Improving the Quality of Visual Web Browsing by Using Weighted Graph Drawing

Mao Lin Huang
Department of Computer Systems
The University of Technology, Sydney
NSW 2007, Australia.

Binyang Zhou
Department of Mathematics and Computing
University of Southern Queensland
Toowoomba, QLD 4350, Australia

Abstract *Visual Web Browser[1] provides an on-line graphical views for users to explore the web-space dynamically without requiring the whole web graph to be known. It allows the user to graphically snoop the World Wide Web via a sequence of successive visual maps of web structure, where the visual node corresponds to a web page and the arc corresponds to a hyperlink interconnected with web pages.*

However, the preliminary version of Visual Web Browser can only display visual nodes with the same size, regardless of the importance of nodes in the web locality. The user has no knowledge of what nodes are landmark nodes and more important to them, and what nodes are less important to them while surfing through the web.

This paper discusses a new method of applying Weighted Graph Drawing in our Visual Web Browser. Under the weighted graph model, we use different size of nodes to indicate the importance of the node in the web locality. Thus, the user can gain the knowledge of importance of nodes while he/she is browsing through the web-space. This gives the user some ideas of where he/she is in the web locality and helps the user to make up the decision of where to go next during the navigation.

Keywords: information visualization, web navigation, web browsing, web graph and graph drawing.

1 Introduction

The incredible size of the web, accompany with its inherent lack of structure both at the inter- and intra- document levels, introduces great challenges for information discovery. The web has no navigation structure or any sort of complete index of the content available.
Because the web has now become so large and so entrenched, the user can become disoriented while navigating the available web space. If, while surfing through the web, the user comes to a particular web page and feels lost, some ideas about the position of this page in the overall web space will help to orient the user. Similarly, when the user specifies the URL to jump to a particular page using a web browser, some visual senses about the location of this page in the web space will be very helpful.

One of the possible solutions we used to provide users with the sense their position is the use of *Visual Web Browser* [1].

The essential technique adopted in *Visual Web Browser*, known as *OFDAV* [2, 3], provides a major departure from traditional visualization methods. It allows the user to visualize the entire web space that is available through the hypermedia system, but does not require the whole web graph to be known.

In *OFDAV*, the user's view is focused at any point in time on a small subset of the entire web space. The corresponding sub-graph of this subset is called a logical *viewing frame* and is defined by its *focus nodes*. Conceptually, the *focus nodes* form a first-in-first-out queue with user's highest interest focus. The *viewing frame* is updated smoothly following the changes of the user's interest focus.

OFDAV uses a *force-directed* algorithm [4, 5] to draw sub-graphs and the logical neighborhoods around the sub-graph.

Each *viewing frame* shows the detail of a particular node; nodes in the immediate neighborhood, those that can be reached from the document are show. This would be very useful for the user to identify where he/she is in the web locality. This is similar to a common geographical navigation strategy: a lost person will try to find where he/she is using her immediate neighborhood and important geographical landmarks.

However, the preliminary version of the *Visual Web Browser* can only display visual nodes with the same size, regardless of the importance of nodes in the web locality. The user has no knowledge of what nodes are landmark nodes and more important to them, and what nodes are less important to them while surfing through the web.

This paper discusses a new method of applying *Weighted Graph Drawing* in *our Visual Web Browser*. Under the weighted graph model, we use different size of nodes to indicate the importance of the node in the web locality. Thus, the user can gain the knowledge of importance of nodes while he/she is browsing through the web-space. This helps the user to make their decision of where to go next during the navigation.

There are many examples of applying weighted graph model in the area of information visualization. For example, a weighted graph model was proposed in [6, 7] to represent email relationships: if two users *u1* and *u2* exchange email frequently, then the weight on the edge (*u1*, *u2*) is small; otherwise the weight is large. Further applications of weighted graph drawings are given [8, 9].

The more important of a node, the larger of it's visual representation. The size of a node displayed in the *viewing frame* is purely up to the importance of the node in the web locality. We use three metrics $\delta_1(v)$=*Connectivity*, $\delta_2(v)$=*Access frequency* and $\delta_3(v)$=*Depth* to measure the importance of a node v (a web page) in the web locality.

We then calculate the size of each node image, based on metrics associated with this particular node.

2 A weighted graph model and On-line graph drawings

A *graph* consists of a finite set N of nodes and a finite set E of edges, where each edge is an unordered pair of nodes of graph G. A node μ is said to be *adjacent* to a node v if (μ, v) is an edge of G; in this case, the edge (μ, v) is said to be incident with μ and v.

A *straight line* drawing of a graph $G = (N, E,)$ is a function $D: N \rightarrow R^2$ that associates a drawing $D(v)$ to each node $v \in N$. Since all drawings in this paper are straight line drawing we omit the term "straight line".

In the actual layout creation, a drawing $D(v)$ of a node $v \in N$ is normally represented by a graphic box (perhaps enclosing some text) appearing on the screen with the position (x_v, y_v) at the center of the box, where (x_v, y_v) are the pixel coordinates of a reference point of the node. Therefore, there are two additional graphic attributes h_v and w_v associated with each drawing $D(v)$, where h_v represents the height of the graphic node and w_v represents the width of the graphic node.

A *weight* $W(v)$ is a non-negative real value associated with a node v of a graph. A weighted graph $G = (N, E, W)$ consists of a graph $G = (N, E)$ and a weight $W(v)$ for each $v \in N$. A *weight* $W(v)$ may be divided into a set of sub-weights, $W_1(v)$, $W_2(v)$, ..., and in this case the value of $W(v)$ is the sum of these sub-weights.
That is,

$$W(v) = \sum_{v \in N}^{i=1} W_i(v) \qquad (1)$$

where $i = 1, 2, \ldots$ n. Suppose that $G = (N, E, W)$ is an weighted graph and $W: N \rightarrow R^+$ assigns a weight to each node of G. Graph drawing functions may be required to draw G so that the height h_v and width w_v of the node image of a node $v \in N$ depend on the values of the weights $W_1(v)$, $W_2(v)$, ... If we draw the graphic node as a square, then the height h_v and width w_v of a node image can be calculated as following,

$$W_i(v) = (\delta_i(v)/max_{\delta i})*C_i \qquad (2)$$

Where $\delta_1(v)$, $\delta_2(v)$ ... are metrics used to measure the importance of node v (a web object) in the web locality. The max_δ is the maximum value of the variable $\delta_i(v)$. Where the C_i is a real value and the sum of C_1, C_2 ... must equal to one, that is,

$$\sum^{i=l} C_i = 1.$$

Thus from Eq. (1) and Eq. (2), we have

$$h_v = w_v = W(v) * S_{max} = \sum_{v \in N}^{i=l} W_i(v) * S_{max} \qquad (3)$$

Here S_{max} is the maximum value for h_v and w_v.

The visual exploration of the web graph is based on the online graph model defined in [3]. The web journey proceeds by visualizing a sequence $F_1=(T_1, Q_1)$, $F_2=(T_2, Q_2)$ ... of logical frames. Each logical frame $F=(T, Q)$ consists of a spanning tree T of the union of the neighborhood trees and a queue Q of focus nodes. To limit the size of these frames, we assume a upper bound B on the number of focus nodes (clicked hyper-documents) in each frame, that is, $|Q_i| <= B$ (i > 1), where $|Q_i|$ denotes the number of nodes in Q_i. To obtain F_{i+1} from F_i, the user selects a non-focus node u in F_i with a mouse click, if u has more than one neighbor, then it becomes a new focus node of F_{i+1}. This is analogous to clicking on the text (within a document) which represents a hyperlink.

If $|Q_i| = B$, we delete an old focus node v from Q_i, that is, the queue of focus nodes in F_{i+1} becomes $Q_{i+1} = Q_i + \{u\} - \{v\}$. For the Visual Web Browser [1], we choose v to be one of the focus nodes in Q_i whose graph-theoretical distance from u is the largest in F_i. Using these focus nodes, we can easily calculate F_{i+1}.

The detail of *On-line Graph Drawing* is described in [2, 5]. The layout of drawings D_i must satisfy the usual readability requirements of graph drawing [10]. Experience has shown that traditional force-directed algorithms are moderately successful in drawing this kind of graphs. However, for on-line graph drawing, we have some further requirements. Therefore, we use a "modified spring algorithm" to draw the web graph that achieves the extra requirements. The detail of *Modified Spring Algorithm* is described in [2, 5].

3 The method for calculating the importance of nodes

In this section, we discuss the calculation of importance $W(v)$ for each node v in *the viewing frame*. Once the value of importance $W(v)$ is calculated, we then can easily calculate the height h_v and width w_v of the node image for visualization using Eq. (3).

We may use three metrics $\delta_1(v)$=*Connectivity*, $\delta_2(v)$=*Access frequency* and $\delta_3(v)$=*Depth* to measure the importance of a node v (a web page) in the web locality. A *Connectivity* metric indicates that how a node is connected to other nodes in the web space. An *Access frequency* metric indicates how many times the node has been accessed in the recent time period and a *Depth* metric indicates at what depth the node resides in the file system hierarchy of the web locality.

- **Connectivity**

The importance of a node could be reflected by the connectivity of the node. This means if a node is connected to more other nodes in the web space, then we say this node is to be more important in the web locality. To find out the connectivity of a node the hyperlink structure (the topology) among the web pages needs to be extracted. We need to use some structure analysis tools, such as **Harvest** [xx], to extract the linkage information for the nodes (pages) that are displaying in the current *viewing frame*.

We use two terms, *outdegree* and *indegree* to measure the connectivity of a node. The *outdegree* of a node is the number of other nodes that can be directly reached from this node with the graph theoretic distance of 1. Similarly, the *indegree* of a node is the number of other nodes that can directly reach this node with the graph theoretic distance of 1.

- **Access frequency**

The importance of a node could also be reflected by the access frequency of the node. This means that if a node (page) had more visitors in a periodical time, then we say this node is to be more important in the web locality. This information can be obtained by analysis of the log files. For each node we may calculate the number of times it was accessed in the preceding month. To keep the access frequency uptodate this process needs to be done periodically.

- **Depth**

The importance of a node could also be reflected by the position of the node in the hierarchical file system of a web site. Generally, when a web site is developed, the file system hierarchy is formed so that we assume that the important nodes are normally higher up in the hierarchy. For example, in the Faculty of Information Technology site at UTS, the top directory of the web site has the home page of the Faculty.

These pages in the top directory of a site normally give more general information and therefore more important for understanding a web locality are higher up in the hierarchy than the pages containing detailed specific information. Because of this, the depth of a node can also be used to determine the importance. The depth of a node can be determined from the URL. For example, http://www.it.uts.edu.au/ has a depth of 1 where http://www.it.uts.edu.au/department/ has a depth of 2.

- **The calculation of importance**

Suppose that

- I(v) = the indegree of node v
- O(v) = the outdegree of node v
- Fr(v) = the access frequency of node v
- D(v) = the depth of node v

We divide the importance $W(v)$ of node v into three parts, the *connectivity part* $W_1(v)$, the *access frequency part* $W_2(v)$ and the *depth part* $W_3(v)$, defined as follows:

$$W_1(v) = ((I(v) + O(v)) / Max_{\_value1}) * C_1$$

where $Max_{\_value1}$ is the maximum value for $I(v) + O(v)$,

$$W_2(v) = (Fr(v) / Max_{\_value2}) * C_2$$

where $Max_{\_value2}$ is the maximum value for $Fr(v)$, and

$$W_3(v) = (1/D(v)) * C_3$$

here $C1 + C2 + C3 = 1$. The overall importance of a node v is the sum of these three values, that is,

$$W(v) = W_1(v) + W_2(v) + W_3(v),$$

it is a number between 0 and 1. The parameter *C1*, *C2* and *C3* determining the importance of connectivity, access frequency and depth that can be controlled by the user.

4 Drawing of the graphic node images

We use Java programs to implement *OFDAV* and draw the graphic node images in each *viewing frame*. Suppose that a node image is a graphic square appearing in a Java Applet. We define each node image as follows:

$$Image(v) = f\,(h_v, w_v, x_v, y_v)$$

Where $h_v = w_v$ represents the size of node image and (x_v, y_v) are the pixel coordinates of the center point of the node image. Next step is to set up a maximum value S_{max} and a minimum value S_{min} for h_v and w_v .for h_v and w_v . For example, we may set $S_{max} = 60$ and $S_{min} = 20$.

Thus, we can calculate the size h_v and w_v of a node image, base on the value of the importance of the node v, that is,

$$h_v = w_v = \begin{cases} S_{min}, & \text{if } w_v < S_{min} \\ W(v) * S_{max} & \text{otherwise} \end{cases}$$

or

$$h_v = w_v = \begin{cases} 20 & \text{if } w_v < 20 \\ W(v) * 60 & \text{otherwise} \end{cases}$$

We can then use one of the graphic functions in Java to draw this node image, once we get the actual value of h_v and w_v as shown above, that is

$$g.drawRect\,(x_v - w_v/2,\ y_v - h_v/2,\ w_v - 1,\ h_v - 1)$$

4.1 An example of a weighted drawing of web graph

Figure 1 shows the web graph of a retrieved page, http://www.it.uts.edu.au/ . This is the home page of the Faculty of Information Technology, University of Technology, Sydney. This graph is drawn using a modified spring algorithm (a kind of force-directed drawing), without applying any kind of weighted drawing for node images. In the graph, we can only see the hyperlinks (relationships) among the pages (nodes), and cannot see the importance of each node in the web locality. Therefore, the user gains no knowledge of what nodes that are more important in the web localities and worthwhile to click into it having a look.

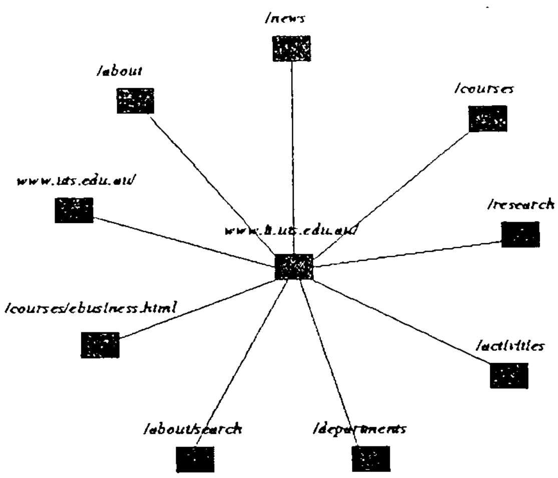

Figure 1: A drawing of a web graph without applying of a weighted node layout.

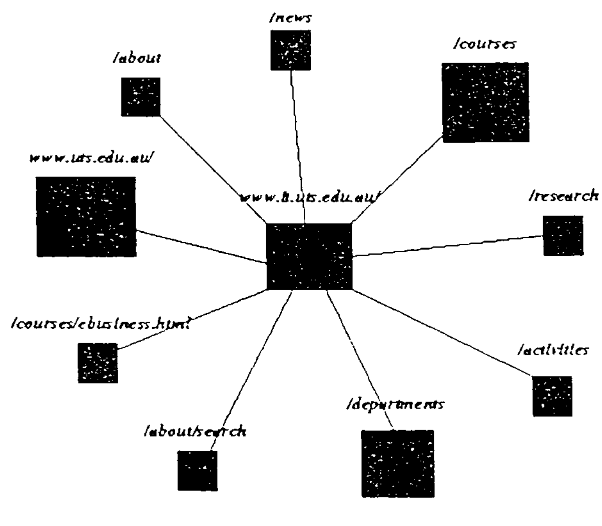

Figure 2: A drawing of the same web graph with applying of a weighted node layout. The user can easily see the importance of nodes in the graph.

Figure 2 shows the layout of the same web graph as shown in Figure 1. However, this layout represents node images with different sizes that indicate the importance of nodes in the web localities. This gives the user a clear idea of what nodes that are more important in the web localities and worthwhile to have a look.

For example, the node "*www.uts.edu.au/*" has the biggest image box in the *viewing frame* because it's the home page of the University and it's access frequency is very high. The node "*/courses*" also has a large image box because this page contains the course information, so it is heavily accessed by a large number of students.

5 Conclusion

Using weighted graph layout for web visualization may improve the quality of visual web browsing. Under this scheme the user can gain the knowledge of importance of each node in the web localities, while surfing through the web. This helps the user to make the decision of where to go next while he/she is interactively browsing through the web via the visualization (*viewing frames*).

In this work, we use only one of the visual attributes of nodes (the size of node image) to indicate the difference of these nodes in the web space. However, a graphic node is usually associated with more than one visual attribute, such as the *color* and *shape* of the nodes. These attributes may also be used to present the different roles of nodes in the web localities. I would like to do some work on this topic in the near future.

References

[1] M. L. Huang, P. Eades, W. Lai and Y. Zhang. *Visual Web Browser - Mapping and Browsing the Web with a Sense of "Space".* In the poster proceedings of 8th International World Wide Web Conference (WWW8), page 86-87, at Toronto, Canada, 1999.

[2] M. L. Huang, P. Eades, and J. Wang, *Online Animated Visualization of Huge Graphs using a Modified Spring Algorithm,* Journal of Visual Languages and Computing (Academic Press), vol 9, no.6, pp. 623-645, ISSN 1045-926X, 1998

[3] M. L. Huang, P. Eades, and R. F. Cohen, *WebOFDAV - Navigating and Visualizing the Web On-line with Animated Context Swapping.* Computer Networks and ISDN Systems (Elsevier Science B.V.), vol 30, no.1-7, pp. 638-642, ISSN 0169-7552, 1998

[4] P. Eades, *A Heuristic for Graph Drawing* Congressus Numerantium, vol 42, pp. 149–160, 1984.

[5] M.L. Huang, P. Eades, and J. Wang, *Online animated graph drawing using a Modified Spring algorithm,* in: Proc. of the 21st Australasian Computer Science Conference (ACSC'98), 1998, pp. 17–28.

[6] P. Eades, W. Lai, and X. Mendonca, *A Visualizer for E-mail Traffic*, In 4th Int. Conf. Proc. Pacific Graphics'94 / CADDM'94, pages 64-67, 1994.

[7] Wei Lai, *Building Interactive Diagram Applicatio.*, PhD thesis, University of Newcastle, Australia 1993.

[8] T. Kamada. *Visualizing Abstract Objects and Relations.* World Scientific Series in Computer Science - vol 5, 1989.

[9] T. Kamada and S. Kawai, *Automatic Display of Network Structures for Human Understanding.* Technical Report 88-007, Department of Information Science, Faculty of Science, University of Tokyo, 1988.

[10] G. Di Battista, P. Eades, R. Tamassia & I. G. Tollis, *Algorithms for Drawing Graphs: An Annotated Bibliography.* Computational Geometry Theory and Applications, vol 4, pp 235-282.