HistoryPaper: A Magazine-Style Layout of Representative Web Pages Extracted From Browsing History

Chica Matsueda
Ochanomizu University
2-1-1 Otsuka, Bunkyo-ku
Tokyo, 112-8610 Japan
coco@itolab.is.ocha.ac.jp

Takayuki Itoh
Ochanomizu University
2-1-1 Otsuka, Bunkyo-ku
Tokyo, 112-8610 Japan
itot@is.ocha.ac.jp

ABSTRACT

Web browsing history of people who use internet everyday represents processes they learned. Visualization of summarized browsing history can contribute to learn what we did. This paper proposes HistoryPaper, a system to extract representative Web pages from browsing history of users, and arranges them like a newspaper. The system helps users to remember what they did in a particular day. HistoryPaper firstly clusters browsed Web pages based on their contents. Then, it calculates the priority of Web pages from search terms and rality of accessed Web pages, and selects the representative page of each cluster. HistoryPaper arranges the representative pages by utilizing a layout algorithm following the magazine style applied in many news sites. This paper presents the numeric evaluation of the layout results to demonstrate the reasonableness of the algorithm.

INTRODUCTION

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 equipments. On the other hand, recording conscious life logs requires continuous labor.

Web browsing history has both advantages of unconsciously and consciously recorded life logs. We do not consciously record the browsing history, while it contains fruitful information representing behavior and knowledge of the users. Also, we expect that reflection of browsing history corresponds to the reflections of actions of particular days or previously studied knowledge of the users.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
VINCI'15, August 24–26, 2015, Tokyo, Japan.
Copyright 2015 ACM 978-1-4503-3482-2...$15.00
DOI:http://dx.doi.org/10.1145/10.1145/2801040.2801044

This paper presents "HistoryPaper", a system which displays sets of Web pages corresponding to the summaries of the browsing history of the particular days. The technique extracts important Web pages in the browsing history of the day, and represents the sets of Web pages like a newspaper on the Web.

There have been several services which display representative Web pages representing 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. Paper.li applies the magazine-style design to arrange the sets of Web pages, to realize the quick overview of user-interested Web pages. 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.

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, as shown in Figure 1. 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

Figure 1. Example of a Web page applying the magazine-style design. (The New York Times, http://www.nytimes.com/)

rectangular subregions adjacently.

We measured the sizes and aspect ratios of rectangular subregions in the magazine-style news Web site [2] to verify [Condition 3]. Figure 2 shows the distribution of the measured aspectratios.

We defined the ideal aspect ratios of the rectangular subregions as shown in Table 1 based on the result. In this definition, we suppose one or no image is displayed in each of the subregions. We also suppose that the image is displayed in the upper or left side of the subregions.

Table 1. Ideal aspect ratios defined in this study.

	width/height (image position)
With image	0.9 (upper) 1.6 (left) 3.0 (left) 3.8 (left)
Without	1.0 3.8 5.0

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

This paper presents an automatic Web page layout algorithm which satisfies the above conditions.

[2]The 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/)

Figure 2. Aspect ratios of rectangular subregions in magazine-style new Web sites.

REPRESENTATIVE WEB PAGE SELECTION

HistoryPaper automatically selects multiple representative Web pages browsed on a particular day, and displays in a Web browser window. This section describes the processing flow to select representative Web pages.

Our technique firstly calculates the 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. [5] 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 presented technique implemented for HistoryPaper firstly applies a clustering algorithm to the Web pages based on the similarity of their contents, and then selects most important Web pages from each of the clusters. We can summarize the activity of the users on a particular day by displaying the selected

Web pages.

Clustering of Browsed Web Pages

HistoryPaper firstly divides all the Web pages browsed on a particular day into n clusters. It specifies n (between 6 and 12 in our implementation) based on the sizes of Web browser windows and the number of browsed pages on the day in advance.

The following is the brief description of the clustering process.

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.

Importance Estimation for Web Pages

The presented technique then calculate the importance of the Web pages p_p by applying the following equation:

$$\mathrm{p}_p = p_a q + p_b m + p_c mq \tag{1}$$

where,

- m is the total number of keywords used for search query in the day and contained in the current Web page.
- $q = m_{wp}/m_{all}$ is the valuableness of the current Web page, where m_{wp} is the number of accesses to the current Web pages in the day, and m_{all} is the total number of accesses to the all Web pages in the day.
- p_a, p_b, and p_c are experimentally defined constant real values.

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.

The technique selects the Web pages which has the maximum p_p value in each of the clusters as the representatives.

Importance Estimation for Clusters

The technique then calculates the importance of clusters c_p by the following equation.

$$\mathrm{c}_p = \sum_{k=1}^{n_c} p_p(k) \tag{2}$$

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 the equation (1). The value c_p is used to calculate the areas of Web pages in magazine-style layout.

MAGAZINE-STYLE LAYOUT

This section describes our algorithm to place the selected representative Web pages mimicking magazine-style layout. There have been several techniques on layout of multiple Web pages applying optimization techniques [4]; however, there have been few studies on magazine-style-specific layout for multiple Web pages.

Preliminary Experiment

We tested Squarified Treemap [1] for the layout of Web pages before the development of the magazine-style layout algorithm. Squarified Treemap divides a rectangular region into a set of nearly-square subregions, while areas of the subregions can be freely specified. Figure 3 shows an example of rectangle layout by Squarified Treemap assigning specific areas to the subregions. We applied the implementation of Squarified Treemap provided by d3.js.

Figure 3. Example applying Squarified Treemap.

As a result, we found the following problems on the rectangle layout by Squarified Treemap after testing with several sets of Web pages.

- Squarified Treemap may generate too small subregions. 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 may generate small number of very thin rectangles while others are well-shaped.

This section describes a new algorithm mimicking magazine-style layout while solving the above mentioned problems.

Data Structure

This section defines 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,...
- S: Set of Web pages which can be described as either R, G, G_i, or C.

Also, this section defines the following variables.

- $|S|$: The number of Web pages contained 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 Table 1.
- $type_R$: 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.

Here, we suppose that the browser window is not scaled or scrolled during the usage of this technique.

Cluster Grouping

The technique groups 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:

$$c_{pj} \leq \lceil (c_{pi} \times 1.2) \rceil \tag{3}$$

where c_{pi} and c_{pj} are the importance value of clusters R_i and R_j respectively. The technique does not apply the above equation, if R_i has already belonged to another group, or boolean values of the two clusters are different.

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

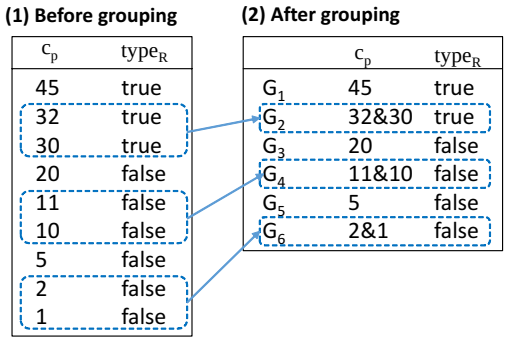

Figure 4. Grouping of clusters.

Area Assignment for Clusters

The technique then 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, we defined the areas of the rectangular subregions as the following equation.

$$area_{R_i} = \left(\frac{priority_{R_i}}{W_{min} H_{min}} \frac{WH}{priority_R} \right)^{\frac{1}{\alpha}} \tag{4}$$

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 our preliminary experiment.

Figure 5 illustrates an example of the set of areas of cluster groups illustrated in Figure 4(2).

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

Figure 5. Calculation of areas for representative Web pages.

Web Page Layout

The technique then divides the Web browser window into multiple rectangular subregions according to the areas calculated by the equation (4), and finally maps the representative Web pages. Figure 6 illustrates the layout algorithm described in this section.

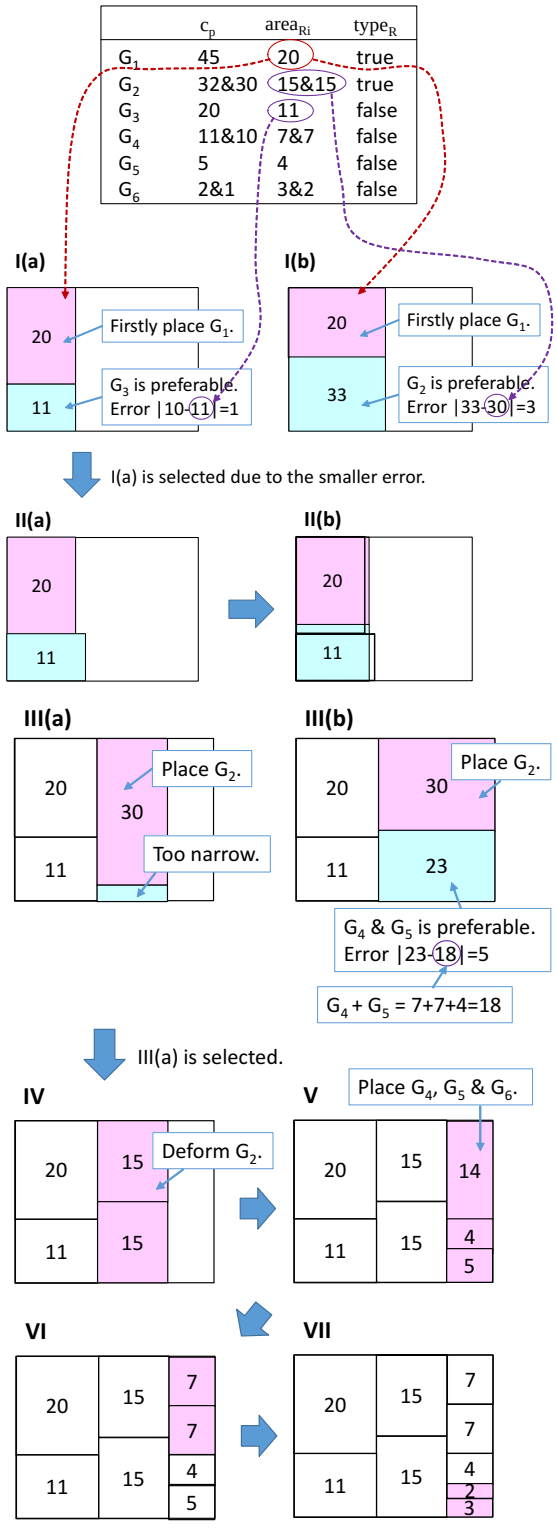

Figure 6. Illustration of our magazine-style layout algorithm.

Following is the processing flow of our layout algorithm.

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 6 (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.
   - 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 6(III(a)) depicts an example of narrow blank at the bottom, and Figure 6(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 6(I(a)(b)) and Figure 6(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 6(I), G_3 is selected for the layout in Figure 6 (I(a)), and G_2 is selected for the layout in Figure 6 (I(b)). In this case G_2 or G_3 corresponds to S_{opt}. Layout in Figure 6 (I(a)) is then selected, because the error of areas is smaller.

   This process is also applied to the empty regions after left end of the display space has been filled. Figure 6(III) depicts an example of this situation after 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 6(III(a)) and Figure 6(III(b)).

4. Deform the placed rectangles to gather their right end edges. Figure 6(II(a)) illustrates the rectangles before the deformation, and Figure 6(II(b)) illustrates after the deformation. The technique gathers the edges so that the rectangles satisfy the equation $area_{before} = area_{after}$, where $area_{before}$ is sum of the areas of rectangles before the deformation, and $area_{after}$ 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 6(V) is divided into two recrangles, as depicted in Figure 6(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 6(III(a)(b)).

(c) If $|S| < 3$:
Apply 7. As a result, the process fills the right end regions as illustrated in Figure 6(V).

7. (a) If S corresponds to G_i:
Horizontally or vertically subdivide into $|S|$ rectangles, and select the 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 candidate layouts (2 in our implementation)in 3., because this layout problem is NP-hard. We estimate the complexity as $O(|R|^{l_{max}})$ where l_{max} is the maximum number of layouts, while the complexity would be $O(|R|^{|R|})$ if we do not limit the maximum number of candidate layouts.

The above algorithm roughly divides the display space into vertically-long subregions to satisfy [Condition 1], and finally generates rectangular subregions assigning importance-based areas to them to satisfy [Condition 2]. The grouping process 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.

Representative Web Page Display

HistoryPaper 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 [2][6] which extracts important sentences in the beginning of the documents to generate the summarized text.

RESULTS

Web Page Selection

This section introduces an example of Web page selection result from the browsing history of the first author. Table 2 shows the list of the selected Web pages, where c_p denotes their importance.

Table 2. Web page selection result.

c_p	Title of Web pages
399	2D bin packing with JavaScript and *canvas*
243	Output of log files in Android
239	Packing algorithms (Kyoto university)
199	Port check
107	Meaning of "heuristic"
90	Packing algorithm and complexity
72	Approximation of rectangle packing
57	Questions on the rental server
16	FINAL FANTASY Official song

Layout

Figure 7 shows an example of Web page layout result by HistoryPaper. The set of Web pages is same as applied in Figures 3 and 4. 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.

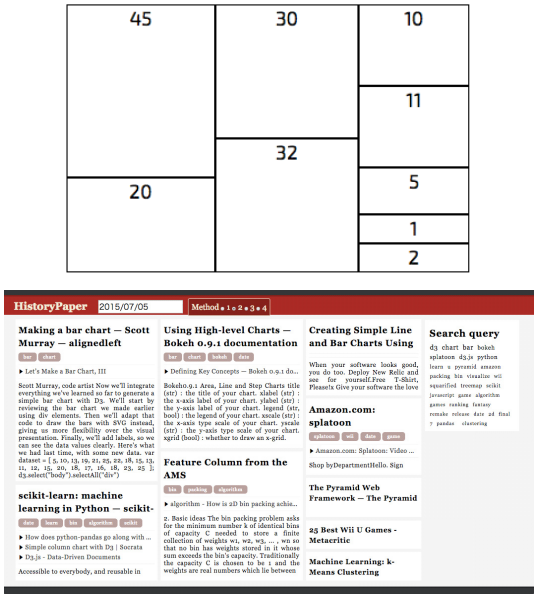

Figure 7. Example of the layout result by HistoryPaper. (Upper)Layout result. (Lower)Web pages mapped to the rectangular subregions of the layout result. Remaark that left three columns are formed from the layout result. The right end column just shows a list of search query terms.

Evaluation of Layout

We numerically compared the layout results of HistoryPaper and Squarified Treemap. We defined the 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 shown in Table 1 and the actual aspect ratio, by applying the equation (5).
   Here, x is either 0.9, 1.6, 3.0, or 3.8, if the Web page contains images. Otherwise, x is either 1.0, 3.8, or 5.0.

$$D_{ratio} = \sum_{i=0}^{n} 1 - (\min\{f(x,i)\})^2 \qquad (5)$$

   where,

$$f(x,i) = \begin{cases} \dfrac{x}{r_i} & (x > r_i) \\ \dfrac{r_i}{x} & (x \leq r_i) \end{cases}$$

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

$$W_{ratio} = \max\{D_{ratio}(i), i = 1, 2, \cdots, n\} \qquad (6)$$

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

Comparison of Layout Results

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

- Squarified Treemap without area adjustment by equation (4)
- Squarified Treemap with area adjustment
- HistoryPaper

Figure 8 shows examples of the layout results applying the same set of Web pages, and values of D_{ratio} and W_{ratio}.

The result suggests Squarified Treemap may occasionally generate 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.

User Evaluation

This section introduces our user evaluation with 10 student participants in master's course. We asked them to use HistoryPaper for one hour while observing the visualization results with their own browsing histories. For this experiment, 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 of Web pages

We then asked them the following two questions:

1. Did you have anything to remember your past actions or previously discovered knowledge while using HistoryPaper?
2. Which result brought from the three implementations was the best result for you?

As a result, 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.

CONCLUSIONS

This paper presented HistoryPaper, a system which selects sets of important Web pages browsed on a particular day, and displays the sets of Web pages by utilizing a new layout algorithm mimicking magazine-style layout. HistoryPaper visualizes the summary of users' daily activity and acquired knowledge. This paper introduced the comparison of layout results by HistoryPaper and Squarified Treemap to demonstrate the effectiveness of the presented layout algorithm.

As a future work, we would like to discuss how to evaluate beauty 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, and evaluate the layout results of HistoryPaper by comparing with the actual Web sites.

REFERENCES

1. Bruls, M., Huizing, K., Van Wijk, J. J., Squarified treemaps, Data Visualization 2000, 33-42 (2000).
2. Edmundson, H. P., New methods in automatic extracting, Journal of the ACM (JACM), 16(2), 264-285 (1969).
3. Genest, P. E., Lapalme, G., Nerima, L., Wehrli, E., du Langage, T., A symbolic summarizer for the update task of tac 2008, The First Text Analysis Conference (2008).

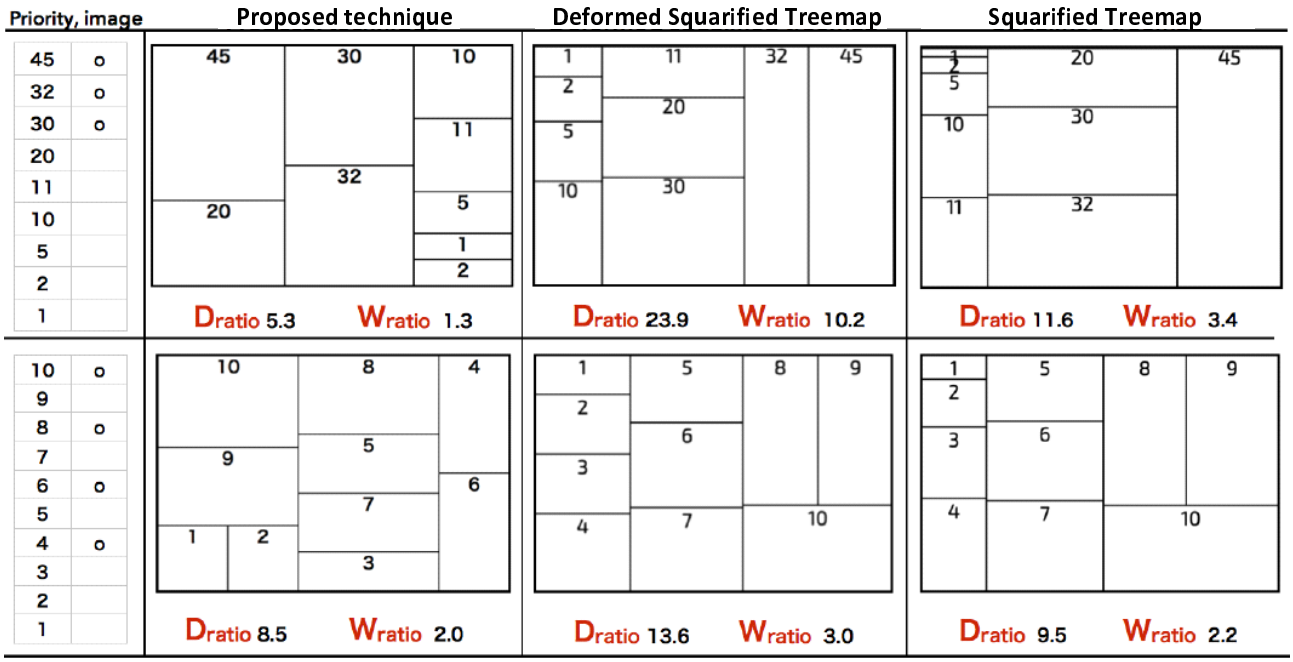

Figure 8. Comparison of layout results.

4. Gonzalez, J., Rojas, I., Pomares, H., Salmeran, M., Merelo, J. J. Web newspaper layout optimization using simulated annealing, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 32(5), 686-691 (2002).

5. Lin, C. Y., Hovy, E., Identifying topics by position, Fifth Conference on Applied Natural Language Processing: Association for Computational Linguistics, 283-290 (1997).

6. Litvak, M., Last, M., Friedman, M., A new approach to improving multilingual summarization using a genetic algorithm. 48th Annual Meeting of the Association for Computational Linguistics: Association for Computational Linguistics, 927-936 (2010).

7. Wartenberg, C., Holmqvist, K., Daily Newspaper Layout - Designers Predictions of Readers Visual Behaviour - A Case Study, Lund University Cognitive Studies, 126, 1101-8453 (2005).