Chapter IX
Visualization in Support of Social Networking on the Web

J. Leng
Visual Conclusions, UK

Wes Sharrock
University of Manchester, UK

ABSTRACT

In this chapter the authors explore the contribution visualization can make to the new interfaces of the Semantic Web in terms of the quality of presentation of content. In doing this they discuss some of the underlying technologies enabling the Web and the social forces that are driving the further development of user-manipulable interfaces.

INTRODUCTION

The internet is a communication device. Its interface consists not just of static text but other static and dynamic constructs e.g., tables, images, animations and customised web applications. Some of these elements hold meaningful content while others are used for graphic reasons. The rise of social networking in the form of Weblogs, discussion boards, wikis, and networking sites allows the general public to share content on the web. Such non-technical users require high level web-apps to help design and deliver their content with as little explicit dependence on the technicalities as possible.

Before talking about social networking on the web it is worth considering what this means. The expression ‘Social Networking Sites’ is used for sites with the primary purpose of supporting or creating sociable relationships, prominently friendships, but we use ‘social networking’ in a more inclusive way to include the formation of all kinds of networks such as those, for example, that form to collaborate on a task (as with an open source development).

The naming conventions used within visualization can confuse, this relates to the difficulty in producing a visualization taxonomy which is discussed later. Visualization is a visual means to analyse data and is cross-disciplinary. The name
of the discipline normally identifies a particular theme but where the name of the discipline is also the name of a technique the naming convention lacks clarity. For example social networking visualization could be the use of visual methods to show and analyse social networks or could be the use of visualizations in the support of social networking i.e., as a means to help people form and co-ordinate their activities. This chapter looks at the latter, at how visualization can aid communication on the web. Whilst this can include the visualization of social networks (because users of the web may like to understand the social networks they participate in) that is not the primary focus.

In this chapter we introduce visualization, its history and the two main visual paradigms in use, dividing the visualization community between those concentrating on scientific and information visualizations respectively. We survey the technologies that shape the web and the applications running on it. This allows us to look at how the technology shapes visualization systems (the visualization pipeline and the flow of data) and how these can be distributed to work efficiently in web environments. Finally we review some web applications that support social networking and consider what future trends may be.

WHAT IS VISUALIZATION?

The History of Visualization

There is no accepted definition of visualization but it can be adequately summarized for our purposes as using visual means to aid the communication and understanding of information. Modern visualization increasingly uses computer graphics technology to make information accessible. Visualization’s long history predates the origin of computers by at least 8 thousand years. Maps are one of the oldest forms of graphical aid whose continued usefulness is demonstrated by the fact that mapping applications are amongst the most popular web-based applications. Before computers visualizations generally were not interactive, though there are exceptions to this as some scientists developed models and pop ups in books to explain their ideas but these were rare and expensive (Tufte, 1997; Tufte, 2001).

The roots of visualization are tangled into our history; a timeline of visualization is available on the internet (Friendly, 2008). Many historical breakthroughs were made possible through visualization, such as John Snow’s use (in London in 1854), of maps to show the distribution of deaths from cholera in relation to the location of public water pumps. Visualization has never been an isolated discipline; it has been an integral element of scientific, intellectual and technical developments.

The timeline of visualization shows that the development of visualization has accelerated since 1975, since when important changes have depended upon advances in computing. Improved computer speed and capacity increasingly allow data to be visualized by increasingly intensive computational methods. Computers make visualizations more interactive and allow direct manipulation of data, e.g. selecting data by linking, brushing or using animation in grand tours. Also driving the development of visualization is the fact that visualization methods are being applied to and developed for an ever-expanding array of problem areas and data structures, including web applications that enable social networking.

Modern Visualization

Modern computer-based visualization developed through the accumulation of three specific areas (Schroeder, 1997). Scientific visualization was the first. It is a discipline stemming from computational science and started as an IT support activity. Computational simulations produced digital data representing natural phenomena, for example the weather forecast. Commonly the data
has an inherent geometrical shape and the data represents continuous fields within this geometry. The data were produced as large arrays of values that were difficult to analyse by hand or eye. Instead, computers were used to produce images of the data within its native geometrical shape.

‘Scientific visualization’ is typically categorised by the dimensionality of the data values (number of dependant variables), and whether the data is scalar, vector, tensor, or multivariate (Brodie, 1992; Schroeder, 1997). However another taxonomy distinguishes by technique between global, geometric and feature-based techniques (Post, 1999).

Subsequently, it was realised that techniques for scientific visualization could be applied to data from the humanities and social sciences. Since this data commonly looks at geographical distributions of populations this field focuses on statistical graphs and thematic cartography. Gradually, the field of data visualization diverged further and information visualization, the final area of visualization, emerged. This field was primarily aimed at visualizing computerised databases so that relationships could be found; commonly using tables, graphs and maps. Currently visual display and analysis of text in the form of academic papers, web pages, and patient records, are included in information visualization. ‘Information visualization’ can also be classified by data type, common types being multi-dimensional databases (with more than 3 dimensions), text, graphs and trees (Bohm, 2001; Schneiderman, 1996). Similar to scientific visualization, a technique classification system is based upon display styles which include table, or information landscape (Card, 1997; Chi, 2000). However unlike scientific visualization some classifications also look at tasks, for example gaining an overview or drilling down on detail (Schneiderman, 1996).

Over this period visualization has been transforming into an increasingly independent academic subject within the field of computer graphics and is commonly taught in computer science departments, and is divided between two main specialisms categorised as ‘scientific visualization’ (developing visualizations for purposes of aiding scientific research) and ‘information visualization’ (providing ways of communicating information generally e.g. in the design and presentation of charts, diagrams, graphs, tables, guides, instructions, directories and maps which can also be used to aid research whether scientific or not). However this division does not best reflect visualizations true nature, and there are attempts to improve on this classification. Tory (2004), complains that the division between scientific and information visualization encourages segregation of different kinds of activities from one another when many visualization problems cross that divide, proposing instead, a systematic scheme ‘based on characteristics of models of the data rather than on characteristics of the data itself’. An easy-to-understand taxonomy would be useful to both users of visualization and researchers into visualization and if it existed it would have been presented here.

The application area of interest in this chapter falls mostly within the information visualization paradigm although map based visualization applications are also important but these techniques are on the border of the 2 paradigms drawing from both scientific and information visualization (Tory, 2004). However visual appearance is not the only theme in visualization research of importance to web 2.0. Visualization system design and the distribution of that system are relevant research themes. The web is a distributed computing environment, with users also distributed and if they are to work together then synchronization of their activities must be facilitated by the visualization system. These themes were originally titled ‘remote visualization’ and ‘collaborative visualization’, growing out of scientific visualization. To understand the underlying technology and social forces driving the development of the web it is necessary to understand the position of and the relationship between the information visualization and scientific visualization communities.
A Survey of Information Visualization Techniques

There are hundreds of techniques with thousands of low level elements that could be considered. This chapter is too short to take on this task so we aim to give the reader an idea of where to look for further information and an overview of a few relevant techniques. Many information visualization techniques rely on the use of colour, animation and interactive input from the user but in this book only static grayscale images are possible so where possible we refer the reader to websites for examples.

Visualization is a young subject without a definitive text but the selection of papers, books and websites in the references section should provide a suitable starting point. Friendly’s timeline of visualization (Friendly, 2008) presents an online timeline of visualization with static images for each entry in the timeline. This site is a good starting point for examples of visualization techniques developed before 2004. Images and demos of information visualization techniques are on the websites of commercial vendors of general purpose information visualization software (although demos are not always available and users may require registration). There are 5 stable information visualization companies known to the authors: AVS (Advanced Visual Systems) has OpenVis (AVS, 2008); Spotfire focuses on business information (Spotfire, 2008); Tableau is similar to Spotfire (Tableau 2008); Steema has TeeChart and TeeTree (Steema, 2008); and finally aiSee is designed for large data (aiSee 2008).

Some key techniques from information visualization are:

- **Graphs**: a 2D plot showing the relationship between parameters. Typically orthogonal axes show the values of the parameters represented in the plot. Harris (1999) illustrates all the visual elements of graphs, tables and other information plots and recent research is available in the Journal of Graph Algorithms and Applications (Tamassia & Tollis, 2008).
- **Trees and Networks**: nodes are connected by lines showing relationships between nodes. If the nodes represent people then the plot represents a social network. These plots can be difficult to understand if many nodes are used. Networks do not occupy a real geography but are optimized by complex mathematics to look good in a 2D display. In information visualization these are considered to be graphs. Freeman (2000) and Bertini (2008) survey the history and applications in this area, and two case studies of social network visualization applications are in Brandes (2001) and Shen (2008).
- **Multiple Related Views**: a number of 2D plots are viewed in one problem solving environment. If the plots are graphs then the plots tend to be arranged so that the horizontal or vertical axis that relate to the same parameter are aligned and a user selection made in one graph can be seen to relate across to the other plots. The next 4 techniques use multiple related views but relationships are made by different user interactions.
- **Drilling**: if the plots are geographical then they may represent different levels of detail of a region the user selects from the higher resolution plot.
- **Brushing**: if a user is interested in the geographic distribution of different age groups within the US then using the histogram and selecting (brushing) regions from the histogram this selection is displayed on a map.
- **Linking**: many elements are related so colour is used across multiple views to clearly identify each relationship.
- **Grand tours**: an animated tour is given through the data display space so that the user can get an overview of the data.
The Development of Visualization in Relation to Computer Technologies

The development of visualization is driven not only by the development of visualization techniques, but in response to the development of new computer technologies offering new possibilities for visual representation. Changes in computer hardware, graphics hardware, computer display devices, computer input devices, software design, collaborative working environments, remote visualization and visualization services all play an influential part in determining what can be visualized and how the user can manipulate it.

IN TRANSITION FROM WEB PAGES TO WEB APPLICATIONS

The Handbook is about the changes from Web 1.0 to Web 2.0, made possible by a mix of new technologies and approaches. The step changes in technology and user participation are not yet completed. We review the current state of the technology and how it might adapt to provide better graphical tools for the ‘semantic web’.

From Static to Dynamic Web Pages

One of the main changes is the development of Content Management Systems (CMS) providing a web page delivery system that replaces static HTML web pages (North 2008). CMS separates two aspects of a web page, the content from the presentation. In CMS the content is separated from the web page as it no longer sits in an HTML file waiting for a user to access it, but is in a database so that the HTML page is dynamically reconstructed for the user when they select that web page.

The change from static to dynamic content has relied on these changes in technology:

1. Static web pages: the content and presentation are not only included in the same file but are mixed together throughout the text in the file
2. Cascading Style Sheet (CSS) web pages: the content and presentation are separated into separate text files. HTML (Hyper Text Markup Language) is the language encoding the information on a web page. It provides the information as text along with text that defines the style (presentation) of the informational text. The presentation information can be separated from the content information by the use of Cascading Style Sheets. Such CSS enable changes to the look and feel of a whole website through alterations to only one file, rather than requiring the individual rewriting of every page on the site.
3. Dynamic web pages: content and web pages are separated.

CMS are powerful because they separate out the responsibility for designing and developing a website from providing the content. CMS build in tools to enable non-technical users to enter and manage their content (which is why CMS are popular for blogging sites). Many social networking websites give the user the option whether to produce their content as text or HTML, which contributes to the control that users now want over web content (manifested by the popularity of open source communities), giving them control over its style. There are in effect two interfaces to a website, one between the site and its non-technical users, the other between the site and those with technical competences. Site providers increasingly encourage technically competent users to create and share service specific applications by releasing their API (Application Program Interface) and disseminating technical information which makes the addition of new modules easy.
HTML remains the underlying language that is recognized by web browsers so CMS must reconstruct HTML pages dynamically for the viewer. Even where web delivery technologies, e.g., wikis, have their own markup language or where websites allow plain text entry these inputs are still translated into HTML. Standardising interpretation of HTML across all browsers has taken considerable time and effort, improvements to HTML being coordinated between browser vendors by the World Wide Web Consortium (W3C). It would be too problematic to get the W3C to coordinate this effort and for browser developers to commit the resources for each brand of markup language, so wikis and comparable applications have developed their own markup with special functionality for their relevant formatting and semantic issues. Nonetheless, wiki applications have their own translator to convert that markup into more basic HTML components. HTML has limited functionality with regard to style, so the functionality that handles the graphical content and interactivity important to visualization is provided by supplementing HTML with other languages. These languages commonly provide applications (e.g., web-based games) or structural and navigational tools (e.g., menu systems) for a website. Depending on their specific functionality these languages can create applications that are stored in and served from a database in a CMS site, adding to the CMS functionality or providing style for a static web page. The main languages that contribute to visualization on the web were developed in the 1990's, including Java, JavaScript, VRML (Virtual Reality Modeling Language) and Flash.

The Java programming language was developed by Sun Microsystems (Gosling, 2005; Java Home Page, 2008; Wikipedia, 2008). It is an interpreted language meaning that a program written in Java can run on all the popular computer platforms without any adaption; “Write Once, Run Anywhere”. Combined with its other features this makes it ideal for the web so web browsers quickly incorporated small Java programs (called applets) within web pages. Later Java was configured for particular platforms e.g. J2ME (Java 2 Micro Edition) for mobiles. Java Servlets, instead of being embedded in a web page as applets, are used to extend the functionality of the web server, allowing extra content to be added to web pages dynamically from the server. Whilst Applets and Servlets can be used to add important visual content to the web, Java is much more powerful for such a purpose as it has dedicated libraries that handle necessary functionality: the SWING library adds user interface functionality, the Drag and Drop library allows object manipulation by mouse, and the 2D and 3D libraries allow the graphics modeling of 2D and 3D objects.

JavaScript is a scripting language commonly used for client-side web development; the web browsers incorporate the ability to interpret JavaScript programs (W3 Schools, 2008; Wikipedia, 2008). JavaScript was designed to look like Java but has less functionality and is easier for non-programmers to work with. The primary purpose of JavaScript is to embed interactive functionality into web pages, typically to inspect or create content dynamically for that page. For example, a JavaScript may validate input values in a web form, control the opening of pop-up windows or change an image as the mouse passes over it. Because JavaScript code runs locally in a user’s browser (rather than on a remote server), it can respond to user actions quickly, making an application feel more responsive. Furthermore, JavaScript code detects user actions which HTML cannot. JavaScript is heavily used in many web-based applications including CMS and those that support social networking through gmail and facebook.

VRML is a standard text file format similar to HTML used to represent 3D interactive objects (W3D Consortium, 2008; Wikipedia, 2008). Developed with the web in mind a browser can interpret VRML by installing the appropriate plug-ins. A number of small geometrical primi-
tives are defined within the format and each of these primitives may have a number of properties that define the visual aspects of the object such as colour or transparency. The shape of large 3D objects are defined by combining the correct geometrical primitives e.g., 3D surfaces are defined as a mesh of triangular primitives. These models are interactive in a number of ways. Web links can be made by clicking with a mouse on a node, timers and external events can trigger changes in the scene and Java or JavaScript can be incorporated into the world (VRML files are called worlds, the term world is used where in other graphical systems the term scene would be used). The VRML format is an open format that has an ISO standard making it suitable for sharing geometrical model data which ensures its popular within academia. Successors to VRML include X3D and 3DMLW (based on XML).

VRML is useful in applications where the 3D shape of an object is important such as teaching anatomical structures to medical students. An early interactive web-based application using VRML was created to teach medical students to perform lumbar punctures (John, 2001). A model of the external skin, spinal bones, spinal cord and cerebrospinal fluid (CSF) were combined in one world, enabling the student to manipulate the model in the viewer, viewing it from any angle and altering the transparency of all the elements. The student could then place the puncture needle in the model to simulate taking a sample of CSF. The student would be given feedback on their performance and could alter the transparency to see into the puncture site.

Flash (currently Adobe Flash, previously Shockwave Flash and Macromedia Flash) is a multimedia graphics program used to create interactive animations for web pages (W3 Schools, 2008; Wikipedia, 2008). Its features make it especially suited for a web environment. It uses vector graphics, which means that the graphics can be scaled to any size of display area without losing quality and it supports bi-directional streaming so that it can load into a web page more quickly than animated images. Most web browsers can interpret Flash but the Shockwave Player can be downloaded for free and used as a plug-in. Flash is being ported for use on mobiles and PDA. Flash applications are developed through an authoring tool.

From HTML to XML

HTML provides static content for the web and deals with formatting and style of text. While HTML allows the publisher to present their information in a particular style, Web 2.0 uses XML (Extensible Markup Language) to handle the data that is passed across the internet. XML has a user defined format to handle particular types of data which means that without standardization of XML file formats the variation in XML file content makes them difficult to handle. File formats based on XML and handling data relevant to visualization are developing e.g., the previously mentioned 3DMLW format. While that data does not have to be human-legible both HTML and XML are designed to be human-legible. Currently XML is rendered as raw text in a web browser with no unified display protocol for XML across all web browsers. In order to style the rendering of XML data the XML file must reference a style sheet that can not only give style but can also convert regions of the data into other data formats such as HTML.

Eventually XML may replace HTML and if it does good styling and visual display functionality may be included in the format. However currently the importance of XML is not in how it is rendered but as a way of formatting data to allow machine readable semantic information to be incorporated into the web. The best known function of XML is in price comparison websites, but it could be used to extend content to give information of interest (on when, what and the provenance of information) to social networking sites. Comparisons of large amounts of information where the structure,
layout and interaction are important would benefit from visualization techniques.

VISUALIZATION SYSTEMS AND THE DEVELOPMENT OF TECHNIQUES

The Visualization Pipeline

The ‘visualization pipeline’ (Figure 1, from Haber and McNabb (1990a) gives an abstract presentation of the visualization computational process), indicating how scientific simulations are rendered in computerized form and thus made available to computational scientists. While this abstraction was developed for scientific visualization it can be expanded to encompass visualization more generally, and, some argue, covers computation generally. It shows (Figure 2) which parts of the pipeline are dependent on which computer hardware component. The performance of a pipeline is determined by the component with the worst performance called a “bottleneck”. Developments in areas of computer technology on which the pipeline depends alter what is possible in the corresponding part of the visualization process. The graphics hardware and the display device are closely tied because the graphics card produces the images seen on the screen and this stream of images is a heavy load. The short dedicated cable connecting graphics processor and screen in a desktop computer is sufficient to enable the graphics card to service an acceptable refreshment of on-screen images—requiring at least 25 frames per second—but this works less well if the cable is replaced by a shared network for example the internet. While software on the pc comes with advice on minimum hardware requirements web-based applications do not, so users with low performance hardware will either have a poor experience of the application or the application must adapt to the user’s system. The development of adaptive systems (Brodie, Brooke, Chen, Chisnall, Hughes, John, Jones, Riding, Roard, Turner, & Wood, 2007) allows the detection of and adjustment for the nature of display systems and network speeds, allowing tailoring of the output so that, for example, a small mobile device could be fed lower resolution images to those fed into a wired pc or home entertainment system.

Physical interaction with a visualization is by two different types of input (Figure 3). Generally speaking interaction with the graphical object held on the graphics hardware is fast and uses a mouse or other device that feels natural. Interaction with the computation that occurs in the pipeline before the data reaches the graphics hardware tends to use application specific menu systems with mouse and keyboards input separated from the visual scene and disruptive of activity as the user must shift concentration from the graphical objects they are working with. Virtual Reality menu systems (Curington, 2001) appear to lessen disruption to the actions of the user but these can cause clutter in the visual scene and to reduce that clutter these menu systems have limited functionality. A third and dominant, but not physical, interaction is between the functionality of the application and the requirements of the user: if the visualizations do not appropriately represent reality then the user dependent on them may be

Figure 1. An overview for the whole of the visualization pipeline originally given by Haber (1990a) in text and Domik (2008) diagrammatically and refers to computational science. “Computer representation of reality” referred to computer simulation but could refer to a database or any digital data

```
| Reality | Computer representation of reality (data) | Image(s) | Viewer(s) |
```

189
Figure 2. Overview of the whole visualization pipeline in figure 1, but in this case the hardware that each element of the pipeline depends on is identified. Computer and graphics hardware can be part of the same machine but this is not necessarily so, but graphics hardware is normally closely tied to the display device.

Figure 3. Overview of the whole visualization pipeline in figure 1, but with the 3 main types of user interaction given and numbered to match the order of discussion in the body of the chapter.

Figure 4. Overview of the visualization pipeline for just the visualization system showing how internally the visualization system handles data-flow (from left to right) and categorises types of interaction by data transformation type.
Visualization in Support of Social Networking on the Web

misled when making their decisions, and if they are not confident that the results are useful they will abandon the system.

The visualization pipeline of the visualization system alone was further abstracted to categorise the interaction with the user and so exhibit the data-flow paradigm that is inherent in the software structure of visualization systems (Figure 4). The data is shown as flowing through the system while transformations are performed to deliver the final image. Haber (1990a) identified three types of transformation. The viewer is able to interact with the visualization and so affect each of these transformations, potentially gaining new insight into the data being displayed.

First, raw data is transformed by a data enrichment step into 'derived data'. Raw data can come from any digital source such as simulations, data files and databases. Enrichment fits the data to the desired model perhaps including data manipulations such as smoothing, filtering, interpolating, or even hypothetical alterations. Preparing data for visualization can be time consuming and so deter the first time users.

The second transformation, mapping, converts data into an abstract visual object (AVO). Attributes of the data are converted to corresponding geometric features according to a set of classification rules. By adjusting these rules, the viewer can cause distinctions in the data to become perceptually obvious. This step establishes the representation of the data, but not necessarily the final visual appearance on the screen.

The third transformation, ‘rendering’, ultimately converts the AVO to a displayable image. This involves computer graphics techniques such as view transformations (rotation, translation, scaling), hidden surface removal, shading, shadowing. The transformations at this stage occur on the graphics hardware and alter the perspective on the display, contribute to the overall appearance and dictate what is or is not visible.

This pipeline makes explicit the transformations the data must go through to produce images. These transformations must be performed in this sequence. The concept of the data passing successively through each transformation is built into visualization software as ‘data-flow’. This pipeline is drawn to reflect generic features of data transformation but specific ones can be drawn for particular applications and used to optimize and/or split the process. This is a common strategy for complex visualization applications such as Google Earth which branch the pipeline and spread the computation across a number of computational resources in a way invisible to the user but designed to improve the functionality and interactivity for the user.

Interactive Rates

Interactive rates strongly constrain visualization strategies. There is no exact criterion for an appropriate interactive rate. In TV and film production there are three different frame rate standards, the slowest at 25 frames per second. Stereographic systems need at least twice this rate as one frame is produced for each eye. On a single unloaded machine a visualization system will maintain a good frame rate up to a threshold in data size where the threshold is determined by the data’s geometric composition, the complexity of the visualization techniques used, and the memory usage rather than by the disc space that the raw data occupies. Higher frame rates improve user satisfaction particularly for stereoscopic projection but in exceptional circumstances a rate may be acceptable down to about 10 frames per second, as when the user has no other way of viewing their data, however below this rate the user will prefer static images or animations produced in batch mode.

These issues affect the interactive rate of transformations occurring in the graphics hardware but changes to parameters in the earlier transformations (enrichment and mapping) alter the representation of the data output from a particular visualization pipeline without affecting the frame
rate, so the rate that enrichment and mapping are updated is slower. There is no exact figure, but a lag of more than a few seconds causes delays that may confuse the user, who cannot be sure what transformations have already been applied and which are waiting to be fed into the scene currently being viewed.

**Visualization Systems**

Scientific visualization software was initially developed before the idea of web services had taken hold. It mainly consisted in standalone software taking advantage of specialist graphics hardware and specialist display and interaction devices. This close tying of software to hardware improves performance and made this a suitable setting for the development of virtual environments. Scientific visualization started as a support activity for computational science when simulations were run on supercomputers that were administered in computing centers. The need for the visualization of simulations drove the development of computer graphics hardware and virtual reality environments by Silicon Graphics, a supercomputing vendor. Scientific visualization systems exploited local hardware using many processors and optimization strategies to allow the largest possible datasets to be visualized. Initially the display was to a simple screen but larger display areas were developed using projectors or tiled screens so that a number of users could explore and interact with the visualization scene at the same time. New input devices were needed in these environments such as the space mouse (a mouse designed for manipulating objects in a 3D environment by permitting the simultaneous control of all six degrees of freedom) as the users would walk close to the screen and away from the keyboard and mouse which were then wired into the computer terminal. The 3D scenes produced by the visualization software seemed flat; meaning that information such as the depth of an object was difficult to interpret so stereoscopic output was integrated into these environments to trick the eye into seeing these objects in their “real” 3D form.

When the visualization systems were physically tied to the hardware, the user needed to be in the computing center to take advantage of them. Thus users did not make intensive use of this specialist and very expensive equipment so the idea of remote visualization developed to allow users to stay in their office and make use of the graphics hardware over a network. The first remote visualization systems were developed before the web and were developed alongside the abstraction of the visualization pipeline (Haber, 1990b). Pioneering visualization web services used scientific visualization software on a supercomputer to produce static 3D images from medical scans served interactively through a web browser to medics (Jackson, 2000). Later more dynamic visualizations dependent on specialist graphics hardware again physically located in the computing center using a product, Silicon Graphics VizServer were developed. However their use was restricted in practice to projects where the visualizations were produced on a proprietary Silicon Graphics supercomputer and a dedicated high bandwidth network was in place; the images produced on the graphics hardware were compressed before passing down a dedicated network to users who could interact with the objects in the images. The Op3D project delivered and projected 3D medical visualizations onto the wall in operating theatres so surgeons could compare the patient to the 3D visualizations side by side (McCloy, 2003). Typically specialized research systems produced their software from scratch, in this case developing a user interface employing a joy stick that was easy to use and suitable for a sterile environment. Collaborative visualization allowed users to share their experience either by sharing the results of the visualization on a large display or by splitting the visualization pipeline at some point and distributing the output across the network where it is rendered and displayed for each collaborator.
Visualization in Support of Social Networking on the Web

Collaborative environments used for teaching, consultations and group discussions enhance how knowledge is learnt and shared. The idea of adaptive visualization has been furthered by the Grid research initiative which aims to provide computer resources and services to academics across the web (Brodlie, et al. 2007). It is not a visualization system but it is a remote service that supplies users with visualization resources intelligently. In principal the user cannot choose which visualization software they use on which machine but instead define the scene that they wish to interact with and their local resources. As all visualization systems produce visually different results even for the same algorithm the user could find results from different softwares were not comparable. Visualization is a diverse and cross-disciplinary field so that the terminology within visualization and its applications is not consistent and this impacts on the ability to create an ontology or taxonomy for visualization making it difficult creating a visualization modeling language standardised across all applications (Brodlie & Mohd Noor, 2007). Technologies developed by the games industry are now the main driver for collaborative synchronous graphics environments.

Information visualization software followed a different path, developing later than scientific visualization software. Some early visualization systems for viewing social networks were adapted from scientific visualization software for molecular visualization using stereographic displays and VRML (Freeman 2000). However dedicated information visualization software was coded in Java rather than C or C++ and was suitable to integrate with web services from the start. Information visualization like scientific visualization displays large quantities of data often using different display techniques requiring less performance from the graphics hardware. Characteristically information visualization applications are more meaningfully displayed in 2D rather than 3D and use multiple related views that are easy to develop as web services. However such applications are computing intensive in the enhancement and mapping transformations rather than the rendering transformation (meaning that demands on the graphics card are contained). ManyEyes, a research application from IBM (Viegas 2007) is a web-based system that allows users to upload and visualize data through their browser. The visualizations are produced by java applets running asynchronously on the user’s computer allowing each user to visualize and comment on uploaded datasets in a social networking environment similar to video and photo sharing in flickr or YouTube. Designed for visualization novices the system uses “ShowMe”, a special interface to help users select suitable visualization techniques for their data, eliminating the need for a visualization modeling language. Its developers advocate good design web-based because it increases ease-of-use, as the authors of this chapter agree. Java applets are suitable for techniques that are not compute intensive.

Independently of the visualization community the Access Grid, an internet based video conferencing system, has developed. It is superior to video conferencing because the equipment is cheaper, using ‘multicasting’ technology that allows many sites to participate, facilitating online meetings, seminars and conferences. Access grid works by streaming video and audio over the internet, it is easy to stream visualization scenes across the internet and so include it in a session. This is interactive at the host site but not yet at other sites. Access grid sessions can be recorded and played back using visualization techniques designed to augment the analysis and playback of meetings (Buckingham Shum 2006; Slack 2006).

In Google Earth, a visualization web service combining satellite images, maps and other GIS information, vast amounts of computation are needed to serve the data to the user so that the best results are obtained for each user no matter what their graphics hardware (Jones, 2007). Web
Visualization in Support of Social Networking on the Web

services will always be slower than locally run software running on high performance hardware. The distribution of computational resources needs to be balanced between the server and client, which are more noticeable for visualization than for non-graphical and non-interactive web applications. Currently, it is only by providing high performance data serving as in Google Earth that this complex visualization application becomes possible. There are other visualization applications that may be useful to social networking for example the production of large graph trees for particular members of a social networking website, perhaps including a famous politician who has an account on a social networking website. Large graph trees require high computational intensity and a low graphical intensity so the performance of an application aiming to produce such a tree would also require specialist hardware and code optimization, most likely to come from the proprietors of a social networking website rather than the open source community that is extending the functionality of the website. More generally such complex applications are most likely be developed within the commercial sector.

Visualization Design

Understanding visualization techniques is only part of the story. Knowing which techniques to pick, how to combine low level design elements and how to appraise a visualization are also important. Many elements of design are open to debate. Here we give the reader an idea of possible approaches to design and analysis.

The most scientific approach to design comes from psychology. Colin Ware, perceptual psychologist who for a short time worked as an artist before becoming interested in visualization, has studied visualization design from the perspective of the science of perception. His book (Ware, 1999) is the definitive guide to issues such as when to use words and/or images and the possibilities for visual programming languages. Lately techniques from psychology are used to test how “good” a newly developed visualization technique is by a method called ‘user assessment’.

Another way to understand visualization design and analysis is through the history of how information has been visually represented. If a number of visual information representations can be understood from throughout history then it may be possible to use that understanding to analyse current visualizations and even to predict future developments (Friendly, 2008; Tufte, 1997; Tufte, 2001; Tufte, 2006; Wainer, 2005). By studying history it is possible to understand better the social context in which the techniques were developed and explain two types of negative outcomes. Firstly, the situation where techniques were developed but at that time it was disregarded only sometime later to become an accepted visual representation. This was the case with William Playfair who in the 16th Century developed the grounding for modern statistical plots 150 years before they were accepted (Playfair, 2005). Secondly, where the poor visual representation of information have contributed to the failure of projects. This approach has been applied to two NASA based projects involving the launch of the space shuttle Challenger and the space flight of the shuttle Columbia in 2003 (Tufte, 1997; Tufte, 2006). To improve his analyses Tufte shares information with a variety of experts who use visual methods to communicate information, creating a moderated web forum for this purpose.

Cross-disciplinary teams can not only be good for analyzing visualizations but also for designing and creating visual representations. The term “renaissance team” was coined in the mid 1980s by an artist, Donna Cox, who worked collaboratively with computer scientists and scientists to find novel visual ways to explore how the universe was formed. Collaborative cross-disciplinary teams including artists trained in design can be used to create novel and visually pleasing techniques. Art criticism is the discussion or evaluation of visual fine arts, and it has been suggested that it
Visualization in Support of Social Networking on the Web

be extended for the analyses of the products of visualization i.e., images, animations or virtual environments.

The final approach to visualization design is not a formal design strategy or methodology but is mentioned only because it has the potential to affect the type of visualization applications available on the web. It involves the exploration and testing of technical possibilities for visualization through developing web-based open source applications by experienced graphics programmers who will probably have worked in creative environments reliant on computer graphics who create visualizations for fun, as does Jim Bumgardner who among other things has contributed open source software to flickr, an online photo management and sharing application (Bumgardner, 2008).

A SURVEY OF VISUALIZATION IN USE TO SUPPORT SOCIAL NETWORKING ON THE WEB

The current state of visualization to support social networking is poor. The web pages on social networking sites tend not to be visually novel or pleasing. Good graphic design means some websites have a visually appealing header and menu system though there are arguments that web pages should have little visual clutter and that graphical headers are a distraction. We do not get involved in this argument as it is only how information is structured within the pages and the functionality of web-based applications that is really of relevance.

The layout of the web pages within social networking websites tend to be structured into tabular formats with information structured into long web pages that require scrolling, a typical artifact of CMS. Web pages in a blog style are placed in a list ordered by the date the blog entries were created. In websites primarily handling other types of data relevant to social networking such as images, video, film reviews or web bookmarks, the data are again tabulated but the entities are ranked by some property other than date. Often users of these websites will rate the data or other ways are used to calculate the popularity of the data. One of the features making these websites clumsy for social networking is the way different kinds of data are segregated within them i.e., the separation of images from video and the lack of support for other types of files such as pdf, powerpoint or even applets. While it can make sense to segregate data there are activities such as planning a wedding that require integration of a number of activities which may be carried out through social networking sites. Also, if file formats that can be uploaded and viewed cannot be annotated mean then their features of interest must be described in text that is isolated from the file.

Two web applications offering greater visual diversity are Google Earth/Maps (an online geographical application that combines maps and satellite images) and flickr (an online photo organization application).

Google Earth/Maps have advanced functionality allowing real time manipulation of large amounts of geographical information, with an open API to these applications so that further applications can be developed that extend the functionality for particular sets of users such as recreational runners who may use the MapMyRun website to share routes and to plot graphs of changing elevations over the run. Other information can be added for example 3D models of buildings can be attached to the maps or satellite images or outputs from simulations can be visualized within the Google Earth environment. The API is of commercial value to Google as the professional versions of these applications can be extended further than the non-commercial version.

Flickr also has advanced functionality but it really stands out for its graphical novelty. It has an open API but that is not supported by flickr. There are fewer stand-alone applications extending the functionality of flickr through the API than with Google Earth. One exception is the
trip planner, using flickr’s mapping functionality to link photos to a geographical location on a map, extended to allow planning trips in a shared environment. Flickr is graphically pleasing and especially within its open source community there is graphical experimentation, perhaps because this community is interested in photography and as such is aesthetically motivated.

FUTURE TRENDS

Though a great deal is happening in the field of Visualization, it is not yet a significant presence in the public life of the internet, though this situation will probably change drastically. With many fundamental problems of making information available over the internet now solved, demand for improvements to both presentational quality of information and to the appearance of websites and stand alone web-based applications will surely increase. This will bring the existing and evolving techniques of visualization to much wider attention. Google Earth and Google Maps, a visualization application, have ‘introduced hundreds of millions of users to what have, in the years since Ivan Sutherland’s Sketchpad, been concepts at the core of visualization research—projections, 3D user interaction, user feedback, motion models, level of detail, frame-rate management, view-dependent rendering, streamed textures, multimodal data composition, data-driven extensibility, and direct manipulation’ (Rhyne, 2007).

Visualization may further facilitate user’s access to and control over web content through contribution to the development of code writing. The development of techniques to support this task involves either the use of visualization to aid the writing of textual programs or the development of visual metaphors providing a visual language. In the first case the user must still understand the principals of programming, but techniques from text visualization may be appropriate (Card 2004) or functionality similar to that used in debugging packages could be applied. In the latter case the metaphors will only be helpful if they are apt but harmful if they are not and it is difficult to make them apt (Ware, 1999).

Using the web to aid academic research has been popular from the start but it seems that the semantic web will extend that popularity, aiding researchers in an ever increasing number of ways (Waldrop, 2008). Sharing textual information is the underlying means of communication. However other forms of communication will develop such as the access grid which allows video conference style communication but to multiple users is also popular for the delivery of seminars, conferences and meetings. Visualization could be used to structure information, for disciplines that have a geometrical or geographical dimension (such as engineering, anatomy or archeology) to share shape and location and in fields where there is intense cross-disciplinary activity which is not currently well supported. This trend could result in greater demands for more effective real time display across the internet, involving much greater visual access, and calling for shared visual environments. The semantic web also offers an important means of communication to hobbyists and intelligent lay people requiring similar functionality for different application areas such as football, patchwork quilting and dieting.

Visualization will become more relevant to domestic and leisure uses of the internet as the display and input devices available multiply in form, variety and functionality, with their presentational quality enhanced. With ever larger screens and projection systems used in home entertainment suites social groups and families could benefit from advances made in virtual environments (such as stereographic displays and the space mouse which has similar functionality to the Wii controller) and the access grid allowing groups to socialize through video streaming. Also the increasing array of input devices means that online gaming and training simulators can become more natural.
Comparable opportunities exist with mobile devices, in connection with problems of visualization for small screen spaces, the servicing of touch screens, the development of menu systems and the progressive improvement in interactivity. Google Earth can be accessed on mobile systems and this has the potential to turn any mobile device into a GPS (Global Positioning System).

Visualization can also contribute to the interfaces underlying social networking services. For example, Mayaviz (Roth, 2004) developed a system that uses web-based technology to help synchronize complex logistical efforts similar to planning social events. The ideas used in this product address many of the clunky features noticed in current social networking sites. Worksheet like constructs can be created by users and access can be limited appropriately. Data files of any type can be dragged and dropped anywhere on the sheet (they do not have to be in a tabular layout) and users can make clusters of elements holding related information. Data elements can be drawn on and annotated by the user and changes in data/information can be propagated into other related data without the user having to control these changes. An alternate example is improvements to techniques rather than systems such as to graphical representations of large social networks or to combine both social connectednesses in virtual space with geographic relations on a map.

With very powerful levels of functionality now available on the internet, there is the possibility of a rebalancing the emphasis toward more aesthetic and ease-of-use concerns. Future generations of technology will be integrated into the semantic web to provide superior audiovisual experiences. To this end visualization practices themselves will continue to evolve, taking advantage of the new devices and innovative functionality that continues to come on-stream. Not only will visualization become more commercially driven but more frontal emphasis to aesthetic quality is likely as it becomes more intensely cooperative with graphic designers and creative artists in forming the new user interfaces of the semantic web.

REFERENCES


CONCLUSION

The internet may be on the edge of a golden age for visualization as its techniques find use in many more application areas than those specialized niches within which it has developed. Much of the more complex functionality and varied applications will probably develop commercially but there is still a place for academic research into visualization and for the single open source developer to create interesting visual techniques.


**KEY TERMS AND DEFINITIONS**

**API:** (Application Program Interface) contains all the elements that a programmer needs to extend an application.

**CMS:** (Content Management System) a web delivery system separating content from presentation. These allow users to add content making them popular in social networking sites but the web pages have a tabular form that isolates the elements that make up the content.

**Data visualization:** The second area of visualization to emerge that focused on statistical plots and thematic cartography has now merged with information visualization.

**Flash:** is a multimedia web application adding animation and interactivity to web pages by using efficient streaming and vector graphics techniques.

**Google Earth:** The most popular visualization tool ever. It is a standalone web application combining maps, satellite and GIS information into a meaningful spatial context that gives the user direct manipulation of the applications elements.

**Information visualization:** The final area of visualization to emerge that aimed to show visually the relationships within databases.

**Java:** A powerful programming language that adds functionality to the web at the server side and the client (browser) side. Several libraries relevant to visualization are included within Java.

**JavaScript:** A scripting language that adds interactivity into web pages.

**Scientific visualization:** The first distinct area of visualization to have developed. Initially computer graphics technology was used to “view” the result of computer simulations which had an inherent geometry e.g., the flow of air over an aircraft.

**Visualization:** There are many definitions of visualization. In this chapter we use the term to cover the use of computer and computer graphics technology to present data to aid human understanding and communication. Today visualization is somewhat arbitrarily divided into scientific and information visualization.

**VRML:** is a file format that holds 3D models. Some animation and interactivity is encoded into the file.
ENDNOTES

1 This does not mean graphic design is unimportant or that the structure and layout of information is not part of the aim of visualization. However the original web language (HTML) made no allowance for design so that constructs such as html tables have been adapted to take on a dual role i.e., one of enforcing a design.

2 Single machine is stated here because the visualization system could run on a cluster where the process is split over several machines. The ‘unloaded’ term is more important. On supercomputers there may be other users taking control of resources such as memory, processing power or I/O systems that affect the visualization system’s ability to produce the optimum frame rate. On machines dedicated to the use of a single user this may still be a problem. If a machine is running background processes for example installing updates or if the user is using other software at the same time then there may also be a conflict in the sharing of resources affecting the frame rate.

3 The authors have worked with three CMS Joomla!, Zope and BSCW.

4 The authors could not view every possible website however they attempted to view websites that supported as many different activities as possible: YouTube (http://uk.youtube.com/), Flickr (http://www.flickr.com/), Fantasy Football (http://fantasyfootball.metro.co.uk/fantasy-games/), FaceBook (http://www.facebook.com/), LinkedIn (http://www.linkedin.com/), TopCoder (http://www.topcoder.com), ManyEyes (http://services.alphaworks.ibm.com/manyeyes/home), Del.icio.us (http://delicious.com/), Jango (http://www.jango.com/), Wikipedia (http://www.wikipedia.org/), MapMyRun (http://www.mapmyrun.com/), FetchEveryone (http://www.fetcheveryone.com/) and Now Public (http://www.nowpublic.com/). A blog was also developed on the blogspot website (https://www.blogger.com/start) that uses Google’s blogger interface.