Interactive Poster: Web Site Visualization Using a Hierarchical Rectangle Packing Technique

Yumi Yamaguchi Takayuki Itoh Yuko Ikehata Yasumasa Kajinaga
IBM Research, Tokyo Research Laboratory
1623-14 Shimotsuruma, Yamato-shi, Kanagawa 242-8502 JAPAN
{yyumi, itot, ikehata, kajinaga}@jp.ibm.com

1. Introduction

Visualization of Web sites is interesting not only for the navigation for end-users, but also for their monitoring and analysis. Many of existing Web site analysis software provides simple representations, such as bar charts, pie charts, and ranking tables, to provide access statistics of the Web sites. It is often difficult to quickly find subconscious but interesting access trends by only watching such simple representations.

This poster introduces our hierarchical data visualization algorithm [1,2] and presents the application of the algorithm for the Web site visualization. Our visualization algorithm represents hierarchical data as a group of nested rectangles. The algorithm places thousands of rectangles in nearly minimized display spaces in several seconds. Our application imports access log files of Web servers, and represents Web sitemaps by constructing hierarchical data according to the directory structures of Web pages, and access statistics by mapping them onto the sitemaps. This poster also shows an example of interesting access trend discovered by using our application.

2. Proposed Hierarchical Data Visualization

This section briefly describes our hierarchical data visualization algorithm. Figure 2 shows an illustration of the order of data layout in our technique. Our algorithm first packs icons (painted square dots in Figure 2) that denote leaf nodes, and then generates rectangles that enclose the icons to denote non-leaf nodes. Similarly, it packs a set of rectangles that belong to higher levels, and generates the larger rectangles that enclose them. Repeating the process from the lowest level toward the highest level, the algorithm places all of the data onto the layout area.

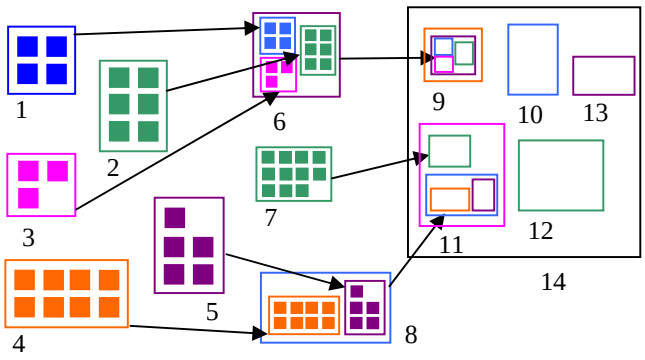

Figure 2. Algorithmic overview of the layout process for hierarchical data. Numbers in this figure denote the order of the process.

Our algorithm places a set of rectangle one-by-one, while it satisfies the following two conditions:

[Condition 1] Rectangles must be placed without overlapping each other.

[Condition 2] Rectangles should be placed where the layout area occupied by the rectangles is minimized.

To promptly pack rectangles, the algorithm sorts the rectangles according to their areas, and places them on a display space in the sorted order, as shown in Figure 3. Here, our algorithm favors accelerating the rectangle packing process rather than entirely minimizing the layout space. We therefore did not apply optimization schemes to find the configuration of rectangles, but used a heuristic to quickly find gaps and place the remaining rectangles in the gaps. The heuristic uses a triangular mesh connecting centers of previously placed rectangles. Our algorithm refers triangular elements in the order of their sizes to quickly find gaps to promptly place rectangles. References [1,2] describe the detailed algorithm.

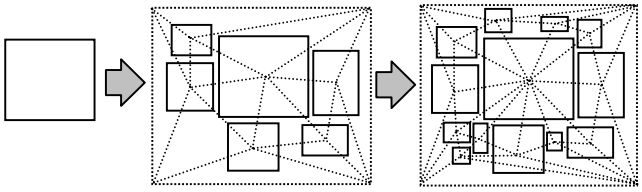

Figure 3. Illustration of rectangles packing algorithm. The algorithm first places the largest rectangle, and then places the others in the order of their areas. A triangular mesh that connects centers of rectangles are referred to quickly find gaps to promptly place the rectangles.

3. Experiments with Web Site Visualization

This section describes our Web site visualization tool and an example of interesting access trend discovered by using the tool. Our Web site visualization tool imports access log files of Web servers. Our tool first extracts lines whose resources denote pages (HTML, CGI, etc.) and eliminates the others from the input access log file. It then collects the URLs of the resources from these lines, and builds the hierarchical Web site data according to the directory structure of the URLs. It then calculates the positions of the resources by using the visualization algorithm [1,2], and displays the data as a sitemap. Simultaneously, it groups the extracted lines according to user-specified attributes and represents the access statistics by a bar chart. Figure 4 shows an overview of the described procedure.

The tool supports the following two operations for the exploration of access distributions:

[Operation 1] When a user clicks one of the groups on the bar chart, the tool counts the number of accesses in the group for each resource, and maps the counts to heights of the icons of the sitemap.

[Operation 2] When a user clicks one of the icons on the sitemap, the tool groups the accesses for the clicked resource according to user-specified attribute, and represents the access statistics for the resource by another bar chart.

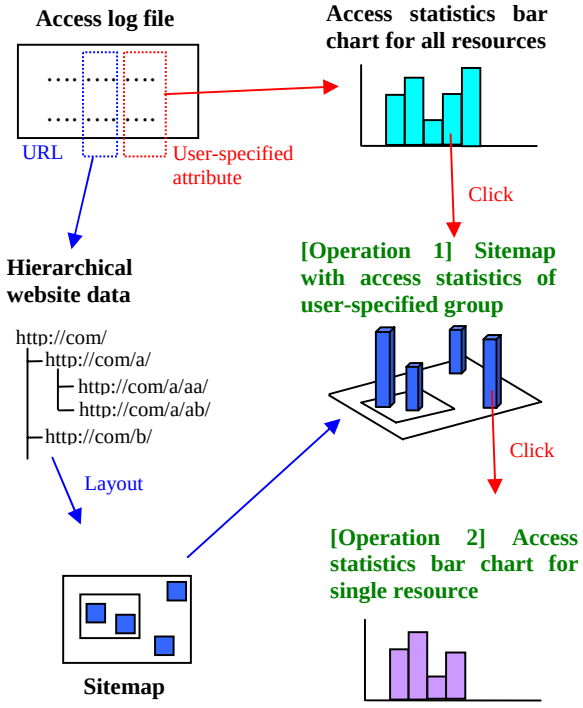

Figure 4. Overview of our website visualization tool.

In our experiment we combined the access log files of a research laboratory for one week, and visualized it using our tool. Figure 5 represents two kinds of the statistics of accesses as bar charts. The heights of the bars denote the number of accesses, and each bar is divided into 24 parts that denote each hour of the day. Figure 6 represents the sitemap generated by our hierarchical data visualization algorithm.

Figure 5(Left) shows that the number of accesses increased a great deal on the 7th day. When a user clicks one of the parts in the bar for that day as [Operation 1], the tool maps the access distribution in the clicked hour onto the sitemap. Figure 6 denotes the number of accesses in that hour for each Web page. Red icons denote the Web pages that had accesses in the hour, and their heights denote the number of accesses. Here we found two access characteristics in the figure. The upper green circle in the figure shows that most of Web pages in a particular API references directory were accessed during the hour. It is not easy to see such access patterns from traditional access ranking tools, but our system visually shows such characteristics. The lower green circle shows that a particular Web page was accessed very frequently in this hour. It was not the top page of the Web site, but just a Web page that describes a research project. By applying [Operation 2] and grouping accesses for the page according to their referrers, we found that the page was mentioned by a business newspaper Web site that day, and therefore many users accessed the URL directly. It was also interesting that we found that the Web page was active only in the daytime on the 7th day.

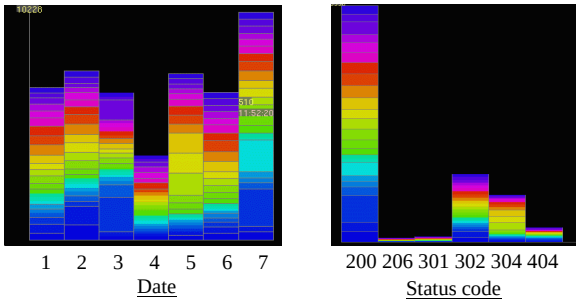

Figure 5. Examples of statistics of Web accesses represented as bar charts. (Left) 7 bars denote dates, and 24 colors denote hours. (Right) 6 bars denote status codes, and 24 colors denote hours.

Figure 6. (Upper) A sitemap with access distribution in one hour on the 7th day in Figure 5(Left). (Lower) A bar chart that represents the access statistics for the clicked resource grouped by their referrers.

References

[1] Itoh T., Kajinaga Y., Ikehata Y., and Yamaguchi Y., Data Jewelry-Box: A Graphics Showcase for Large-Scale Hierarchical Data Visualization, IBM Research, TRL Research Report, RT0427.

[2] Itoh T., Yamaguchi Y., Ikehata Y., and Kajinaga Y., Hierarchical Data Visualization Using a Fast Rectangle Packing Algorithm, IEEE Transactions on Visualization and Computer Graphics, in process.