, M., Morris, N., Balakrishnan, R.: Temporal Thumbnails: rapid visualization of time-based viewing data. In: Proceedings of the working conference on Advanced visual interfaces. AVI '04 (2004) 175–178
18. Stoev, S.L., Straßer, W.: A case study on interactive exploration and guidance aids for visualizing historical data. In: Proceedings of the conference on Visualization. VIS '01 (2001) 485–488
19. Janssen, W.C.: Document Icons and Page Thumbnails: Issues in Construction of Document Thumbnails for Page-Image Digital Libraries. In: Proceedings of 8th European Conference on Digital Libraries. ECDL (2004) 111–121
20. Lam, H., Baudisch, P.: Summary thumbnails: Readable Overviews for Small Screen Web Browsers. In: Proceedings of the SIGCHI conference on Human factors in computing systems. CHI '05 (2005) 681–690
21. Aula, A., Khan, R.M., Guan, Z., Fontes, P., Hong, P.: A comparison of visual and textual page previews in judging the helpfulness of web pages. In: Proceedings of the 19th international conference on World wide web. WWW '10, ACM Press (2010) 51–59
22. Platt, J.C.: AutoAlbum: clustering digital photographs using probabilistic model merging. In: Proceedings of IEEE Workshop on Content-based Access of Image and Video Libraries. (2000) 96–100
23. Coelho, F., Ribeiro, C.: Image abstraction in crossmedia retrieval for text illustration. In: Proceedings of the 34th European conference on Advances in Information Retrieval. ECIR' 12 (2012) 329–339
24. Chu, W.T., Lin, C.H.: Automatic selection of representative photo and smart thumbnailing using near-duplicate detection. In: Proceeding of the 16th ACM international conference on Multimedia. MM '08 (October 2008) 829–832
25. Kherfi, M.L., Ziou, D.: Image Collection Organization and Its Application to Indexing, Browsing, Summarization, and Semantic Retrieval. IEEE Transactions on Multimedia **9**(4) (2007) 893–900
26. Henzinger, M.: Finding near-duplicate web pages. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '06 (2006) 284–291
27. Broder, A., Glassman, S.: Syntactic clustering of the web. Computer Networks and ISDN Systems **29**(8-13) (1997)
28. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing. STOC '02 (2002) 380–388
29. Manku, G.S., Jain, A., Das Sarma, A.: Detecting near-duplicates for web crawling. In: Proceedings of the 16th international conference on World Wide Web. WWW '07 (2007) 141–149
30. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: Proceedings of the third ACM international conference on Web search and data mining. WSDM '10 (2010) 441–450
31. Pawlik, M., Augsten, N.: RTED: a robust algorithm for the tree edit distance. Proceedings of the VLDB Endowment **5**(4) (December 2011) 334–345
32. Park, H.S., Jun, C.H.: A simple and fast algorithm for K-medoids clustering. Expert Systems with Applications **36**(2, Part 2) (March 2009) 3336–3341