Börner, Katy and Zhou, Yuezheng. (2001) A Software Repository for Education and Research in Information Visualization. Information Visualisation Conference, London, England, July 25-27, pp. 257-262.

A Software Repository for Education and Research in Information Visualization

Katy Börner
Indiana University, SLIS
Bloomington, IN 47405 USA
E-mail: katy@indiana.edu

Yuezheng Zhou
Indiana University, Computer Science
Bloomington, IN 47405 USA
E-mail: yuzhou@cs.indiana.edu

Abstract

This paper argues for the creation of a software repository for research and education in information visualization (IV). It starts with an introduction and overview of IV online resources and software repositories. Next, we introduce the code repository we created and demonstrate how it was used in the IV course one of the authors teaches. Sample implementations by students are presented as well. We conclude with a discussion of ways to grow and maintain the software repository and invite contributions.

1. Introduction

Information Visualization (IV) combines aspects of scientific visualization, human-computer interaction, data mining, imaging, and graphics, among others. The field is about 10 years old and fast-growing. Recently, a number of excellent textbooks have been published by Card et al. (1999), Chen (1999), Ware (1999), Spence (2000), and Dodge & Kitchin (2000). However, 2-dimensional printouts on paper often cannot convey the true visual appearance and interactive performance of certain visualizations. Therefore, several books come with accompanying web sites containing screen-sized snapshots of user interfaces as well as animations and movies. However, we believe that education and research on IV could be considerably enhanced if a general IV software repository was created. The repository would not only facilitate sharing, evaluation, and comparison of algorithms and software but also reduce the time and effort spent for repeatedly re-implementing algorithms.

Data or software code repositories are frequently used in other research communities. For example, the *CMU Artificial Intelligence Repository* provides access to a large number of Artificial Intelligence (AI) software packages at http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/0.html. More *AI Programming Resources* are accessible at http://www.cs.berkeley.edu/~russell/prog.html. In addition, a large number of *Artificial Intelligence* books come with software code. For more information see http://yoda.cis.temple.edu:8080/UGAIWWW/books/.

In physics research, many software repositories exist through which researchers all over the world can share code. One example is the CERN Program Library at http://wwwinfo.cern.ch/asd/. It is large collection of general purpose programs maintained and offered in both source and object code. Researchers and students are encouraged to use this code repository rather than their own code. Aside from saving users time and effort, the repository code is more likely to be correct after having been tested by many other people.

The aim of this paper is to generate discussion about a non-commercial IV software repository that can be used to help collectively understand the issues underlying IV and to pool existing and future IV resources.

The paper starts with a review of existing IV online resources and code repositories that are known to us. Subsequently, we introduce the software repository that we designed to teach the Information Visualization course at Indiana University, Bloomington (IUB). Results of class projects accomplished by students who used the software repository are discussed. The paper concludes with a discussion and an outlook.

2. Information visualization resources and software repositories

During our online search for public domain source code that could be used to teach IV, we encountered many excellent information visualization resources. Among them are:

- OLIVE: On-line Library of Information Visualization Environments (http://www.otal.umd.edu/Olive/) is a great IV resource created as a class project for Ben Shneiderman's fall 1997 CMSC 828/838S graduate course on Information Visualization at the University of Maryland, College Park, Department of Computer Science.
- Martin Dodge’s An Atlas of Cyberspaces (http://www.cybergeography.org/atlas/atlas.html)

provides a continually growing, up-to date overview of many graphic representations of the geographies of the new electronic territories of the Internet, the World-Wide Web and other emerging Cyberspaces.
- Ivan Herman, M.Scott Marshall, and Guy Melançon's IV links (http://www.cwi.nl/InfoVisu/links.html).
- Gary Ng's collection of IV links that can be found at http://www.cs.man.ac.uk/~ngg/InfoViz/.

In addition, several research groups provide access to their software:

- The Human-Computer Interaction Lab (HCIL) provides access to a number of IV resources http://www.cs.umd.edu/hcil/research/visualization.shtml such as Jazz[1] (http://www.cs.umd.edu/hcil/jazz/), a toolkit that creates robust, full-featured graphical applications in Java with striking features such as zooming and multiple representation, or Fisheye Menus (http://www.cs.umd.edu/hcil/fisheyemenu/). See also their Licensed Products webpage at http://www.cs.umd.edu/hcil/pubs/products.shtml.
- The Visualization Toolkit software by Kitware Inc. can be freely downloaded from http://www.kitware.com/vtk.html.
- Visualization Software available at NCSA is linked from http://www.ncsa.uiuc.edu/SCD/Vis/.
- CAIDA Visualization tools are accessible at http://www.caida.org/tools/ .
- Geomview, an interactive 3-D viewing program for Unix, is available at http://www.geomview.org/.

Besides course web pages that link to one or two specific software packages, we could not find a site that links to a larger number of available IV software.

We are aware that the coverage of today's Web search engines is limited and that we may have missed valuable sites. We would appreciate any information on other software packages (especially written in Java) that are available for non-commercial purposes.

3. Creating a software repository

In Fall 2000, we started to develop a software repository that could be used by students to learn about IV by designing IVs. The repository is intended to complement the theoretical study of specific IV algorithms and the critique and evaluation of existing applications. It is supposed to facilitate the design and customization of IVs for different user groups, visualization tasks, and data sets. We wanted students to concentrate on interface and interactivity aspects of visualizations as opposed to the implementation of the algorithm itself. In addition, the code repository enables the comparison of different algorithms as well as the management of larger class projects.

After checking out available online resources, we selected different data mining and visualization packages. For each package, we created an informational web page that includes:

- A general description of the algorithm.
- A short characterization of the value of the algorithm for information visualization.
- Descriptions and links to (commercial) example applications of the algorithm.
- A description of the code including acknowledgments.
- A detailed explanation on how to unpack, compile, and run the package as well as hints for easy modifications.
- References.

Given that several packages were acquired for educational purposes, students had to sign an agreement sheet confirming that they would use the software for the IV course exclusively.

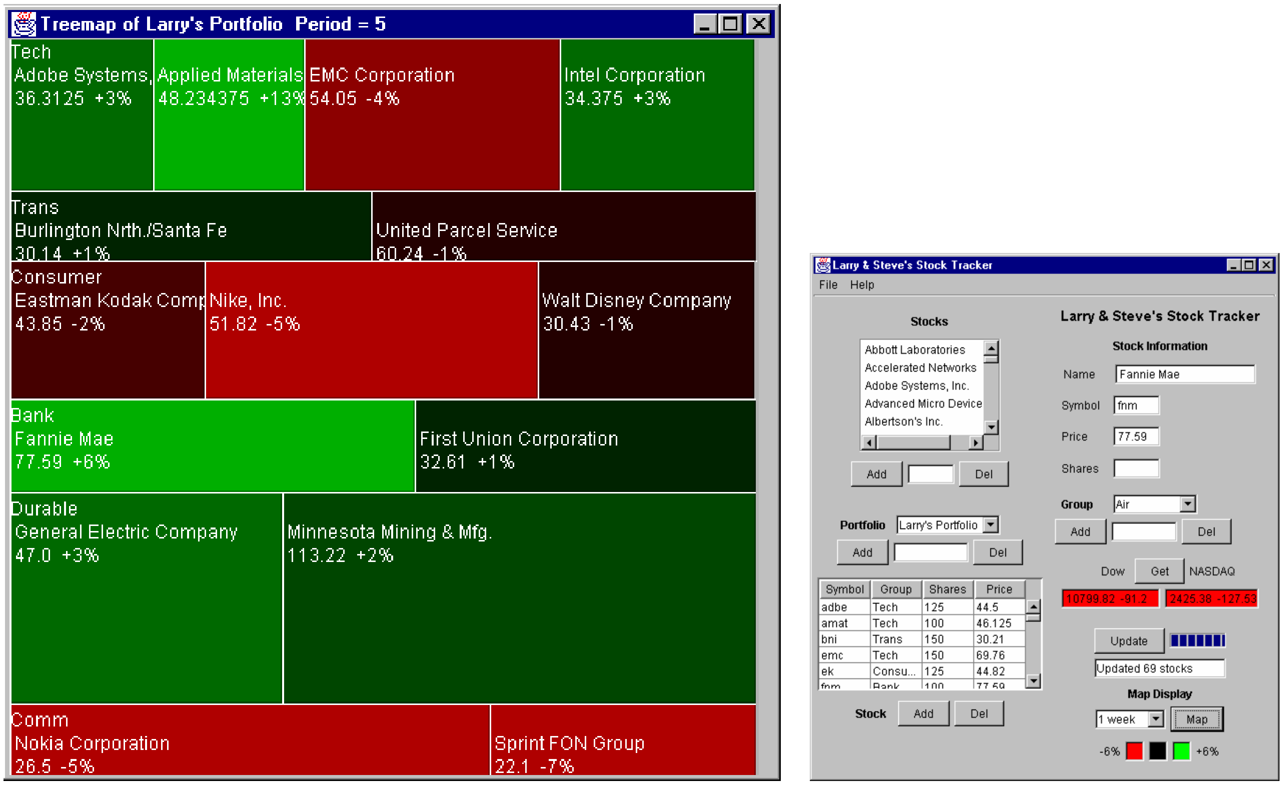
The main software repository web page is accessible at http://ella.slis.indiana.edu/~katy/L697/code/.

4. Using the software repository in class

Students taking the IV course at IUB attend lecture and lab sections. During lecture, students learn about visual perception, theoretical principles of IV design, plus a variety of existing data mining and visualization techniques, algorithms, and systems. They develop skills in critiquing and evaluating visualization techniques as applied to particular tasks and for specific user groups. During lab, students run, discuss, and evaluate different information visualizations using the software repository and gain a hands-on experience with those IV concepts that are inherently dynamic.

In addition, students constructively apply knowledge acquired in lecture and use the software repository to design novel IVs in three class projects. In the first programming project, students had to utilize *Treemaps* (Ahlberg et al., 1992) and *Hyperbolic Trees* (Munzner & Burchard, 1995) to visualize a data set of their choice for a

[1] Jazz has been released to the interested development community as OSI Certified Open Source Software to ensure that the widest possible community has access to and builds on the Jazz platform.



Treemap of Larry's Portfolio Period = 5

Tech
Adobe Systems, 36.3125 +3%
Applied Materials 48.234375 +13%
EMC Corporation 54.05 -4%
Intel Corporation 34.375 +3%

Trans
Burlington Nrth./Santa Fe 30.14 +1%
United Parcel Service 60.24 -1%

Consumer
Eastman Kodak Comp 43.85 -2%
Nike, Inc. 51.82 -5%
Walt Disney Company 30.43 -1%

Bank
Fannie Mae 77.59 +6%
First Union Corporation 32.61 +1%

Durable
General Electric Company 47.0 +3%
Minnesota Mining & Mfg. 113.22 +2%

Comm
Nokia Corporation 26.5 -5%
Sprint FON Group 22.1 -7%

Larry & Steve's Stock Tracker

Stocks: Abbott Laboratories, Accelerated Networks, Adobe Systems, Inc., Advanced Micro Device, Albertson's Inc.

Portfolio: Larry's Portfolio

Symbol	Group	Shares	Price
adbe	Tech	125	44.5
amat	Tech	100	46.125
bni	Trans	150	30.21
emc	Tech	150	69.76
ek	Consu...	125	44.82
fnm	Bank	100	77.59

Stock Information

Name: Fannie Mae
Symbol: fnm
Price: 77.59
Shares:
Group: Air

Dow 10799.82 -91.2
NASDAQ 2425.38 -127.53

Updated 69 stocks

Map Display: 1 week

Figure 1. The *Personalized Stock Tracker* interface by Larry Mongin & Steve Rice

self-selected user group. Within three weeks, students designed a *Personalized Stock Tracker* (Larry Mongin & Steve Rice), visualized the *Class inheritance hierachy of Java programs or APIs* (Ying Feng), designed visualizations of *File Directories* (David Heald; Alan Lin) and visualized the *Organizational Hierarchy of IU Employees* (Jason Baumgartner and Tim Waugh).Figure 1 shows the *Personalized Stock Tracker* interface implemented by Larry Mongin & Steve Rice (http://ella.slis.indiana.edu/~sdrice/L697/l697project2.html). It was inspired by the SmartMoney website at http://www.smartmoney.com/marketmap/ and uses the Treemap algorithm to visualize a dynamic list of stocks in a user portfolio. Users can create or modify a portfolio (which contains, for example, personal stocks) via an input panel, Fig. 1 (right). The NASDAQ and DOW index values for the selected stocks are retrieved either when the program is loaded or on demand. The stocks are visualized in a Treemap, Fig. 1 (left) in which each stock is represented by a rectangular area, labeled with the name and current index value of the stock. The size of the area corresponds to the amount of stocks the user holds and their value. A red-green color scheme is used to quickly convey the negative-positive performance of stocks over a selected time span. Existing portfolios as well as downloaded index values are stored upon the closing of the application. In sum, the *Personalized Stock Tracker* enables users to keep track of personal stock investments.

Ying Feng implemented a visualization of the *Class Inheritance Hierarchy of The Hyperbolic Tree* Java package (http://ella.slis.indiana.edu/~yingfeng/L697/proj2/doc.html) shown in Fig. 2.

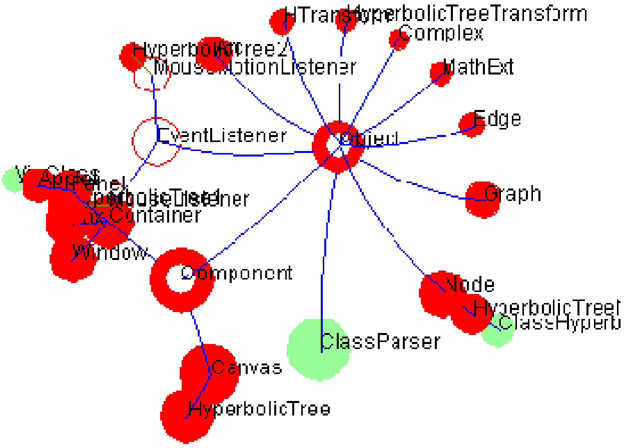
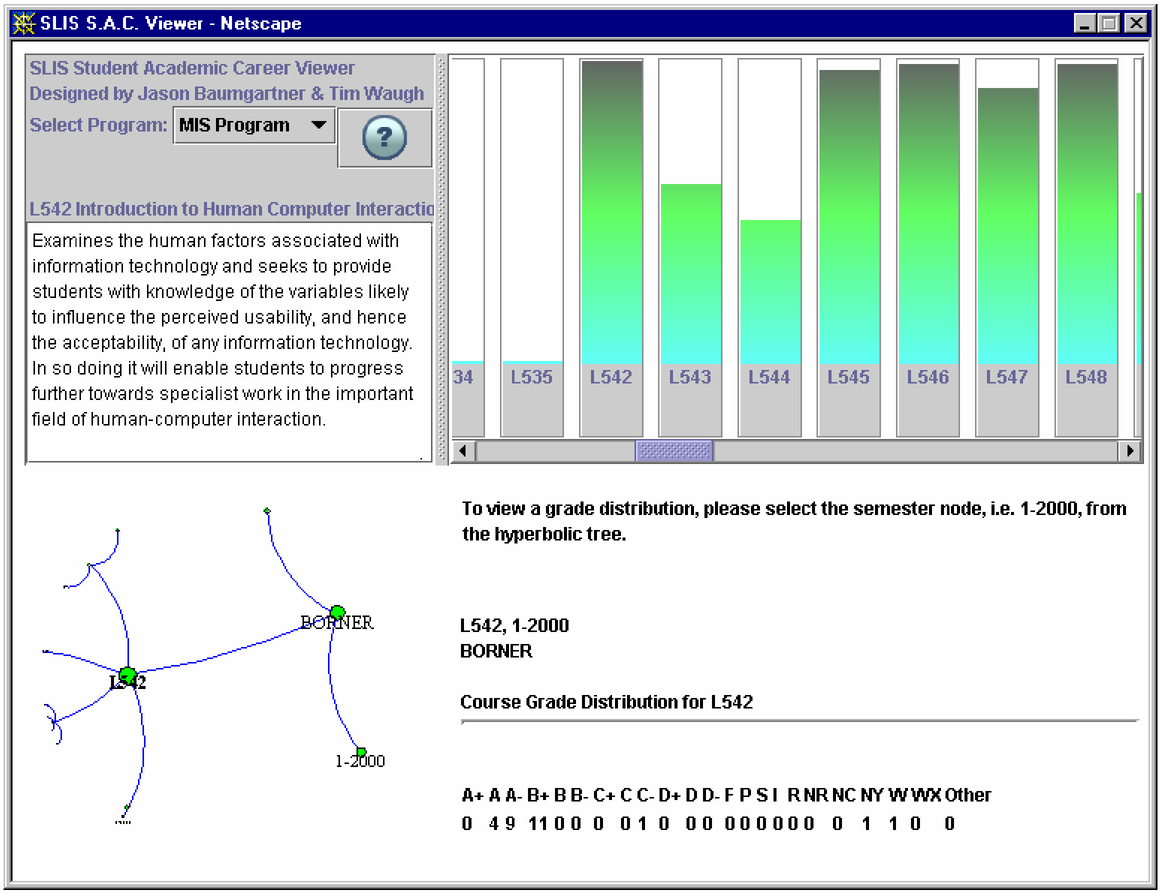


Figure 2. Visualizing the *Class Inheritance Hierarchy of The Hyperbolic Tree* by Ying Feng

It uses the hyperbolic tree algorithm to visualize its own code structure but can be used to visualize other packages as well. Different node shapes and colors are utilized to represent different kinds of Java classes. Different edge colors denote different class (inheritance) relationships see Fig. 2.



SLIS Student Academic Career Viewer
Designed by Jason Baumgartner & Tim Waugh
Select Program: MIS Program

L542 Introduction to Human Computer Interactio

Examines the human factors associated with information technology and seeks to provide students with knowledge of the variables likely to influence the perceived usability, and hence the acceptability, of any information technology. In so doing it will enable students to progress further towards specialist work in the important field of human-computer interaction.

To view a grade distribution, please select the semester node, i.e. 1-2000, from the hyperbolic tree.

L542, 1-2000
BORNER

Course Grade Distribution for L542

```
A+ A A- B+ B B- C+ C C- D+ D D- F P S I  R NR NC NY W WX Other
0   4 9  11 0 0  0   0 1  0   0 0  0 0 0 0 0 0   0   1   1  0    0
```

Figure 3. SLIS Student Academic Career Viewer by Jason Baumgartner & Tim Waugh

Project 3 concentrated on data mining. Students had four weeks to visualize a data set of files reporting what courses students took during their graduate study at IU to get a Master of Library Science (MLS), Master of Information Science (MIS), or Computer Science (CSCI) graduate degree in 1997 and 1998. The three files have the same format – each line contains a number assigned to a particular student, the student's class standing during the semester the class was taken, the semester code and year the course was taken, the section number for the particular course, the department where the course was offered, the four character course code, the course title, an indication of whether a student withdrew from the course or not, the number of credit hours awarded to the course, the degree the student received (MLS, MIS, CSCI), and the date the student received the degree.

software

Students were familiar with the SVDPACK by M. Berry (1996) for computing the singular value decomposition for Latent Semantic Analysis as well as with Multi-dimensional Scaling, Spring Embedding Algorithm (Battista et al., 1994), Treemap Algorithm, Hyperbolic Trees, Pathfinder Network Scaling (Schvaneveldt, 1990), and the ArcView software.

The students took very different approaches toward visualizing the data sets. Almost all applied Latent Semantic Analysis to extract the semantic relationships between different courses or different students. Some applied clustering techniques to further group semantically similar courses.

A number of students concentrated on the analysis of the data and the visualization of semantic relationships between courses or students using Multi-dimensional Scaling, Spring Embedding, or Pathfinder Network Scaling.

Jason Baumgartner & Tim Waugh implemented a *SLIS Student Academic Career Viewer* as shown in Fig. 3. The bars in the top right corner reflect the number of students who took a certain course. Clicking on a bar results in the display of a course description on the top-left and the visualization of different courses via a Hyberbolic tree browser in the left lower corner. Tree nodes corresponding to certain course sections can be selected and lead to the retrieval and display of the course grade distribution. The visualization is supposed to guide students in the selection of specific course sections. An online demo of the interface is linked from http://ella.slis.indiana.edu/~tawaugh/L697/project3/project3.html.

In their final project, students worked in collaboration with IU faculty on data mining and information visualization tasks related to the faculty member's research. They worked in teams on five projects: A Newsgroup Votes Visualization, a TransPac Network Traffic Visualization, an Hyperbolic Tree Visualization of Roget's Thesaurus, a Spring Embedder Visualization of Term Relationships for Concept Discovery, and a Visualization of Bookmark Files. The results of four of these projects have been or will be summarized and submitted to conferences and workshops.

Colored, full-size versions of figures 1, 2 and 3 are accessible at http://ella.slis.indiana.edu/~katy/IV2001b.

5. Discussion & outlook

Obviously, the construction of any software repository is an on-going process. Source code, descriptions, references, etc. need to be continuously revised and expanded to adapt to changes and new developments. Standards need to be established to enable diverse software packages (e.g., implementations of data mining & layout algorithms) to interact and share data seamlessly, to compare existing algorithms, and to incorporate new algorithms easily.

We are in the process of implementing an XML-based interchange format[2] for all Java software packages currently covered in the IV repository. The new factory and interface classes will allow all software packages to implement and to use a standard XML format and ensure that packages can be easily interchanged, compared, and combined. In addition, simple configurations of the XML input format should suffice to use the packages in a wide variety of applications. Our hope is that, in the foreseeable future, more source code will be released as Open Source Software. The XML interchange format will provide an easy way to incorporate this code. Last but not least, users will be able to use the XML standard to save interaction data such as manipulation changes in the data, the state of the visualization, etc. that could be advantageous to compare visualizations of different data sets among others.

In the short run, the software repository could lead to a WWW-based "textbook" with chapters covering specific IV topics, each an up-to-date survey of a sub-area of IV written by experts in this area and accompanied by source code. Each chapter could serve as:

- **Visual reminder** – showing snapshots of implemented interfaces that remind and attract attention to an algorithm.
- **Primer** – giving an introduction and explanation of the basics of the algorithm.
- **Textbook** – providing suggestions and hints on how to use, modify, and extend an algorithm and perhaps examples of lab exercises.
- **Encyclopedia** – listing links to articles and related web pages.
- **Meeting place** – inviting user to join virtual reading groups, chat rooms, FAQs.

In the long run – and analogous to repositories of other research communities – the IV software repository could exist as distributed content managed by editorial control. The editorial board would aim to establish and promote a common vocabulary and style and it would be in charge of attracting contributions and volunteer help in the spirit of the GNU project (http://www.gnu.org/). Ideally, the IV resource could be funded by selling access to institutions, by linking it to a comprehensive textbook, or by accepting advertisements from IV related companies.

Assuming that competition and privacy issues can be resolved, we believe that the proposed IV software repository would not only improve IV education but also boost creativity in IV research by easing access to existing work, consultation with others working on related topics, implementation of new (commercial) applications (which in turn challenge the development and improvement of the algorithms), exploration of new ideas, and, last but not least, the dissemination of results to science (Shneiderman, 2000).

6. Acknowledgements

We are grateful to the students taking the very first IV class at IUB in Spring 2001. They have been and still are a great motivation for the on-going design and expansion of the software repository. I would like to thank Jason Baumgartner and Maggie Swan for detailed comments on an earlier version of this paper. Ben Shneiderman

[2] We are utilizing the Sun JAXP API, which will be included in the JavaTM 2 Platform Standard Edition (J2SETM) 1.4 and is currently included in the JavaTM 2 Platform Enterprise Edition (J2EETM) 1.3.

generously contributed the GRIDL source code (Shneiderman et al., 2000).

7. References

Ahlberg, C., Williamson, C. and Shneiderman B. (1992) Dynamic queries for information exploration: An implementation and evaluation. In Proceedings of ACM SIGCHI'92, pages 619-626.

Battista, G. , Eades, P., Tamassia, R., and Tollis, I.G. (1994) Algorithms for drawing graphs: An annotated bibliography. Computational Geometry: Theory and Applications, 4 (5), pp. 235-282.

Berry, M. et al. SVDPACKC (Version 1.0) User's Guide, University of Tennessee Tech. Report CS-93-194, 1993 (Revised October 1996). See also http://www.netlib.org/svdpack/index.html.

Bederson, B. B., Hollan, J.D., Perlin, K., Meyer, J., Bacon, D., and Furnas, G. (1996). Pad++: A zoomable graphical sketchpad for exploring alternate interface physics. Journal of Visual Languages and Computing, March 1996, vol.7, (no.1):3-31.

Card, S. K., MacKinlay, J. D., Shneiderman B. (Editors) (1999) Readings in Information Visualization: Using Vision to Think, Morgan Kaufmann Publishers.

Chen, C. (1999b) Information Visualisation and Virtual Environments. Springer Verlag.

Dodge, M. & Kitchin, R. (2000) Mapping Cyberspace. Routledge.

Munzner, T. and Burchard, P. (1995) Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space. Proceedings of VRML '95 (San Diego, California, December 14-15, 1995), special issue of Computer Graphics, pp 33-38, ACM SIGGRAPH, New York.

Shneiderman, B., Feldman, D. and Rose, A. & Grau, X. F. (2000) Visualizing digital library search results with categorical and hierarchical axes. Proceedings of the fifth ACM Conference on Digital Libraries, San Antonio, TX, pp. 57-66.

Shneiderman, B. (2000) Creating creativity: User interfaces for supporting innovation. ACM Transactions on Computer-Human Interaction, Vol 7, Issue 1, pp. 114-138.

Spence, R. (2000) Information Visualization. Addison-Wesley.

Schvaneveldt, R. (Editor) (1990) Pathfinder Associative Networks: Studies in Knowledge Organization. Norwood, NJ: Ablex.

Ware, C. (1999) Information Visualization: Perception for Design, Morgan Kaufmann Publishers.