# Visual Data Mining of Web Navigational Data

Jiyang Chen, Tong Zheng, William Thorne, Osmar R. Zaiane and Randy Goebel
Department of Computing Science
University of Alberta, Edmonton, Alberta, Canada
{jiyang, tongz, thorne, zaiane, goebel}@cs.ualberta.ca

## Abstract

*Discovering web navigational trends and understanding data mining results is undeniably advantageous to web designers and web-based application builders. It is also desirable to interactively investigate web access data and patterns, to allows ad-hoc discovery and examination of patterns that are not apriori known. Visualizing the usage data in the context of the web site structure is of major importance, as it puts web access requests and their connectivity in perspective. Various visualization tools have been developed for this task, but often fail to provide visual data mining functionalities to generate new patterns. Here we present our visual data mining system, WebViz, which allows interactive investigation of web usage data within their structure context, as well as ad-hoc knowledge pattern discovery on web navigational behaviour.*

*Keywords*—**Visual Data Mining, Web Visualization**

## 1 Introduction

In addition to the large growing volume of information explicitly accessible in the World Wide Web (WWW), there is also an increasingly rich and complex fabric of meta data. Not only does the hyperlink connections and structure induced by those connections provide another valuable information source, but the dynamic information resulting from the use of that structure and retrieval of information within that structure is an even richer knowledge resource. But that knowledge resource does not give up its insight easily, for it is not well-studied or well-understood, and there are few methods or measures that help provide insight into how the WWW is organized, how it is changing, or how it is used. There are some disciplined studies about the WWW (e.g., [13]), and even some interesting measures of the size of the Internet (e.g., [14]), both of which note the potential for the value of the meta information within the world's growing digital resource. However, it is clear that identifying and understanding that information requires new methods, most of which must exploit visualization to find and interpret interesting patterns.

This paper provides an overview of our development and application of visualization tools for both the structure of, and dynamic usage of the WWW. In our work on the development of web navigation tools (e.g., Navigation Compression Models [21]), and the development of algebras to manipulate and interpret dynamic usage data [4], we have found that the volume of dynamic data is so large that we had to develop methods that not just provide visual representation of compressed data, but also provide insight into the machine learning methods that we use to extract structure from that data. Our visualization system, called WebViz, combines a number of visualization and visual manipulation techniques from a broad range of existing web mining and visualization research. WebViz embeds a visualization algebra tested on our initial prototype [4] that provides an interface metaphor similar to those that have been long used by cartographer; this provides users with the ability to compute properties of aggregate web data, and then layer that information in a visual space.

The most important contribution of this paper is our development of a method for coupling the pattern generation with the visualization process, and propose novel operators to generate and distinguish knowledge patterns in their visualizations. We also present an improvement of the radial tree layout algorithm to minimize the occlusion problem. In an earlier paper [4], we proposed a visual data mining framework and a preliminary prototype to describe the data operations. Here, on the other hand, we extend the prototype with novel visual data mining operators, provide a more interactive user interface and improve the layout algorithm to use the available visual space. Contrary to existing visualization systems that only visualize data or mining results, our approach is novel in that data and patterns are visualized in their structure context, and interactive visual data mining operators are designed to manipulate data visualizations to discover interesting and implicit facts.

The remainder of the paper is organized as follows: We first present our visual data mining framework in Section 2. We then illustrate the visual data mining process with WebViz in Section 3, and discuss the user interface design

and implementation issues in Section 4 and Section 5. Finally we discuss pertinent related work in Section 6 before concluding in Section 7.

## 2 The Visual Data Mining Framework

As distinct from the definition in [1], which describes visual data mining as a step of the Knowledge Discovery and Data Mining (KDD) process, we define *Visual Data Mining* as a supplement for the KDD process: visual data mining combines data mining methods and computer-aided, interactive visual techniques in order to discover novel and interpretable patterns with the help of the human perception abilities. Visual data mining is not a required step, but can help data interpretation and mining in three different aspects of the KDD process: data integration, data mining, and pattern evaluation. Visual data mining can be seen as a visual hypothesis generation and verification process: visualizations allow the user to gain insight into the data, formulate new hypotheses, then verify them via visual data mining methods.

In addition to visualizing data from web access logs and patterns derived from web mining processes, the main motivation guiding the design of our system is to provide means to interactively manipulate visualization objects to perform ad-hoc visual data mining and interpretation of the discoveries. A prototype framework WebKVDS and a web mining operator algebra was proposed in [4]. WebViz is based on WebKVDS but implements more interactive visual functionalities and adds new operators to the algebra for visual data mining.

In our framework, the object used for both visualizing web usage information and expressing visual data mining operations is a web graph. A *web graph* is a multi-tier object that combines all relevant information, including the web site structure data, usage data, and knowledge patterns. The first tier, which we call *web image*, is a tree representation of the web site structure and is visualized as the background of the graph, also referred to as "bare graph." Each other tier, called an information layer or pattern layer, is a coherent collection of web data abstractions that can be laid over its context, which is the web image. These layers represent either pre-processed web usage statistics (e.g., page visits), or discovered patterns (e.g., association rules).

Combining layers with the web image means laying web usage data onto their structure context, and results in a corresponding web graph. The visualization of these data layers maps usage data to visual cues such as colour, size, shape of the nodes, as well as colour and thickness of links. The web image, representing the web connectivity structure, is always used as the background of the layers, allowing the localization of any information vis-à-vis their web context. By separating web data and mining results into different information layers, we are able to render each type of the usage data and pattern using different visual cues. Since each layer can be inhibited or rendered when the web graph is displayed, the user can select individual interesting layers to visualize and hide the others. Furthermore, showing several information layers that are mapped to different visual cues at a same time helps understand the implicit relations between usage data attributes.

We believe the idea of visualizing data and patterns in distinct layers on top of a representation of the structure of the web is useful, because it allows the display of information in context, which provides support for data interpretation and pattern discovery in web domain.

## 3 Visual Web Data Mining with WebViz

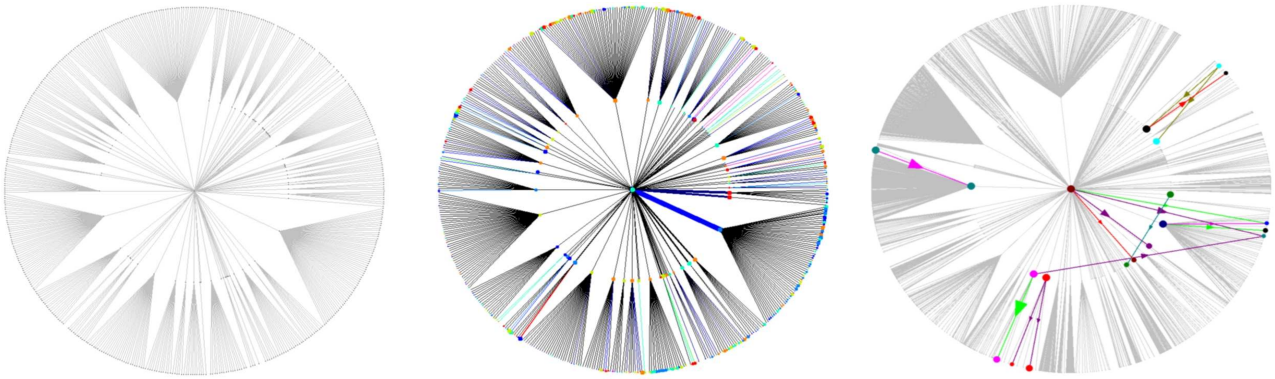
Knowledge patterns are sometimes too complex and abstract for people to understand, even with the assistance of visualization tools. Visual data mining tackles this task by enabling interactive human involvement with data mining methods to exploit visual perception. In WebViz, the web usage data and navigational patterns can be interpreted and manipulated in the visual context of the web connectivity structure, which is constructed from web access records. More importantly, we define visual data mining operators to interactively discover novel visual patterns from existing data and pattern layers.

### 3.1 Visualizing Web Usage Data and Patterns

We adapt the disk tree [6] or radial tree representation to visualize the web image (Figure 1 (a)), in which a node indicates a page and an edge represents the hyperlink that connects two pages; the tree root, a node located in the center of the image, is a selected starting page for the rendering algorithm. The radial tree visualization is a concentric circular hierarchical layout with a root node at the center. The nodes in an outer circle are children of those of the inner circle. We chose the disk tree representation because the screen space is used more efficiently than other layout methods, but we have also improved the algorithm for node layout on the concentric perimeter to minimize the occlusion problem. This is discussed in Section 5.

In the web domain, there are many possible data layers that can be extracted from the web access log dataset. We apply the methods of our previous work [21] to generate the usage data record and user sessions from web access logs. As a default, WebViz uses node size to represent the page visit count, node colour to indicate the average page view time, edge thickness to show the hyperlink usage count, and edge colour to represent the usage count percentage of a hyperlink (out of total count of the hyperlinks that share the same start page). Figure 1 (b) shows an example of the web graph with several information layers.

In addition to information layers that are extracted from web access logs, pattern layers, as outputs of different data



(a) Web Image (b) Web Graph with Information Layers (c) Web Graph with Pattern Layer

Figure 1: Visualizing Web Usage Data and Pattern: (a) visualizes the web structure using a radial tree representation. (b) superposes web usage on visualization of web structure. (c) visualizes the association rule layer on the web graph.

mining modules, can be visualized over the web structure as well, such as classes of pages, or clusters of pages or links. Association rules, which describe relations between items in a database of transactions, can be used to discover relationships between web pages or usage paths that are frequently accessed in a user session in the web context, e.g., an association rule $A \rightarrow B$ indicates the behaviour pattern of who visit page $B$ through page $A$. We use the method proposed in [7] to discover navigational association rules from user sessions. Figure 1 (c) shows a web graph with the association rule pattern layer, mapped to visual cues as edge colours and size arrow heads. For each rule, the colour of the edge indicates the confidence value of the rule and the size of the arrow head represents the support value of the rule. For rules that have multiple antecedent or consequent items, edges between any two pages are connected and display the same value. To provide more information of involved pages of the rule, one colour is used to draw both the antecedent and consequent nodes. The size of the node represents the visit frequency of the corresponding page.

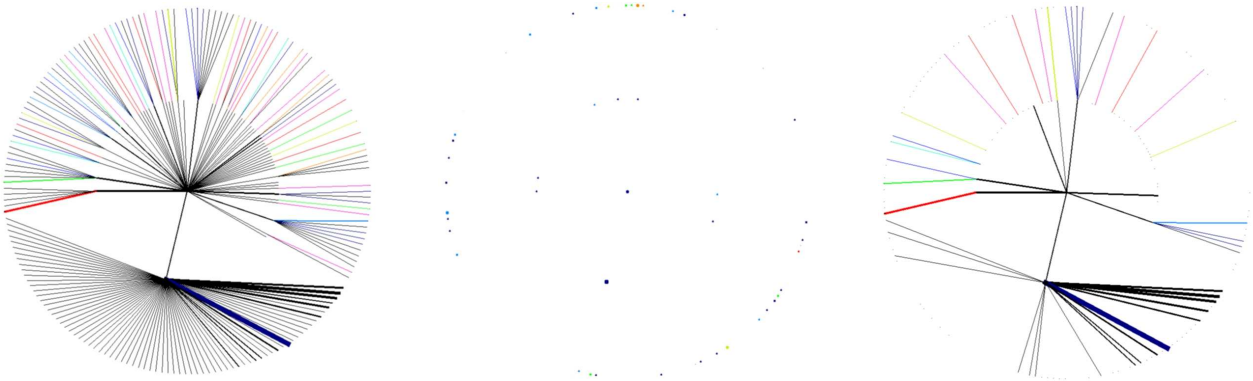
### 3.2 Visual Data Mining with WebViz Operators

Visual exploration methods provide a way to help humans interpret data and understand their meaning. However, visual data mining approaches discover interesting patterns based on existing data visualizations. We proposed a prototype framework in [4], and defined an algebra of visual data mining operators:

- Operator ADD sums the layer content for several graphs and displays the result for summary analysis.
- Operator MINUS generates the value difference between the same layer of different graphs and is mostly used to compare visualizations of different time periods.
- Operator COMMON selects the intersection of graphs and allow the user to combine information from different layers.
- Operator EXCEPT is the opposite of COMMON and represents the structure difference between graphs.
- Operator MINUS_IN and MINUS_OUT subtracts values across different information layers.

Details of these operators can be found in [4]. In WebViz, We have implemented all the operators listed above, as well as several new operators that we believe will provide better visual data mining.

Two new operators are CONNECT-TO and CONNECT-FROM, which manipulate the visualization based on web connectivity structure. They are used to identify and visualize nodes and edges that connect to (from) the visualized objects in the current displayed graph. For website analysis, it is important to check pages that are popular with respect to content and visit frequency, as well as pages that lead or follow these popular pages in a navigation session. The two operators are designed to emphasize *important pages*, which identify important content pages that are heavily visited, as well as *core pages*, which connect important pages. These notions are analogous to Authorities and Hubs [11] except that we consider click-stream visitation in addition to static links. An example of CONNECT-TO operator is shown in Figure 2. In the example, we first use filtering methods to produce a visualization for pages that have more than 500 visits in one particular month, then apply the CONNECT-TO operator to visualize the pages and links that lead the user to those pages in web navigation. Most of the pages that lead to highly visited pages are



Original web graph for one month.

Visualization after we filter out all nodes that have lower than 500 click count.

Visualization of pages and hyperlinks that leads to those popular pages using the CONNECT−TO operator.

Figure 2: Using CONNECT-TO operator to find "core pages" and "important pages": first use filtering methods to produce a visualization for pages that have more than 500 visits in one particular month, then apply the CONNECT-TO operator to visualize the pages and links that lead the user to those pages in web navigation.

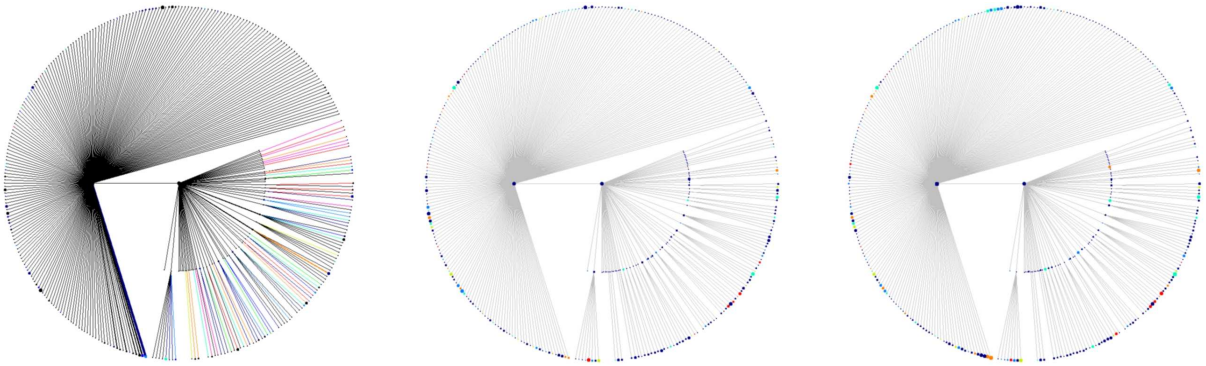
usually popular themselves. However, we may find some unexpected routes that surprisingly lead the navigation to "hot" pages or, in contrast, find routes that fail to do the job as designed. Such discovery can certainly help web administrators and designers improve the quality of their web sites.

Information layers in WebViz are extracted from web usage data. However, interpreting only usage data is not sufficient to discover all the implicit information. For example, page visit count and hyperlink usage data can be extracted and displayed, but information about web site entry page (where people enter the site and start the navigation) and exit pages (where people end the session and leave) are implicit but can not be discovered from what is visualized. Therefore, we develop a new operator VIRTUAL to add new layers, where "virtual" means "not from extracted data." Using WebViz, one can provide an expression to describe how to generate a new layer from provided data. After the user-defined operator is applied, the result is recorded as a new layer and mapped to a visual cue. The resulting layer can also be used as a normal layer for other operators in another "VIRTUAL" operation expression. Figure 3 shows an example that generates virtual web site entry and exit layers. In the example, we apply two "VIRTUAL" operations to create layers that record the frequency of pages used as site entry or exit. Two expressions (Figure 3) are used to describe the operations. The results are mapped to node size as visual cues. From those visualizations, we find that the entry and exit points are quite similar; in other words, pages that start a navigation are typically the last page of other navigations. However, these two layers are not identical; we can still find some differences at the top and bottom of the disktree circle. Overall, operator VIRTUAL allows users to create their own operators, and define data mining algorithms as operators to create new layers. For example, after incorporating data mining packages with WebViz, it is possible to define a virtual operator to cluster nodes based on a usage data layer of web pages, then map the result labels to different colours to display the clusters on the web structure. The cluster layer can be further used in other manipulations or "VIRTUAL" operations.

The WebViz operators are not just advanced data filtering methods, which select objects to visualize according to their attribute values. In WebViz, some operators, e.g. CONNECT-TO / FROM, select objects based on their structural connectivity; some operators, e.g. ADD / MINUS, manipulate layer values and rebuild the visualization; some operators, e.g. VIRTUAL, create new data layers based on existing ones by any user-defined method. These characteristics are not included in any filtering methods. Therefore, by combining WebViz operators together with visual exploration methods, our approach helps a user understand the data and its visualization, and further generate more specific and meaningful graphs in an interactive manner to better comprehend and interpret the navigational behaviour on a web site.

## 4 User Interface

We combined and implemented several interactive exploration methods that existing web visualization tools provide, such as zooming, rotation, filtering, search and legend edit. The user interface (UI) consisted of one graphical view window and three operational menus. It was designed



Original web graph for one month.

Substract sum of the usage of all incoming link from the visit count to generate virtual Entry layer: VisitCount – sum(InEdges–>LinkUsage) Layer is represented by node size.

Substract sum of the usage of all outgoing link from the visit count to generate virtual Exit layer: VisitCount – sum(OutEdges–>LinkUsage) Layer is represented by node size.

Figure 3: Create and Compare Virtual Site Entry and Exit Layers: apply two "VIRTUAL" operations to create layers that record the frequency of pages used as site entry or exit. Two expressions are used to describe the operations. Results are mapped to node size.

to help the user to interpret and discover web usage data and patterns.

Figure 4 shows the graphical view with menus. A radial tree representing the web structure and visual cues representing information layers are visualized in the left window. A brief description and graph ID is shown on the left top corner. The status bar shows the connection status with a data server, and the information of the currently selected object. The loading bar is located in the right bottom of the window, and indicates progress of loading remote data from the server. We classify WebViz functionalities into three categories and display them in three tabs:

- Visual Tab: A user can click on the legend to insert, delete or modify the value range and select preferred colour, change visual cue-layer mapping, or select layers to visualize.
- Operator Tab: Visual data mining operators are listed. A user can click any of the buttons and corresponding menus will pop-up to offer more detailed selection options.
- Information Tab: The context window for zooming is at the top of this tab. An user can click on the context window to change the position of the radial tree drawn in the graphical view window. The center window in the info tab displays the result of node searching. The URLs of search results are displayed here and corresponding nodes in the graphical view window will display as big yellow dots (e.g., the root node in Figure 4). The window at the bottom of the tab shows the detailed information about the selected object.

We also provide several mouse-controlled interaction functions. For example, users can adjust the mouse wheel to zoom in or zoom out, right click to bring out a menu of common functionalities, or left click to select a node or an edge.

## 5 Implementation

WebViz is implemented in Java and is designed for use within a client-server model. The server holds all usage data, web structure data, discovered patterns and meta files. A Java applet client can connect to the server, and select the meta description to load data for visual data mining. By modifying specific meta files, users can personalize their own representation preferences without impacting others or changing the source. After the data is loaded, the user can apply interactive operators to generate interesting patterns based on their understanding of the visualization, then record the result on the server for future reference. A client demo is available online as a Java Applet Application in [18].

### 5.1 System Performance

The web usage and structural data we used are generated from monthly server-produced access logs of our department website. The data preparation [21] consists of crawling the website structure, cleaning irrelevant records, and breaking the access log into user sessions. In order to cut usage records into sessions, we identify users by their authentication or cookie value, then exclude all other

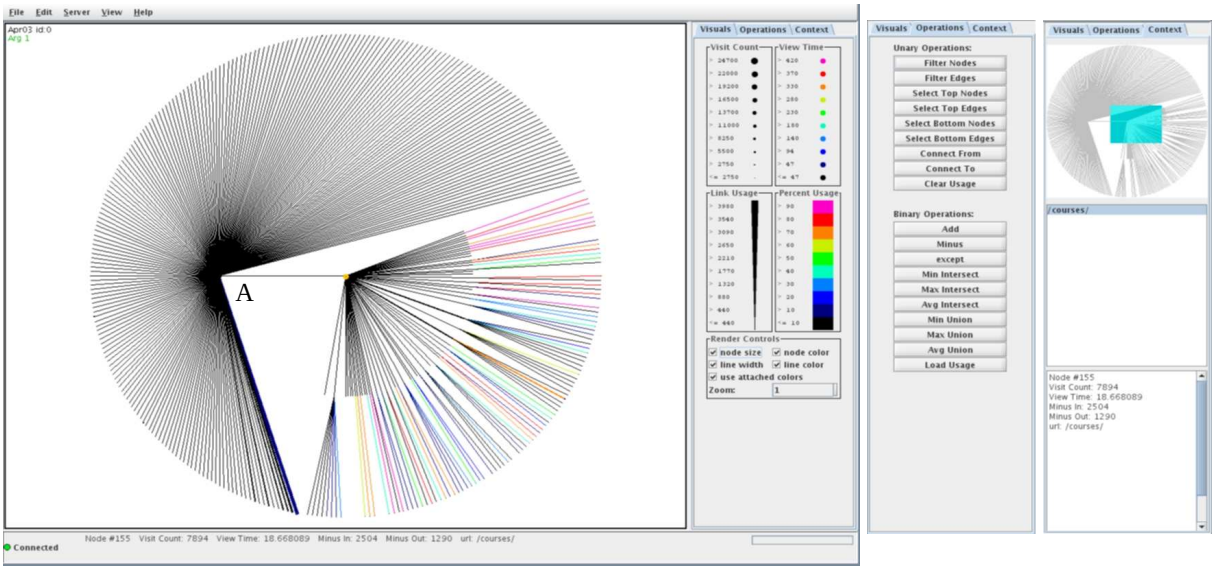



Figure 4: Snapshot of WebViz System and Functionality Tabs

records without both authentication and cookie. The session timeout is set to be 30 minutes. A constraint-based frequent itemset discovery algorithm [7] is applied on these navigational sessions to generate focused associations between web pages.

WebViz is efficient in handling large data. It usually takes the system less than five seconds to generate a disk tree of our complete department site, which contains more than 100,000 web pages. The node query functionality can also be done in seconds. For data sets that have more than millions of nodes, WebViz users can apply filtering operations as well as reasonably set the root and depth of the disktree to achieve fast visualization.

### 5.2 Improvements to the Disktree Layout Algorithm

The WWW is made up of web pages and hyperlinks. Since one page usually connects to many other pages, the structure is more like a connected graph than a hierarchical tree. In order to visualize the web structure, the Disktree algorithm [6] addresses the problem by only showing one link leading into a page: the algorithm selects a "parent" for a certain page from the pages that connect to it. The original disktree algorithm used Breadth First Search and selected the first link that appeared in the data scan. The Usage Based Search [4] improved that algorithm by selecting the link with the highest visit frequency. Both of these algorithms are provided in WebViz.

While web structure is converted into a tree by the Disktree algorithm, web pages do not have any preferred positions in a visual space. To visualize the structure, the disktree method puts the root node in the center then draw nodes in different radius around the root node. Children nodes of the root are drawn in the first level, children of those nodes are located in the next level and so on. For each level, locations of nodes are decided by the total number of nodes of that level $n$, on an angle that determined by $\frac{360^o}{n}$. However, when a node has many more children than other nodes on the same level, this method may cause severe occlusion problems, e.g., in Figure 4, the number of children of node $A$ is more than half of the nodes on the second level. If angles of the nodes on the first level were distributed by the original method, edges from $A$ to its children would overlap with other nodes. In order to minimize occlusions, we have implemented an alternative layout method for the disktree structure. Instead of calculating angles for each level, we distribute the 360 degree among those "dead-end" nodes, which are either located in the outmost level or have no children. Then we recursively compute the angle for nodes of inside levels until we reach the root using the formula: $A_P = \frac{A_{min}+A_{max}}{2}$, where $A_P$ represents the angle for the parent, $A_{min}$ and $A_{max}$ represent the minimum and maximum angle of its child nodes. Our new layout method assigns more visual space to bigger subtrees and effectively solves the occlusion problem of visualizing the tree structure.

## 6 Related Work

As defined in [1], visual data mining uses visualization as a communication channel between the computer and the user, to produce novel and interpretable patterns. There are three classes of visual data mining.

- Visualization of data mining results. Extracted patterns are visualized to make them more interpretable.
- Visualization of the data mining process. The process of a mining algorithm can be visualized to help the discovery.
- Visualization of the data. Data is visualized before a mining algorithm is applied.

While there are many systems [6, 15, 16, 19, 20] that focus on visualizing data mining results and the source data, Ankerst et al. [2, 3] developed the PBC system to visualize decision tree construction and the data classification process. J. Han et al. proposed the RuleViz model [10] and developed the DTViz [9] and CVizT [8] systems to visualize the decision tree and classification rule construction process to use visual space in a more efficient way.

In the WWW domain, visual data mining approaches concentrate on representing the relationship between web usage data and web structure. Ed Chi et al. [6] proposed the Disktree representation to display the usage data and structure information. The graph has been used to visualize web site evolution, web usage trends over time, and evaluation of information "foraging" [5]. The WebKIV [15] system implemented the disktree representation to compare web navigational patterns and defined a three dimensional scale to describe the web visualization task. A visual web mining prototype framework was designed in [4] to describe the object manipulation for visual data mining purposes. Youssefi et al. [19] implement a visual web mining system using 3D representation, but severe occlusion problems make the approach impractical. A recent visual data mining system, WebPatterns [16], focuses on visualizing web usage associations, sequences, and network analysis. Most of these systems apply the idea of visualizing usage data using visual cues of the structure object, however, more interactive operations can be added to describe various possible visual data mining manipulations.

Web structure visualization is paramount for a visual web mining system to provide a concise overview and background for usage data details. One of the most popular methods to visualize web structure is Chi's radial disktree representation [6]. Since the radial tree layout uses screen space more efficiently than other layout methods, many systems [4, 5, 15, 16] adapted the disktree method to visualize web structure. Other similar approaches include the ConeTree [17] and the Hyperbolic tree [12]. As we have discussed in Section 5, a hierarchical tree representation drops many hyperlinks in order to transform the web connectivity graph into a tree, which can suffer object occlusion when dealing with large numbers of nodes.

# 7 Conclusions

We present our visual data mining approaches to interpret the data we extracted from web access logs. We extend our visual data mining prototype framework to visualize and mine web usage data to understand web page access behaviours vis-à-vis the connectivity structure. We describe the concepts of web image, information layers and web graphs, and present the idea of mapping data and pattern to visual cues as distinct layers on top of a radial tree representation of the web structure, which allows in-context information display. We further provide visual data exploration and mining operators in our WebViz system for data interpretation and knowledge pattern discovery. Our visual data mining system can visualize multi-layer web graphs and, with the help of these operators, provides a powerful tool for interactive visual web mining.

# Acknowledgments

Our work is supported by the Canadian Natural Sciences and Engineering Research Council (NSERC), by the Alberta Ingenuity Centre for Machine Learning (AICML), and by the Alberta Informatics Circle of Research Excellence (iCORE).

# References

[1] M. Ankerst. *Visual Data Mining*. Ph.D Thesis. Institute for Computer Science, University of Munich, 2000.

[2] M. Ankerst, C. Elsen, M. Ester, and H. Kriegel. Visual classification: An interactive approach to decision tree construction. In *ACM SIGKDD 1999*, pages 392–396.

[3] M. Ankerst, M. Ester, and H. Kriegel. Towards an effective cooperation of the user and the computer for classification. In *ACM SIGKDD 2000*, pages 179–188.

[4] J. Chen, L. Sun, O. R. Zaïane, and R. Goebel. Visualizing and discovering web navigational patterns. In *Seventh ACM SIGMOD International Workshop on the Web and Databases (WebDB 2004)*, pages 13–18.

[5] Ed H. Chi. Improving web usability through visualization. *IEEE Internet Computing*, 6(2):64–71, March/April 2002.

[6] Ed H. Chi, J. Pitkow, J. Mackinlay, P. Pirolli, R. Gossweiler, and S. K. Card. Visualizing the evolution of web ecologies. In *Proceeding of CHI*, 1998.

[7] M. El-Hajj and O. R. Zaïane. Non recursive generation of frequent k-itemsets from frequent pattern tree

representations. In *Proc. of 5th International Conference on Data Warehousing and Knowledge Discovery*, September 2003.

[8] J. Han and N. Cercone. Interactive construction of classification rules. In *PAKDD 2002*, pages 529–534.

[9] J. Han and N. Cercone. Interactive construction of decision trees. In *PAKDD 2001*, pages 575–580.

[10] J. Han and N. Cercone. Ruleviz: A model for visualizing knowledge discovery process. In *ACM SIGKDD 2000*, pages 223–242.

[11] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[12] J. Lamping, R.Rao, and P.Pirolli. A focus+context technique based on hyperbolic gemetry for visualizing large hierarchies. In *ACM SIGCHI*, pages 401–408, May 1995.

[13] S. Lawrence and C.L. Giles. Accessibility of information on the web. *Intelligence*, 11(1):32–39, 2000. Reprinted from Nature, 400, 107-109, 1999.

[14] The internet mapping project. http://www.cheswick.com/ches/map/index.html http://research.lumeta.com/ches/map/gallery/index.html.

[15] Y. Niu, T. Zheng, J. Chen, and R. Goebel. Webkiv: Visualizing structure and navigation for web mining applications. In *Proceedings of IEEE Web Intelligence Conference (WIC)*, 2003.

[16] C. Oosthuizen, J. Wesson, and C. Cilliers. Visual web mining of organizational web sites. In *IV '06: Proceedings of the conference on Information Visualization*, pages 395–401.

[17] G. Robertson, J. Mackinlay, and S. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *Proc. of ACM SIGCHI conference on Human Factors in Computing Systems '91*, pages 189–194.

[18] WebViz: http://kingman.cs.ualberta.ca/research/demos/content/webviz/demo/src/front/WebVizClientApplet.html.

[19] A. Youssefi, D.Duke, M.Zaki, and E.Glinert. Toward visual web mining. In *Proceeding of Visual Data Mining at IEEE Intl Conference on Data Mining (ICDM), Florida*, 2003.

[20] K. Zhao, B. Liu, T. M. Tirpak, and W. Xiao. A visual data mining framework for convenient identification of useful knowledge. In *ICDM '05*, pages 530–537.

[21] T. Zheng, Y. Niu, and R. Goebel. Webframe: In pursuit of computationally and cognitively efficient web mining. In *PAKDD 2002*, pages 264–275.