

A Daily Life Visualization by Representative Web Page Selection and Magazine-Style Layout

Chica Matsueda¹⁾ Takayuki Itoh¹⁾

1) The Graduate School of Humanities and Sciences, Ochanomizu University

{coco, itot} (at) itolab.is.ocha.ac.jp

Abstract

Web browsing history is a fruitful life log representing behavior and knowledge of users. Visualization of summarized browsing history can contribute to represent these information. This paper proposes a system to extract representative Web pages from browsing history of users, and arranges them like a newspaper, called HistoryPaper. This system firstly clusters browsed Web pages based on their contents. Then, it calculates priority of Web pages from search terms and rarity of accessed Web pages, and selects the representative page of each cluster. HistoryPaper arranges the representative pages by a layout algorithm following the magazine style applied in many news sites.

1 Introduction

Web browsing history contains fruitful information related to behavior and knowledge of users. Visualization of Web browsing history can contribute to the reflections of actions of particular days or previously studied knowledge of the users.

We suppose Web browsing history has both advantages of unconsciously and consciously recorded life logs. Life log is a popular word in the recent information technology field due to the growth of wearable devices. Sensing data including position and acceleration are recorded as unconscious life logs, while users consciously record other kinds of life logs such as dailies. Unconscious life logs can be continuously recorded without any labor of users; however, limited types of information can be stored due to functionality of the devices or equipment. On the other hand, recording conscious life logs requires continuous labor. Web browsing history is a convenient life log, since we do not consciously record, even though it contains fruitful information related to behavior and knowledge of the users.

This paper presents “HistoryPaper”, a system

which displays sets of Web pages corresponding to the summaries of browsing history of particular days. HistoryPaper consists of the following two components. Firstly, it selects representative Web pages in the browsing history of a day based on the numbers of accesses and query words. It then displays the selected Web pages applying magazine-style layout.

2 Related Work

Web page selection has been an active topic since it is useful for summarization and classification of Web sites. Many general purpose Web page selection techniques apply link structures [1] or content similarity [5]. On the other hand, small number of studies on Web page selection from personal browsing histories have been presented.

Web page layout is another key component of HistoryPaper. Several experimental studies on document layout, not only for Web pages but also other media such as newspapers [12], demonstrate that document layout is critical for behavior of readers. Various rectangle packing tech-

niques have been applied for layout of Web pages or other information. Squarified Treemap [2] is one of the most popular techniques which divides given rectangular spaces into particular number of well-shaped small rectangles, and actually applied to many applications on Web page layout [10]. Also, there have been several techniques on layout of multiple Web pages applying optimization techniques [3] [9] [11]. Meanwhile, there have been few studies on magazine-style-specific layout for multiple Web pages. Kuhna et al. presented a semi-automated magazine layout [4]; however, it places pieces of rectangular contents based on image features, not importance and aspect ratios which HistoryPaper applies.

There have been several services which display representative Web pages related to the interest of the users. Paper.li (<http://paper.li/>) is one of the typical services, which displays sets of Web pages shared on social networking services such as Twitter or Facebook. However, Paper.li aims the all-in-one display of already shared Web pages, while we focus on summarization of browsing history and real-time layout of Web pages with magazine-style design.

3 Web Page Selection

HistoryPaper automatically selects multiple representative Web pages browsed on a particular day, and displays in a Web browser window. HistoryPaper firstly calculates importance of Web pages browsed in a particular day. There have been several studies on selection of appropriate number of representative text to summarize the large collections of documents. Lin et al. [6] presented that tf-idf based representative text selection brings satisfactory document summarization results. We learned this knowledge to develop an algorithm for representative Web page selection. The Web page selection technique implemented for HistoryPaper consists of the following processes:

Process 1-1 firstly applies a clustering algorithm to the Web pages based on the similarity of their contents.

Process 1-2 then selects most important Web pages from each of the clusters.

Process 1-3 finally calculates the importances of clusters as sums of importance of Web pages.

3.1 Clustering of Browsed Web Pages

Process 1-1 firstly divides all the Web pages browsed on a particular day into n clusters. It specifies n (between 6 and 18 in our implementation) based on the size of a Web browser window and the number of browsed pages on the day. We calculate n by the following equation:

$$n = \max(\lceil 18w/a_w \rceil, 6). \quad (1)$$

Here, w is the number of pixels of the Web browser window, a_w is the total number of pixels of the display, $\lceil * \rceil$ is the Gauss's notation, and $\max(a, b)$ is an operator to select the larger value.

The following is the brief description of Process 1-1.

1. Convert the contents of Web pages into Bag-of-Words vectors.
2. Apply a dimension reduction to the Bag-of-Words vectors. Our implementation applies Latent Semantic Analysis (LSA) for the dimension reduction.
3. Apply k-means clustering method to the dimension-reduced vectors.

3.2 Importance Estimation for Web Pages

Process 1-2 calculates the importance of the Web pages p_p by applying the following equation:

$$p_p = \frac{m + 1}{q}. \quad (2)$$

Here,

- $m = m_{wm}/m_{max}$ is the ratio of search keyword frequency, where m_{wm} is the total number of keywords used for search query in the day and contained in the current Web page, and m_{max} is the maximum value of m_{wm} .

- $q = q_{wp}/q_{all}$ is the valuableness of the current Web page, where q_{wp} is the number of accesses to the current Web pages in the day, and q_{all} is the total number of accesses to the all Web pages in the day.

This definition is based on our heuristics that Web pages are important if they satisfy one or more of the following conditions:

- they are browsed over and over in the day, even though they are not usually well browsed.
- they contain many keywords used for search queries.

Process 1-2 selects the Web pages which have the maximum p_p values in each of the clusters as the representatives.

3.3 Importance Estimation for Clusters

Finally, Process 1-3 calculates the importance of clusters $c_p = \sum_{k=1}^{n_c} p_p(k)$, defined as a sum of importance of Web pages. Here, n_c denotes the number of Web pages in a particular cluster, and $p_p(k)$ denotes the importance of the k -th Web page in the cluster calculated by equation (2). The value c_p is used to calculate the areas of Web pages in magazine-style layout.

4 Magazine-Style Layout

This section describes our algorithm to place the selected representative Web pages mimicking magazine-style layout.

4.1 Definition of Magazine-Style Web Sites

Magazine-style design is a recent popular style of Web sites, which divides the windows of Web browsers into rectangular subregions, and frames the text and images into the subregions. This design is reasonable when we would like to show various contents in a single page like magazines or newspapers, and therefore many Web sites currently apply this design.

Magazine-style design is a kind of informal design name. There is no standard theoretical definitions of magazine-style at least in our survey. This section describes our own definition of magazine-style design based on our observation of Web sites as the Web design which satisfies the following conditions:

Condition 1: Divide the window spaces into 2, 3, or 4 of vertically-long subspaces, and then frame the Web pages into the subregions.

Condition 2: Assign independent areas to each of Web pages based on their importance, and place the important Web pages in the upper-left part of the window.

Condition 3: Unify the aspect ratios of the rectangular subregions.

Condition 4: Arrange the equally sized and shaped rectangular subregions adjacently.

We measured sizes and aspect ratios of rectangular subregions in the magazine-style news Web site ² to verify [Condition 3]. Based on this experiment, we defined the ideal aspect ratios of the rectangular subregions as follows:

$$\begin{cases} 0.9(\text{if an image is located upper-side}) \\ 1.6, 3.0, \text{ or } 3.8(\text{if an image is located left-side}) \\ 1.0, 3.8, \text{ or } 5.0(\text{if no image is located}) \end{cases}$$

Here, we suppose one or no image is displayed in each of subregions. We also suppose that the image is displayed in the upper or left side of the subregions.

We also observed the same Web sites to verify [Condition 4]. As a result, we found that approximately 95% of Web pages are arranged adjacently with equally sized and shaped Web pages.

4.2 Preliminary Experiment

We tested Squarified Treemap [2] for the layout of Web pages before the development of the magazine-style layout algorithm. We applied

²The New York Times (<http://www.nytimes.com/>), The New Yorker (<http://www.newyorker.com/>), Japan Today (<http://www.japantoday.com/>), sky NEWS (<http://news.sky.com/>), BBC NEWS (<http://www.bbc.com/news/>)

Squarified Treemap provided by d3.js, and found the following problems on the rectangle layout by Squarified Treemap after testing with several sets of Web pages.

- Squarified Treemap generates too small sub-regions. In other words, it is not always visually preferable to assign the exact importance value c_p as the areas of the subregions. Rather, we would like to flexibly adjust their areas to realize visually preferable layout.
- Squarified Treemap generates small number of very thin rectangles while others are well-shaped.

4.3 Data Structure

We define the variables related to clusters of Web pages as follows.

- R : The set of all clusters.
- R_i : The i -th cluster.
- G : The set of all cluster groups.
- G_i : The i -th cluster group.
- C : A set of arbitrary cluster groups G_i, G_j, \dots
- S : Set of Web pages which can be described as either R , G_i or C .

Also, we define the following variables.

- $|S|$: The number of Web pages in S .
- $area(S)$: Total area in the display space occupied by Web pages in S .
- $priority(S)$: Sum of the values c_p of the Web pages in S .
- $ratio(S)$: Ideal aspect ratio of the rectangle occupied by S . See equation 3.
- $type(R)$: Boolean value which denotes the existence of an image in R .
- W : Width of the browser window space.
- H : Height of the browser window space.
- W_{min} : Minimum value of width assigned to Web pages.
- H_{min} : Minimum value of height assigned to Web pages.

We suppose the browser window is not scaled or scrolled unless the layout is recalculated.

The layout technique presented in this section consists of the following processes:

Process 2-1 generates groups of Web page clusters based on their importance values.

Process 2-2 calculates areas of rectangles to be assigned to groups of Web page clusters.

Process 2-3 generate the layout of rectangles.

Process 2-4 displays the representative Web pages based on the layout result.

4.4 Cluster Grouping

Process 2-1 generates groups of Web page clusters based on their importance values c_p , so that it can adjacently display the clusters which have similar c_p values as similarly-sized rectangles. Two clusters R_i and R_j would belong to the same group if they satisfy the following equation:

$$c_{pj} \leq \lceil (c_{pi} \times 1.3) \rceil. \quad (3)$$

Here, c_{pi} and c_{pj} are the importance value of clusters R_i and R_j respectively.

Process 2-1 repeats to group the clusters in the ascending order of their c_p values, so that we can aggressively form groups of smaller clusters. We do not apply the above equation, if R_i has already belonged to another group, or boolean values of the two clusters are different.

Figure 1(Left) shows an illustration of Process 2-1. Figure 1(Left)(1) depicts the list of original importance values, and Figure 1(Left)(2) depicts the grouping result. This section describes the generated groups $G_1 \dots G_{n_G}$, where n_G denotes the number of cluster groups. Process 2-1 contributes to satisfy [Condition 4].

4.5 Area Assignment for Clusters

Process 2-2 calculates the areas of rectangular regions which are to be assigned to Web pages from their importance values c_p . Here, we do not directly apply c_p as relative values of areas. Instead,

(1) Before grouping		(2) After grouping				
c_p	$type_R$	c_p	$type_R$	c_p	$area_{R_i}$	$type_R$
45	true	G_1	45 true	G_1	45	27 true
32	true	G_2	32&30 true	G_2	32&30	20&18 true
30	true	G_3	20 false	G_3	20	12 false
20	false	G_4	11&10 false	G_4	11&10	7&6 false
11	false	G_5	5 false	G_5	5	3 false
10	false	G_6	2&1 false	G_6	2&1	2&1 false
5	false					
2	false					
1	false					

Figure 1: (Left) Grouping of clusters. (Right) Calculation of areas for representative Web pages.

we defined the areas of the rectangular subregions as the following equation.

$$area(R_i) = \left(\frac{priority(R_i)}{W_{min}H_{min}} \frac{WH}{priority(R)} \right)^{\frac{1}{\alpha}} \quad (4)$$

Here, α is a constant real value (1.3 in our implementation) to control the flexibleness of the area assignment. This equation avoids to generate too small subregions, and therefore solves the first problem of Squarified Treemap discussed in Section 4.2.

Figure 1(Right) illustrates an example of the set of areas of cluster groups illustrated in Figure 1(Left)(2).

4.6 Web Page Layout

Process 2-3 divides the Web browser window into multiple rectangular subregions according to the areas calculated by equation (4). Following is the processing flow of Process 2-3, while Figure 2 also illustrates this process.

1. Let $S = G$, $rect_W = W$, and $rect_H = H$.
2. Specify the cluster group S_{top} which has the largest area in S . Calculate the areas of clusters in S_{top} , and tentatively place the clusters at the upper-left end of the empty region of the display space taking into account their aspect ratios $ratio(S_{top})$.

- Figure 2 (I(a) and I(b)) shows illustrations of the tentative layout of G_1 (corresponding to S_{top}) with two choices of the aspect ratio.

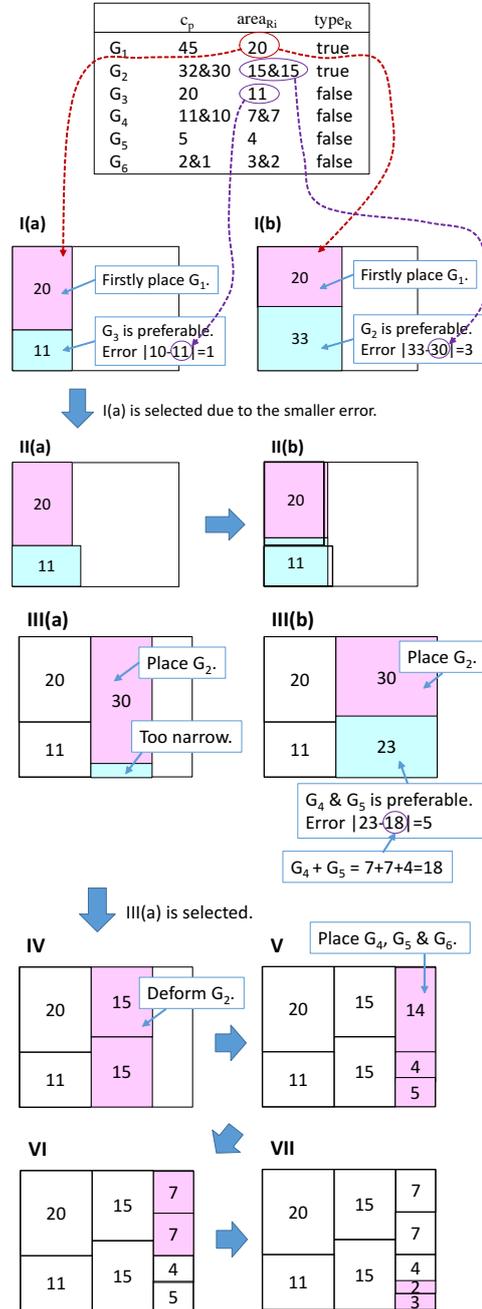


Figure 2: Illustration of our magazine-style layout algorithm.

- If the width of the blank region at the right end is smaller than W_{min} , the algorithm extends the width of S_{top} to fill the right end of the blank. Similarly, the algorithm extends the height of S_{top} to fill the bottom of the blank, if the height of the blank region at the bottom is smaller than H_{min} . Figure 2(III(a)) depicts an example of narrow blank at the bottom, and Figure 2(IV) depicts after the expansion of the rectangles.
3. Calculate the area of the blank region below the tentatively placed rectangles. Rectangular regions painted in sky blue in Figure 2(I(a)(b)) and Figure 2(III(a)(b)) are such blank regions. Repeat this process for each of the tentative layout results, and finally select the layout result which errors of areas between the selected cluster group and the blank region is the smallest. At this moment, our algorithm extracts the candidates of cluster groups to be placed at the sky blue regions. This section describes the set of candidate cluster groups as S_{opt} .
 - In the example illustrated in Figure 2(I), G_3 is selected for the layout in Figure 2 (I(a)), and G_2 is selected for the layout in Figure 2 (I(b)). In this case G_2 or G_3 corresponds to S_{opt} . Layout in Figure 2 (I(a)) is then selected, because the error of areas is smaller.

This process is also applied to the empty regions after the left end of the display space has been filled. Figure 2(III) depicts an example of this situation after the left end of the display space has been filled by G_1 and G_3 . G_2 is selected from S_{opt} as S_{top} , and placed at the upper-left end of the empty space with two choices of aspect ratios, as shown in Figure 2(III(a)) and Figure 2(III(b)).

4. Deform the placed rectangles to gather their right end edges. Figure 2(II(a)) illustrates the rectangles before the deformation, and Figure 2(II(b)) illustrates after the deformation. The technique gathers the edges

so that the rectangles satisfy the equation $asum_{before} = asum_{after}$, where $asum_{before}$ is sum of the areas of rectangles before the deformation, and $asum_{after}$ is sum of the areas of rectangles after the deformation.

5. Apply the following process if S_{opt} is not cleared.
 - a) Let $rect_W = W_{S_{top}}$, and $rect_H = rect_H - H_{S_{top}}$.
 - b) If $|S_{opt}| \geq 3$:
Apply 2. and 3., replacing $|S_{opt}|$ as S . This process applies the nested magazine-style layout to the rectangular subregions.
 - c) If $|S_{opt}| < 3$:
Apply 7., replacing $|S_{opt}|$ as S . As a result, G_4 in Figure 2(V) is divided into two rectangles, as depicted in Figure 2(VI).
 - d) Clear S_{opt} when its all cluster groups are placed in the display space.
6. Apply the following process to S if S_{opt} is cleared.
 - a) Let $rect_W = rect_W - W_{S_{top}}$.
 - b) If $|S| \geq 3$:
Apply 2. and 3. As a result, the process generates a series of rectangles illustrated in Figure 2(III(a)(b)).
 - c) If $|S| < 3$:
Apply 7. As a result, the process fills the right end regions as illustrated in Figure 2(V).
7. a) If S corresponds to G_i :
Horizontally or vertically subdivide into $|S|$ rectangles, and select one of the results which brings the better aspect ratios.
 - b) If S corresponds to C :
Horizontally or vertically subdivide, and apply the following process replacing each of C_i as S .
 - Apply 2. and 3. if $|S| \geq 3$.
 - Apply 7. if $|S| < 3$.

Here, we limit the maximum number of combination of rectangles as 3 in our implementation, because this layout problem is NP-hard. We estimate the complexity as $O(|R|^{l_{max}})$ where l_{max} is the maximum number of set of rectangles, while the complexity would be $O(|R|^{|R|})$ if we do not limit the maximum number of candidate layouts.

Process 2-3 roughly divides the display space into vertically-long subregions to satisfy [Condition 1], and generates rectangular subregions assigning importance-based areas to them to satisfy [Condition 2]. Process 2-1 contributes to satisfy [Condition 3], and flexible aspect ratio selection satisfies [Condition 4].

In the layout results, many of important Web pages are placed in the upper-left side of the display space. It is generally a good property because many people firstly look at the upper-left part of the windows [8] [13].

4.7 Representative Web Page Display

Process 2-4 finally maps the representative Web pages into the rectangular subregions. HistoryPaper displays the title of Web pages, URL, and the summarized text in each of the rectangular subregions. We apply the lead technique [7] which extracts important sentences in the beginning of the documents to generate the summarized text.

5 Results

This section presents results and evaluations of HistoryPaper. We implemented the Web page selection module with Python 2.7 and several libraries including numpy, scipy, and scikit-learn. We also implemented two versions of the Web page layout module with JavaScript¹ and Python 2.7. JavaScript version is developed as a part of Google Chrome Extension, while Python version is just developed for the evaluation of the layout algorithm as shown in this section.

We implemented and executed the client software including Web page layout module as an extension of Google Chrome. The extension sends the browsing history and page contents to

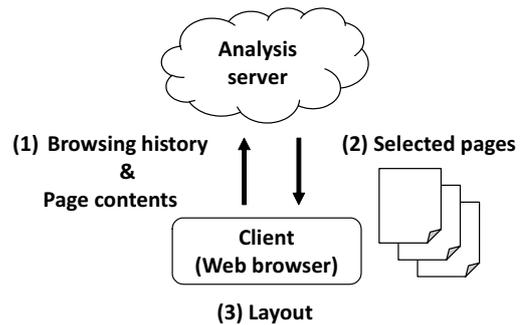


Figure 3: Architecture of our implementation of HistoryPaper.

the analysis server software. We implemented and deployed the server software containing the Web page selection module running on a localhost server. The server then sends the set of selected Web pages to the client software. Finally, the client software calculates the layout and displays the Web pages in the Web browser. Figure 3 shows the system architecture.

5.1 Examples of Layout

Figure 4 shows an example of Web page layout result by HistoryPaper. We can observe that HistoryPaper roughly divides the window into vertically-long subregions, and then divides into well-shaped rectangles which Web pages are mapped. We can also observe that similarly-sized sets of rectangles are adjacently placed. This example demonstrates HistoryPaper realizes the layout of Web pages mimicking magazine-style.

Figure 5 shows other examples of layouts generated from different browsing histories. These examples demonstrate that HistoryPaper realizes various layouts from variety of browsing histories. Meanwhile, Figure 7 shows examples of layouts from the same browsing history while adjusting the number of representative Web pages and ranges of importance values. Table 1 shows the computation time for layout calculation for each of the examples shown in Figure 7. We measured the computation time three times for each of the layouts and showed these averages in this table.

¹<https://github.com/cocodrrips/articlemap-js>



Figure 4: Example of the layout result by HistoryPaper. (Left) Layout result. (Right) Web pages mapped to the rectangular subregions of the layout result.

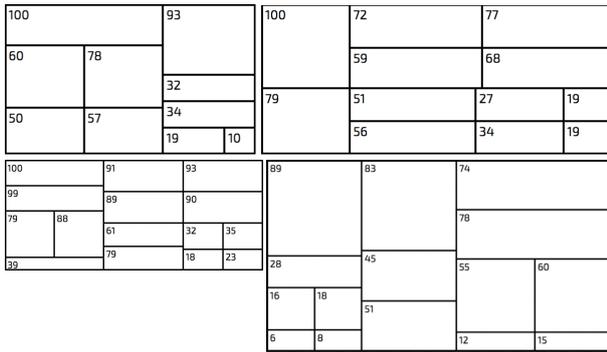


Figure 5: Other examples from different browsing histories.

5.2 Numeric Comparison of Layout Results

We numerically compared the layout results of HistoryPaper and Squarified Treemap. We defined evaluation functions as follows, where r_i denotes the aspect ratio of the i -th rectangle.

1. Calculate D_{ratio} , the difference between the ideal aspect and the actual aspect ratio, by applying equation (5). Here, x is either 0.9, 1.6, 3.0, or 3.8, if the

Table 1: Computation times spent for the layout. (msec.)

rectangles	[0, 10]	[0, 50]	[0, 100]
10	84	77	66
20	150	74	106
30	426	167	236
40	3675	3881	7428

Web page contains images. Otherwise, x is either 1.0, 3.8, or 5.0.

$$D_{ratio} = \sum_{i=0}^n 1 - (\max(x/r_i, r_i/x))^2. \quad (5)$$

2. Calculate W_{ratio} , the worst value of the difference between ideal and actual aspect ratios, by applying equation (6).

$$W_{ratio} = \max\{D_{ratio}(i), i = 1, \dots, n\}. \quad (6)$$

We determine that the layout is better if D_{ratio} and W_{ratio} are smaller.

This section shows the numeric comparison of layout results between HistoryPaper and Squarified Treemap. We compared the following three implementations in this experiment:

- L1** by HistoryPaper.
- L2** by “Deformed Squarified Treemap” featuring the area adjustment by equation (4).
- L3** by “Squarified Treemap” without applying the area adjustment.

Figure 6 shows examples of the layout results applying two datasets. Table 2 also shows D_{ratio} and W_{ratio} of the layout results.

Table 2: D_{ratio} and W_{ratio} of the layout results.

		L1	L2	L3
Dataset 1	D_{ratio}	5.3	23.9	11.6
Dataset 1	W_{ratio}	1.3	10.2	3.4
Dataset 2	D_{ratio}	8.5	13.6	9.5
Dataset 2	W_{ratio}	2.0	3.0	2.2

The result suggests Squarified Treemap occasionally generates thin rectangles in the lower-right ends of the display spaces. Therefore, W_{ratio} sometimes gets extremely bad while using Squarified Treemap. Though this problem can be improved by the area adjustment applying the equation (4), this result demonstrates HistoryPaper obtained better D_{ratio} and W_{ratio} values comparing with Squarified Treemap.

5.3 Subjective Evaluation of Layout Results

We showed the layout results shown in Figure 6 to 21 student participants majoring computer science. We asked to evaluate the layouts in the five-point Likert scale from the following four aspects:

- Q1:** Evaluate total aesthetics and balance of the layout results.
- Q2:** Imagine and evaluate the comfort of reading sentences embedded in the rectangles.
- Q3:** Evaluate the visibility of “every” rectangles.
- Q4:** Evaluate the effect to emphasize rectangle which have larger values.

Table 3 shows the average evaluation values. It demonstrates that HistoryPaper received better evaluations in Q1 and Q2. These results denote that HistoryPaper archived more aesthetic and comprehensive layouts. Also, this table demonstrates that L2 (Deformed Squarified Treemap) received poor evaluations in Q4, while L3 (Squarified Treemap) received poor evaluations in Q3. These results denote that existing techniques have problems on particular rectangles comparing with HistoryPaper.

Table 3: Subjective evaluations of the layout results.

		L1	L2	L3
Dataset 1	Q1	3.59	2.95	2.14
Dataset 2	Q1	4.18	3.90	3.32
Dataset 1	Q2	3.19	2.68	1.91
Dataset 2	Q2	3.91	3.86	3.05
Dataset 1	Q3	3.95	3.95	1.62
Dataset 2	Q3	4.09	4.45	3.27
Dataset 1	Q4	4.18	2.31	2.95
Dataset 2	Q4	3.32	2.77	3.05

5.4 Subjective Evaluation of Web Page Selection Results

Finally, we introduce our user evaluation on Web page selection with 10 student participants ma-

joring computer science. We asked them to use HistoryPaper for one hour while observing the visualization results with their own browsing histories. We prepared the following three implementations of the representative Web page selection process:

- (a):** Selection as presented in this paper
- (b):** Random selection from browsing histories
- (c):** Random selection from clusters

We then asked them the following two questions:

- Q1:** Did you have anything to remember your past actions or previously discovered knowledge while using HistoryPaper?
- Q2:** Which result was the best result for you?

Table 4: Subjective evaluations of the Web page selection results.

Q1	Yes:9, No:1
Q2	(a):8, (b):2, (c):0

Table 4 shows the evaluation results. This table shows that 9 of the participants answered that they remembered something while using HistoryPaper for the first question. This result suggests that the concept of HistoryPaper is effective for the reflection of their past Web browsing. For the second question, 8 of the participants selected (a), while other 2 participants selected (b). The 2 participants mentioned that it was enjoyable if HistoryPaper displayed several unexpected Web pages, and therefore random selection seemed better. We would like to reflect this comments to make HistoryPaper more enjoyable and satisfactory.

6 Conclusion and Future Issues

We presented HistoryPaper aiming to the visualization of summarized daily life information of users’ activity and acquired knowledge.

Following are our future issues to extend HistoryPaper.

Display of large number of Web pages. Our current implementation defines the number of representative Web pages based on sizes of Web browser windows, usually between 8 and 16 in our setting. In our observation there are two types of magazine-style Web sites: the former type does not require any scroll operations by placing all representative Web pages in a limited window space, while the latter type apply vertically-long design which require scroll operations. Our current implementation is based on the former design. On the other hand, it is not necessary to limit the number of Web pages if we apply the vertically-long layout. We would like to extend the algorithm to support such layout.

More factors for representative Web page selection.

In this study we supposed that important Web pages can be specified based on users' aggressive actions estimated from number of accesses and query keywords. That is the reason we applied just two variables m and q to equation (2). We can apply other factors obtained from Web access analysis services such as Google Analytics to select proper representative Web pages.

Analysis of long-term browsing histories. Our current implementation supposes to select representative Web pages from browsing history of a particular day. On the other hand, it is useful to represent long-term (e.g. one year) behavior of a user from his/her browsing histories. It may be difficult to appropriately select representative Web pages from huge number of browsed Web pages using our current algorithm. We would like to extend the algorithm so that we can select reasonable number of representative Web pages from long-term browsing histories.

Aesthetic evaluation of layout results. We would like to discuss how to evaluate aesthetics of the layout results, which is difficult to essentially evaluate by our current criteria D_{ratio} and W_{ratio} . Also, we would like to observe and measure the actual magazine-style Web sites more carefully. Our current implementation on magazine-style layout contain constant values based on our simple observation, such as aspect ratios of rectangles described in Section 4.1, the constant value "1.3" in equation (3), and several other constant values defined in the processing flow described in

Section 4.6. We would like to carefully adjust these values and observe how the layout results are customized, and parameterize such constant values if they are effective to flexibly adjust the layout results. Then, we would like to evaluate the layout results of HistoryPaper by comparing with the actual Web sites.

References

- [1] R. Baeza-Yates, E. Davis, Web Page Ranking Using Link Attributes, International World Wide Web Conference, 328-329, 2004.
- [2] M. Bruls, K. Huizing, J. J. Van Wijk, Squarified Treemaps, Data Visualization 2000, 33-42, 2000.
- [3] J. Gonzalez, I. Rojas, H. Pomares, M. Salmeran, J. J. Merelo, Web Newspaper Layout Optimization using Simulated Annealing, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 32(5), 686-691, 2002.
- [4] M. Kuhna, I.-M. Kivela, P. Oittinen, Semi-automated Magazine Layout Using Content-based Image Features, ACM International Conference on Multimedia, 379-388, 2012.
- [5] O.-W. Kwon, J.-H. Lee, Text Categorization based on k-nearest Neighbor Approach for Web Site Classification, Information Processing and Management, 39(1), 25-44, 2003.
- [6] C. Y. Lin, E. Hovy, Identifying Topics by Position, Fifth Conference on Applied Natural Language Processing: Association for Computational Linguistics, 283-290, 1997.
- [7] M. Litvak, M. Last, M. Friedman, A New Approach to Improving Multilingual Summarization using a Genetic Algorithm, 48th Annual Meeting of the Association for Computational Linguistics, 927-936, 2010.
- [8] M. Nebeling, F. Matulic, M. C. Norrie, Metrics for the Evaluation of News Site Content Layout in Large-screen Contexts, ACM SIGCHI Conference on Human Factors in Computing Systems, 1511-1520, 2011.
- [9] A. Oliver, N. Monmarch, G. Venturini, Interactive Design on Web Sites with a Genetic Algorithm, IADIS International Conference on WWW/Internet, 355-362, 2002.
- [10] T.-H. Ong, H.-C. Chen, W.-K. Sung, B. Zhu, Newsmap: a Knowledge Map for Online News, Decision Support Systems, 39(4), 583-597, 2005.

- [11] T. Strecker, L. Hennig, Automatic Layouting of Personalized Newspaper Pages, Operation Research, 469-474, 2008.
- [12] C. Wartenberg, K. Holmqvist, Daily Newspaper Layout - Designers Predictions of Readers Visual Behaviour - A Case Study, Lund University Cognitive Studies, 126, 1101-8453, 2005.
- [13] H. Weinreich, H. Obendorf, E. Herder, M. Meyer, Not QUite the Average: An Empirical Study of Web Use, ACM Transactions on the Web, 2(1), 5, 2008.

松枝知香

2013年お茶の水女子大学工学部情報科学科卒業。2015年お茶の水女子大学大学院人間文化創成科学研究科理学専攻博士前期課程修了。

伊藤貴之

1990年早稲田大学工学部電子通信学科卒業。1992年早稲田大学大学院理工学研究科電気工学専攻修士課程修了。同年日本アイ・ビー・エム(株)入社。1997年博士(工学)。2000年米国カーネギーメロン大学客員研究員。2003年から2005年まで京都大学大学院情報学研究科COE研究員(客員助教授担当)。2005年日本アイ・ビー・エム(株)退職, お茶の水女子大学理学部情報科学科助教授。2011年同大学教授, 同大学シミュレーション科学教育研究センター長, ACM, IEEE Computer Society, 芸術科学会, 他会員。

Priority, image	Proposed technique	Deformed Squarified Treemap	Squarified Treemap																																																																
<table border="1"> <tr><td>45</td><td>o</td></tr> <tr><td>32</td><td>o</td></tr> <tr><td>30</td><td>o</td></tr> <tr><td>20</td><td></td></tr> <tr><td>11</td><td></td></tr> <tr><td>10</td><td></td></tr> <tr><td>5</td><td></td></tr> <tr><td>2</td><td></td></tr> <tr><td>1</td><td></td></tr> </table>	45	o	32	o	30	o	20		11		10		5		2		1		<table border="1"> <tr><td>45</td><td>30</td><td>10</td></tr> <tr><td></td><td></td><td>11</td></tr> <tr><td></td><td>32</td><td>5</td></tr> <tr><td>20</td><td></td><td>1</td></tr> <tr><td></td><td></td><td>2</td></tr> </table> <p>Dratio 5.3 Wratio 1.3</p>	45	30	10			11		32	5	20		1			2	<table border="1"> <tr><td>1</td><td>11</td><td>32</td><td>45</td></tr> <tr><td>2</td><td>20</td><td></td><td></td></tr> <tr><td>5</td><td></td><td></td><td></td></tr> <tr><td>10</td><td>30</td><td></td><td></td></tr> </table> <p>Dratio 23.9 Wratio 10.2</p>	1	11	32	45	2	20			5				10	30			<table border="1"> <tr><td>3</td><td>20</td><td>45</td></tr> <tr><td>5</td><td>30</td><td></td></tr> <tr><td>10</td><td></td><td></td></tr> <tr><td>11</td><td>32</td><td></td></tr> </table> <p>Dratio 11.6 Wratio 3.4</p>	3	20	45	5	30		10			11	32				
45	o																																																																		
32	o																																																																		
30	o																																																																		
20																																																																			
11																																																																			
10																																																																			
5																																																																			
2																																																																			
1																																																																			
45	30	10																																																																	
		11																																																																	
	32	5																																																																	
20		1																																																																	
		2																																																																	
1	11	32	45																																																																
2	20																																																																		
5																																																																			
10	30																																																																		
3	20	45																																																																	
5	30																																																																		
10																																																																			
11	32																																																																		
<table border="1"> <tr><td>10</td><td>o</td></tr> <tr><td>9</td><td>o</td></tr> <tr><td>8</td><td>o</td></tr> <tr><td>7</td><td></td></tr> <tr><td>6</td><td>o</td></tr> <tr><td>5</td><td></td></tr> <tr><td>4</td><td>o</td></tr> <tr><td>3</td><td></td></tr> <tr><td>2</td><td></td></tr> <tr><td>1</td><td></td></tr> </table>	10	o	9	o	8	o	7		6	o	5		4	o	3		2		1		<table border="1"> <tr><td>10</td><td>8</td><td>4</td></tr> <tr><td></td><td>5</td><td></td></tr> <tr><td>9</td><td>7</td><td>6</td></tr> <tr><td>1</td><td>2</td><td>3</td></tr> </table> <p>Dratio 8.5 Wratio 2.0</p>	10	8	4		5		9	7	6	1	2	3	<table border="1"> <tr><td>1</td><td>5</td><td>8</td><td>9</td></tr> <tr><td>2</td><td>6</td><td></td><td></td></tr> <tr><td>3</td><td></td><td></td><td></td></tr> <tr><td>4</td><td>7</td><td>10</td><td></td></tr> </table> <p>Dratio 13.6 Wratio 3.0</p>	1	5	8	9	2	6			3				4	7	10		<table border="1"> <tr><td>1</td><td>5</td><td>8</td><td>9</td></tr> <tr><td>2</td><td>6</td><td></td><td></td></tr> <tr><td>3</td><td></td><td></td><td></td></tr> <tr><td>4</td><td>7</td><td>10</td><td></td></tr> </table> <p>Dratio 9.5 Wratio 2.2</p>	1	5	8	9	2	6			3				4	7	10	
10	o																																																																		
9	o																																																																		
8	o																																																																		
7																																																																			
6	o																																																																		
5																																																																			
4	o																																																																		
3																																																																			
2																																																																			
1																																																																			
10	8	4																																																																	
	5																																																																		
9	7	6																																																																	
1	2	3																																																																	
1	5	8	9																																																																
2	6																																																																		
3																																																																			
4	7	10																																																																	
1	5	8	9																																																																
2	6																																																																		
3																																																																			
4	7	10																																																																	

Figure 6: Comparison of layout results. Areas of rectangles are calculated from values indicated in columns of priority. Ideal aspect ratios of the rectangles are specified according to presence or absence of an image indicated in columns of image, as discussed in Section 4.1.

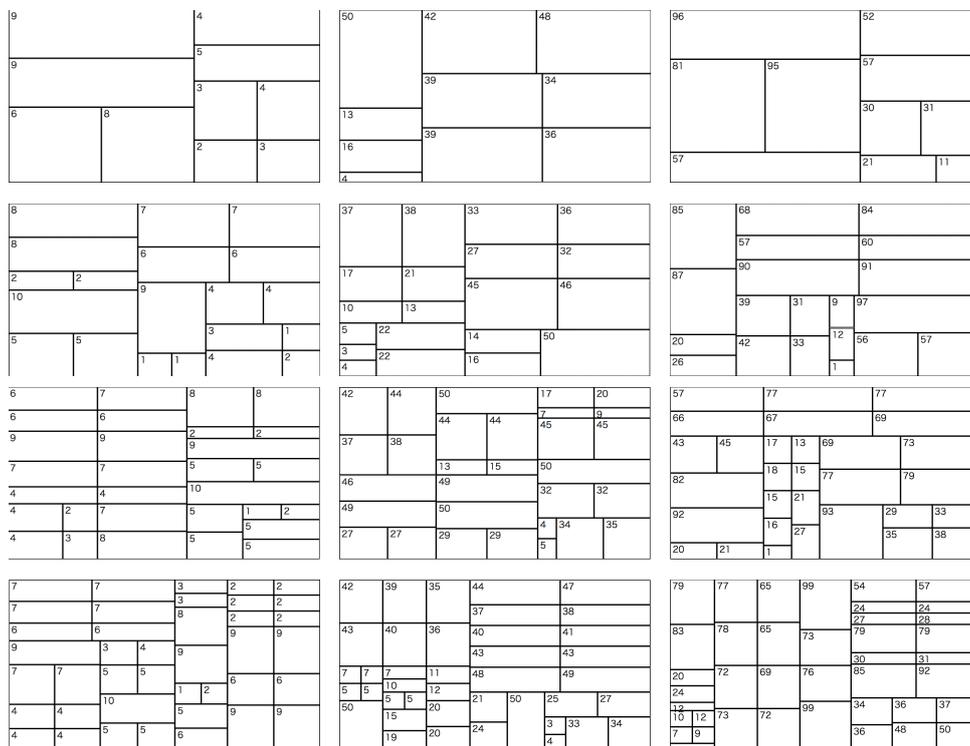


Figure 7: Examples of layout. Number of rectangles are 10, 20, 30, and 40, respectively, from the top to the bottom rows. Left layouts suppose the ranges of importance as $[0, 10]$, while center and right layouts suppose $[0, 50]$ and $[0, 100]$ respectively.