VIEW-BASED WEB PAGE RETRIEVAL USING INTERACTIVE SKETCH QUERY

Yasuyuki Watai†, Toshihiko Yamasaki‡, and Kiyoharu Aizawa‡
†Dept. of Frontier Informatics, ‡Dept. of Information and Communication Engineering,
The University of Tokyo

ABSTRACT

We propose a novel view-based web page retrieval system that enables a user to search web pages using a visual query, namely the user's freehand sketch. We believe the proposed method will suit retrieval from a set of web pages such as a user's local browsing history. The system aims to help the user revisit a particular web page without using query words. Using color signature features and Earth-Mover's Distance, the system evaluates the similarity between web pages and the user's sketch drawn via the GUI of the system. In order to accelerate the interaction, the results of the similarity evaluations are shown immediately after the user draws each stroke of the sketch, with the results being interactively reordered. Experiments using our prototype system showed that users find their target pages after only a few strokes. Experimental results for inexperienced users showed that 71% of search tasks were completed within one minute, using the prototype system. The median time for the tasks was 40 seconds.

Index Terms**—**content-based image retrieval, web page retrieval, query by sketch, user interaction, user interface

1. INTRODUCTION

Web pages are more than text documents. They can be full of carefully designed multimedia content such as images, sounds, and movies. The design, namely the *visual information* or *view* of the web page, is information that is perceptually important for humans. When we browse rendered web pages, we unconsciously remember their visual impression. Therefore, we propose a view-based web page retrieval technique. It is different from multimedia document retrieval systems such as WebMARS [1], which focus on images in the web pages. We focus on the design of the web page, i.e. the visual impression of the page. Then, if we forget (or cannot find) an appropriate keyword but do remember some visual features such as the color of the whole page or the page header, our proposed system finds the target page, by our sketching the target based on our memory and its retrieval of similar pages. The main application we have in mind is revisiting a specific page in the user's local web browsing history. Fig. 1 shows a diagram of the proposed system. The HTTP proxy server is omitted in the prototype system used for evaluation.

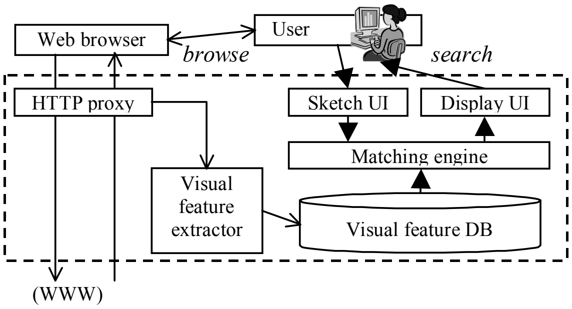


Fig. 1. System diagram of our retrieval system

The most important issue is how to evaluate visual similarity. The main part of a web page is a semistructured HTML document. It seems straightforward to utilize the HTML structure, commonly known as the Document Object Model (DOM), to extract visual features such as the layout. Using this idea, web page search by sketch using DOM analysis has already been proposed [2]. However, it is difficult to extract visual information (design) from the DOM alone because (1) the DOM structure and the layout structure are not always perceived by humans to be the same, and (2) most of the visually rich content of web pages derives from data other than the HTML.

Problem (1) is not only caused by errors in HTML grammar. There may be little relationship between the DOM structure and the rendered result, as is the case for the successors to HTML: XHTML [3] and CSS [4].

An example of Problem (2) is an image. We may know that "an image is there" from the DOM structure, but we do not know "how it seems." It may be an eye-catcher or mere decoration of the page. To get further information, we have to analyze the image. Different types of media need different algorithms for analysis, and new media may appear in the future. Such a DOM-based approach is unrealistic, which is a problem faced by WebMARS.

We propose a fully image-based visual similarity computation for web page retrieval. Our proposed method analyzes a screenshot of the web page and extracts low-level image features. It enables us to handle the whole web page in the uniform manner usual in CBIR systems. Our prototype system calculates the similarity between web pages and the user sketch based on the color signature [5]. Layout is defined as the perceptual information for users. A fully image-based layout analysis and matching technique has been proposed in [6]. However, this is based on a layout model that is so limited that it cannot be applied to our case.

1-4244-1437-7/07/$20.00 ©2007 IEEE

Our system treats layout information by a straightforward template-matching strategy.

In conventional CBIR systems, the expression of user requests has been an issue. The most widely used approach is “query by example” [7]. In such systems, a user selects and/or inputs images similar to the desired one. This is easy for the user, if the user (or the system) has images appropriate to the query. Unfortunately, most CBIR tasks do not meet this condition. This is the case for our system, because the design of web pages has many variations and any subset stored in the system will be limited. Therefore, we introduce a “query by sketch” [7–9] interface to the prototype system. With this interface, users can compose their queries. However, the difficulty and quality of query depends heavily on the skill of each user. We implemented a modified version of the interactive sketch interfaces in [8, 9] to reduce both the difficulty and the total search time. Our interactive interface focuses on visual web page search.

We have evaluated our prototype system for searching a local history of visited web pages. The task was to search for known specified web pages in a set of web pages. Experimental results for first-time users showed that our method is effective for this scenario. Of the search tasks, 71% were completed within one minute.

2. VIEW-BASED WEB PAGE SEARCH

2.1. Visual Feature Extraction

This section describes the visual feature extractor shown in Fig. 1. In the first phase of visual feature extraction, our system renders the given web page and obtains its screenshot. Our prototype uses the relevant component of Internet Explorer 6. The rendering size is fixed at 1024 × 768 pixels and the overflowing part of the page is eliminated.

Second, the screenshot is divided into small blocks (32 × 32 pixels in our prototype) and image features are extracted from each block. We use the color signature proposed by Rubner [5] as the fundamental visual feature. The color signature is a set whose element is a pair comprising a color vector (in CIE L*a*b* color space) and its weight (number of pixels). It suits a sparse color distribution such as a web page because it deals with each color as a set of components. The distance between two signatures is defined as the Earth-Mover’s Distance (EMD) [5].

In practice, color quantization is needed before extracting the color signature to reduce the cost of computing the EMD. We applied a simple algorithm using web-safe colors instead of color quantization. Our prototype system assigns the most similar web-safe color (measured by L_2 in the CIE L*a*b* color space) to each original color. In the Red/Green/Blue (RGB) color space, the web-safe colors are those 216 combinations of R, G, and B for which each component takes one of six values. This is regarded as the standard color palette for web pages.

In the final phase, web pages are registered in index lists of colors. They are used to select matching candidates.

2.2. Page–Query Matching

We designed an algorithm to match web pages with a sketch produced via an interactive sketch interface. Each user input (or “stroke”) to the sketch canvas, such as freehand drawing or flood filling, triggers page–query matching. In one matching cycle, part of the sketch and the corresponding part of the web pages in the database are compared. The total similarity between the user sketch and a web page for one activity is calculated as a weighted sum of result values of all the completed matching cycles.

A query color signature C_Q is extracted from the bounding box of the latest changes produced by user strokes. We call this region the “query region.” Our system treats one query region as one object in the web page. However, we have observed that an object is often input using multiple strokes. Therefore, we implemented this heuristic for updating the query region dynamically: if the query region generated by a new user input intersects with the previous one, and the previous operation is a freehand drawing, then the system merges both query regions, and the previous matching task is replaced by the new one.

The system locates the query region on the screenshot of a web page, and determines the matching window *Pi* around it (our prototype system searches within 128 pixels from the query region). For each search window, the page color signature C_{Pi} is extracted from P_i and the EMD between signatures, EMD($C_{Pi,}C_Q$), is calculated. The minimum sliding step of the window depends on the block size in the feature extraction. As mentioned in Section 2.1, the block size is 32 × 32 pixels in our system, so 9 × 9 = 81 calculation times are required. The similarity *SimPQ* between the page and the query is calculated from the maximum and minimum values of EMD, as follows.

$$SimPQ(P,Q)=\frac{\max_i(EMD(C_{P_i},C_Q))-\min_i(EMD(C_{P_i},C_Q))}{\max_i(EMD(C_{P_i},C_Q))}$$

SimPQ increases when there is an object with similar color in the search area and there is an area with dissimilar color next to it. If the query region is the whole page, the normalized minimum value of EMD is used instead of *SimPQ*.

Note that our algorithm ignores the spatial distance between the query region and the best-matched region on the web page. Some rendered web pages have blank areas on the side of the browser window (e.g. Figs. 3(a) and 3(c)). Those areas sometimes appear in user sketches and sometimes not. When a user ignores the blank area, the user tends to fill the canvas with queries. From our observations, this input pattern was inconsistent both from page to page and from user to user.

2.3. Page–Sketch Matching

The similarity *SimPS* between the sketch S_k after the k-th matching cycle and a web page P is calculated by the following formula:

$$SimPS(P,S_k) = w \times SimPS(P,S_{k-1}) + SimPQ(P,Q_k)$$

where w is a forgetting factor [0, 1]. In the prototype system, we fixed it at 0.9. $SimPS(P,S_k)$ is greater for the pages matching the newest strokes.

3. USER INTERACTION

3.1. User Interface

The goal of the search operation is to find the target(s) as soon as possible. A fast and accurate matching algorithm may help users to achieve this goal, but that is not all. It is important that the user, not the algorithm, can find the intended target in a short time.

The user interface of our prototype is shown in Fig. 2. The left half is an input interface. There is a sketch canvas, with a fixed web-safe color palette and a button to discontinue search task(s). The canvas supports the basic functions for drawing: freehand drawing, flood-fill, and undo.

The search results are sorted in the descending order of similarity and thumbnails are shown on the right half of the interface. Users can scroll the area to see further results. For our system, the user requirement is for only one page exactly matched to the intended one. Our prototype system updates results soon after the similarity calculation between the query and a page in the database is completed. This architecture is not efficient in terms of computational cost because it devotes much time to sorting. However, it returns the target page faster and saves time overall for search operations whenever preliminary searching works well and the matching order is suitable. This architecture also reduces the subjective waiting time for the search because users enjoy the dynamically changing results.

3.2. Interactive Searching by Sketch

With our prototype, the user first starts a search activity by selecting a color from the color palette (Fig. 2(a)). The selected color is an important key, which we use for efficient retrieval. In other words, we find all web pages indexed by colors within a certain distance from the selected color and register them as candidates. This background task is quite fast. At this stage, the candidates and the matching order are determined. The matching order is important for our system because the faster the page is involved in the matching, the faster it appears in the result.

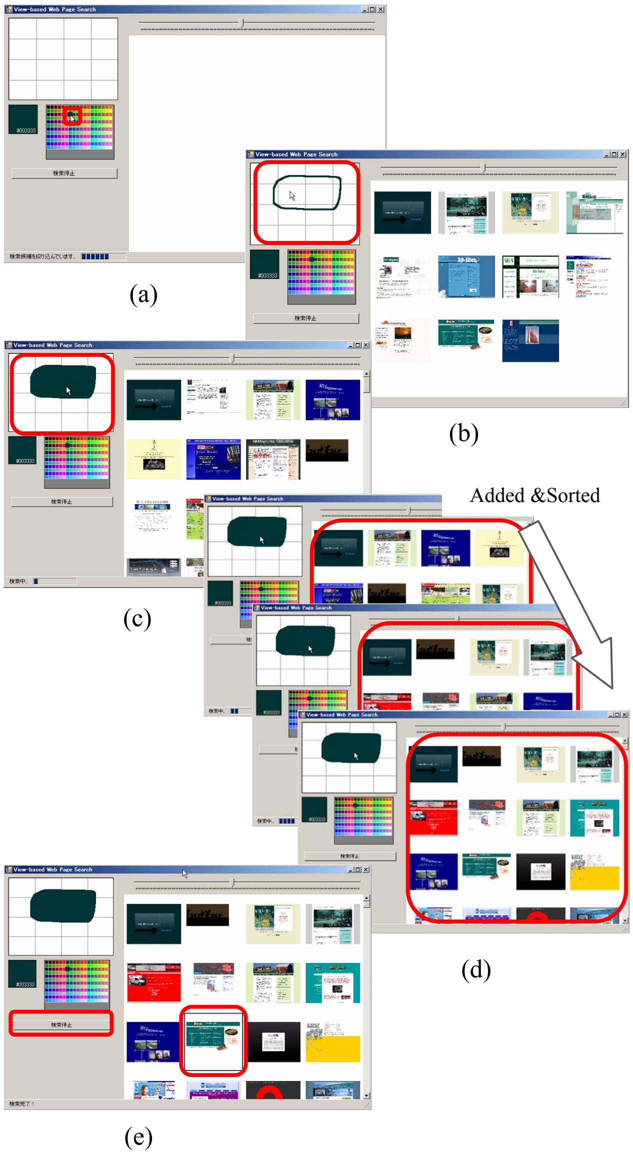
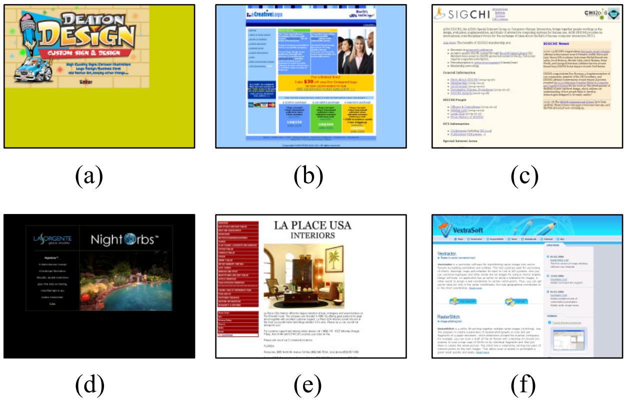


Fig. 1. Example of interactive sketch search: (a) selecting a color, (b) freehand drawing, (c) modifying the sketch by flood-fill, (d) waiting for the target page is shown, (e) the target page is found.

Next, the user starts a freehand drawing (Fig. 2(b)). Soon after one operation is finished (i.e. when the mouse button is released), a query signature is extracted from the query region and matching is started. The results are updated immediately.

In the example in Fig. 2, the user does not find the intended target with the first stroke. The user fills the inside of the contour before the first matching task is finished. The system considers this stroke as a refinement of the previous stroke. The query region and the previous matching task are replaced by those generated from the new operation (Fig. 2(c)). While the order is dynamically changing, the target



(a) (b) (c)

(d) (e) (f)

Fig. 3. Target pages of the experiments

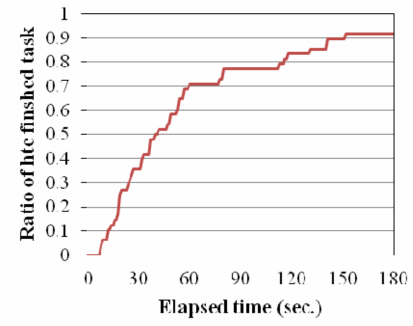


Fig. 4. Success rate of the search activity

appears (Fig. 2(d)). The user cancels all ongoing tasks and clicks on the thumbnail of the target page.

4. EXPERIMENTAL RESULTS

We monitored the user activity for the virtual search task with a fixed dataset. The dataset contained 500 top pages collected randomly from the Arts/Design category of the Open Directory Project (http://www.dmoz.org/Arts/Design/). Eight subjects took part in the experiment. They were all experienced users of PCs and web browsers, but were first-time users of our system.

Before the search task, each subject was requested to browse the web, starting from a given web page. That page was used as the target. One minute later, the user was requested to close the browser window and to start searching for the target page, using our system. The sketch was drawn without seeing the target page. Each user repeated this task for the six target pages shown in Fig. 3.

The success ratio of the finished search task for all (6 (pages) × 8 (subjects) = 48) tasks is shown in Fig. 4. The x-axis of Fig. 4 indicates the time elapsed since the user's first operation. In our experiments, the search time was limited to a maximum of three minutes. The median time for all activity was 40 seconds. Thirty-four (71%) search tasks were finished within a minute, while four (8%) cases were unfinished after three minutes. Three of the failed tasks were for target page (c), and one was for target page (d). The latter failure was because of a memory lapse: the user searched for another page linked to the target. Target page (c) was mainly composed of text, which was difficult to sketch, and our matching algorithm did not process it well. We presume that a conventional text-based search would be suitable for such pages. More semantic analysis, such as the HTML-based analysis proposed in the information retrieval field [10], might be helpful when we have to search for such pages. Note that the performance worsened significantly after the first 60 seconds. This was because of unexpected actions by users when refining their sketches. Most users tended to redraw almost the same sketch when the primary results were not satisfactory. However, this similar sketch returned similar results. These efforts, after the second iteration, were very time consuming.

5. CONCLUSION

In this paper, we have proposed a novel view-based search for retrieving web pages. Our EMD-based matching algorithm and interactive interface are effective. Web pages' visual information, especially their colors and layout, are shown to be a key to searching for previously visited web pages. In our local history search scenario, personalization may lead to enhanced performance. For example, some parameters in the matching algorithm could be tuned for specific users. Relevance feedback techniques will help such approaches.

References

[1] M. O. Binderberger et al., "WebMARS: A Multimedia Search Engine for Full Document Retrieval and Cross Media Browsing," The 6th Workshop on Multimedia Information Systems (MIS2000), 2000.

[2] Y. Hashimoto et al., "Web Page Search by Sketching Layout," Interaction 2004, No. 5, pp. 113–120, 2004 (in Japanese).

[3] W3C, "XHTML™ 1.0 The Extensible HyperText Markup Language (2nd Edition)," W3C Recommendation, http://www.w3c.org/TR/xhtml1, Aug. 2002.

[4] W3C, "Cascading Style Sheets, Level 1," W3C Recommendation, http://www.w3c.org/TR/REC-CSS1, Jan. 1999.

[5] Y. Rubner et al., "The Earth Mover's Distance as a Metric for Image Retrieval," *IJCV*, Vol. 40, No. 2, pp. 99–121, Nov. 2000.

[6] Y. Takama et al., "Visual Similarity Comparison for Web Page Retrieval," The 2005 IEEE/WIC/ACM Int'l Conf. on Web Intelligence (WI'05), pp. 301–304, 2005.

[7] C. E. Jacobs et al., "Fast Multiresolution Image Querying," SIGGRAPH'95, pp. 277–286, 1995.

[8] T. Kato et al., "Query by Visual Example – Content Based Image Retrieval," The 3rd Int'l Conf. on Extending Database Technology, pp. 56–71, 1992.

[9] J. Lee et al., "Image Navigation: A Massively Interactive Model for Similarity Retrieval of Images," *IJCV*, Vol. 56, No. 1, pp. 131–145, Jan. 2004.

[10] D. Cai et al., "VIPS: A Vision-based Page Segmentation Algorithm," Microsoft Technical Report, MSR-TR-2003-79, 2003.