Virtual Worlds for Web Site Visualisation

Danita Hartley Neville Churcher

Software Visualisation Group, Department of Computer Science, University of Canterbury,
Private Bag 4800, Christchurch, New Zealand
dah70@student.canterbury.ac.nz,neville@cosc.canterbury.ac.nz

Greg Albertson

Information Services Department, University of Canterbury,
Private Bag 4800, Christchurch, New Zealand
g.albertson@regy.canterbury.ac.nz

Abstract

Internet and intranet web sites have become an integral part of software development as well as the wider fields of commercial, educational and recreational activity. In order to improve our understanding of web based software engineering we must first understand how web sites are used. Information about the structure and usage of sites is valuable to administrators, maintainers, content developers and users. The volume and complexity of the data generated by typical tools is a major limitation. We describe the use of virtual worlds, including a novel extension of the perspective wall, for visualising web site activity. Examples from our university environment are presented and discussed.

1 Introduction

Few major commercial or educational enterprises would expect to conduct their business effectively without an Internet presence. Typically one or more web sites are used for internal or external purposes. Increasingly, software engineering takes place in an Internet/Intranet environment. Software artifacts as well as processes are developed, delivered and deployed using web-based technology.

User manuals and technical documentation are routinely supplied as HyperText Mark-up Language (HTML) format—often with the intention of being accessed remotely rather than being downloaded *in toto* by users. Effective re-use depends on the ability to locate suitable components and their APIs. The Java development kit includes a tool, javadoc, which generates API documentation in HTML format and similar tools have been in use for some time. Literate programming tools [18, for example], provide yet another source of web-based software artifacts and one of the ways that source code is also being used in a web-based form.

It is important for software engineers to be able to comprehend, measure and manage web-based systems as well as those of a more traditional development environment. As with other aspects of software engineering, the size and complexity of web sites are challenging to conventional techniques.

In this paper we describe some of our applications of information visualisation techniques to web site log data. Similar ideas have been applied in other areas of software engineering [3]. A better understanding of web site usage can then be applied to issues specific to software engineering as well as those of more general applicability.

Web server software varies considerably but individual products are capable of tracking activity in the form of log files of various kinds. Such files are typically rather too large (40MB per week in our case) for ready comprehension by human readers. File and version management involves maintaining the file system structure, dealing with updated versions of individual resources and the addition of new material. Hyperlink management includes the removal of "dead" links and the creation of new links in response to observed user demand. Although the power and sophistication of the tools available to assist site maintainers is increasing, it is still difficult to manage both fine detail and large scale information.

There are many reasons why information about the structure of a site and the pattern of visitor behaviour is useful. These range from site design to marketing banner advertising. Questions of interest include:

- where are our users?
- when do they visit us?
- what do they do while they are here?
- do they find what they want?
- what does our site look like in terms of Hyperlink and file system structures?
- how is our site evolving?

In order to answer such questions access is required to information of various granularities. Bulk data, such as "average number of visitors per day", is aggregated from log records. Other data, such as an individual user trail, is at a much finer level.

While a number of tools, commercial and otherwise, are available to assist the designers, maintainers and managers of web sites the increasing number and diversity of possible features has inevitably led to a lack of coverage in some areas.

Our approach is to attempt to augment the tools available with non-immersive virtual reality visualisations. We are specifically interested in developing systems which are effective on standard workstations. The techniques we describe are also applicable in other domains.

The remainder of the paper is structured as follows. In the next section we outline our reasons for using virtual worlds in visualisation. In section 3 we describe the available data and discuss some potential limitations. Some of our results are presented in section 4 and our conclusions, together with an outline of our future work, appear in section 5. It is difficult to describe virtual worlds adequately through text and pictures and we encourage the reader to visit `http://www.cosc.canterbury.ac.nz/research/RG/svg/intranet-vis/` to experience the worlds described in the paper.

2 Visualisation and VR

In recent years, visualisation has broadened into several related areas. Scientific visualisation [9, 14, 6] has found many applications in engineering and physical sciences while information visualisation [12, 16] is concerned with such applications as navigation in web sites. Visualisation of computational artifacts such as diagrams [21] and software visualisation [1, 4, 17, 13], including both static structure and dynamic behaviour of software, have become established as separate fields.

Each is concerned with enhancing understanding of complex systems by mapping properties of the target system to attributes of one or more visual representations. For example, a file manager might represent individual files by shapes whose geometries represent their type (spheres for text, cubes for images, cones for executables, . . .) whose radii represent the file sizes and whose colours represent their ages.

Visualisations vary considerably in scope and sophistication. We are interested primarily in "quick and dirty" and "just in time" techniques which encourage exploration and experimentation without requiring the services of specialist graphic designers and other professionals.

Virtual reality (VR) has advanced from the realm of fiction [10, for example] and has been applied in many different domains. Both the general aspects of VR [19, 24, 20] as well as more technical issues [23, 5, 22, 11] have been described extensively.

In order to achieve our aim of delivering useful visualisations on standard platforms, we limit ourselves to non-immersive VR—achieved by using an ordinary display screen to give the impression of navigating through a 3-dimensional space—and will not be concerned with the data gloves, helmets and specialised systems found in immersive VR environments.

We use the Virtual Reality Modelling Language (VRML) [2] to represent our results. VRML is essentially a scene description language and is more suitable for our purposes than libraries such as java3d which extend general purpose programming languages.

A VRML world is described in terms of a tree (scene graph) of nodes, each of which has a number of fields. Over 50 primitive nodes types of various categories (such as grouping, geometry, sensors and interpolators) are provided. Some nodes generate events which are routed to other nodes to achieve dynamic behaviour. Events are generated by sensor nodes or when the field values of nodes change. A wiring diagram shows how events, generated by sensor nodes or changes in field values, are routed to other nodes to achieve dynamic behaviour. Custom node types may be written and dynamic behaviour is extended through Java or JavaScript code wired to script nodes.

The appealing features of VRML include a simple text format amenable to generation by software tools. Browsers for displaying VRML worlds are freely available for many platforms, often as plug-ins for Internet browsers such as Netscape. Users require no special knowledge or training. If a visualisation is "good" then it will be easy to use and understand.

3 Available data

Our study involved part of the University of Canterbury site, `http://www.canterbury.ac.nz`. The data reported in this paper covers October 1999 plus the second weeks of August, September and November 1999.

The server log files are flat text files whose detailed content varies according to which server software product is in use and configuration options selected. The log records show the temporal order of requests, together with details of the requester and the resources involved. Figure 1 shows the corresponding data model.

The reader is encouraged to visit `http://www.cosc.canterbury.ac.nz/research/RG/svg/intranet-vis/` for details of the format of the data available for our study together with some sample data records.

Since the logs contain no direct information about the link structure, our picture of user activity is a series of snapshots rather than a smooth animation. We do not know precisely which link a user followed to get from one page to another, whether a bookmark was used or whether a URL was typed in directly. Sometimes some of this additional information is available via referrer logs or other instrumentation. However, this is not widely used as the volumes of data involved can be very large and there is little direct benefit to site administrators. Consequently, we cannot rely on the availability of all the desired data.

Our challenge is to make the most effective use of what is available. There are many additional sources of uncertainty, such as dynamically assigned IP numbers, so we aim to construct visualisations which highlight the underlying effects without becoming buried in the "noise" resulting from the limitations of available data.

In some cases it is possible to infer additional information. For example, noting that a 72 byte server response almost always corresponded to a request for a cached resource enables us to identify such requests.

Our initial naïve approach to world generation involved processing the log files with various filters. Subsequently, a more flexible and extensible approach capable of supporting *ad hoc* queries was developed. The log files are parsed and stored in a database whose schema is based on the data model shown in Figure 1.

4 Results

In this section we describe some of the visualisations we have developed. Please visit `http://www.cosc.canterbury.ac.nz/research/RG/svg/intranet-vis` to experience these worlds for yourself.

The data reported in this paper covers October 1999 plus the second weeks of August, September and November 1999. During this 52 day period 26595 distinct visitor IP numbers and 2080254 requests for 4085 distinct resources were logged.

Around 20% of requests originating from on-campus IP numbers, and 12% of those from off-campus, are for cached resources. Only 0.69% of requests result in errors. Most (63%) of these are from off-campus users who are arguable more likely to be typing in unfamiliar URLs. We were surprised to find about 60% of the requests were for image files.

4.1 Site characteristics

Figure 2 shows a VRML browser implemented as a Netscape plug-in. A range of controls is provided to allow navigation through and manipulation of the world. The precise effects of some controls vary according to the navigation mode—"flying", "walking" or "examining". We include an artificial horizon to help prevent the user from becoming disoriented.

This simple world demonstrates some basic mappings. Files/resources have been separated by type: documents (.html, .ps, .doc, ...) are shown as cubes; images (.gif, .jpg, ...) are omitted; others (cgi scripts, ...) are represented as spheres.

Colour is used to indicate the frequency of resource access. Darker objects are less frequently requested and white objects are the

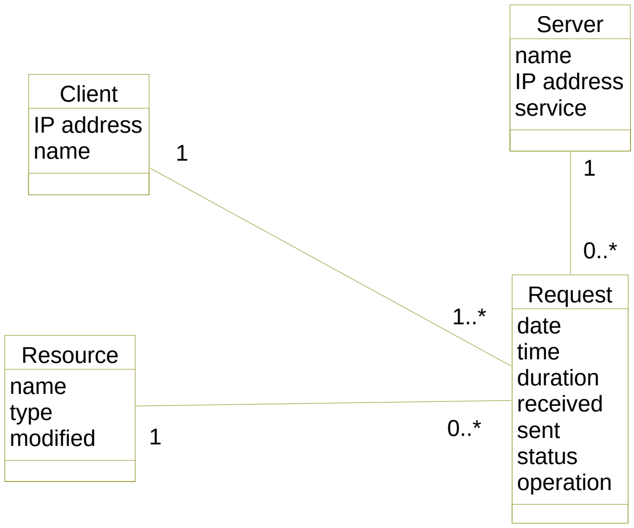


Figure 1: Data model (simplified)

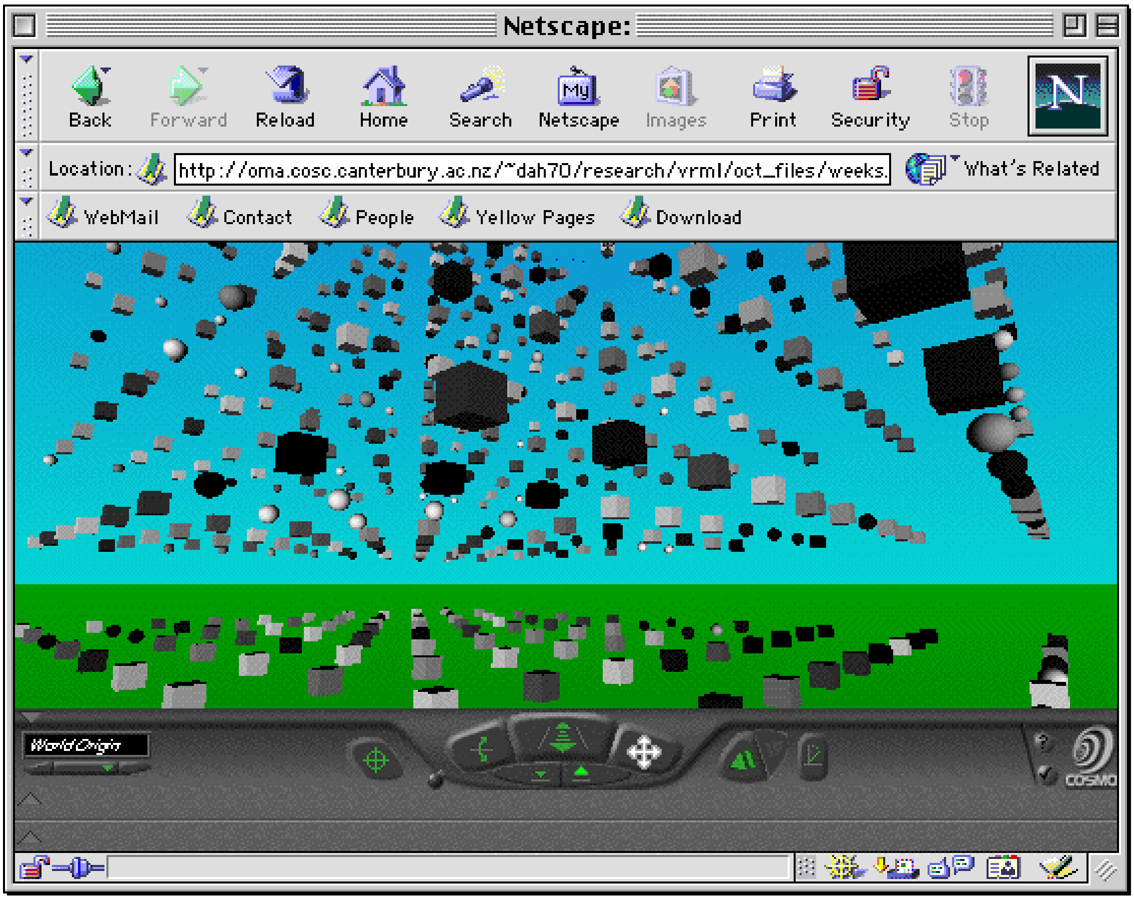


Figure 2: VRML browser & site characteristics

most frequently requested. More complex mappings allow attributes such as age and size to be represented by properties such as colour depth and transparency.

Animation is used in this world to show how the access frequency changes from week to week for the month of October 1999. Resources whose colour changes as the weeks cycle may then be distinguished from those whose access frequency is more steady.

The geometrical representations of individual resources have been laid out on a simple cubic grid in no particular order. The perspective effect is provided by the browser. Other layout strategies might be based on resource properties (type, age, size, ...) or take account of the corresponding hyperlink connectivity or file system structure. Refinements include adding connections representing the probability that one resource will be visited after another to allow high traffic paths to be identified.

4.2 Static visualisation of activity

The worlds shown in Figures 3(a) and 3(b) are based on a surface representation summarising the activity for a single week. Labels have been added in Figure 3(a) and elsewhere to help orient the reader—these are not part of the virtual world. The shorter axis represents the day of the week and the longer axis represents hour of the day. The height of the surface represents the aggregated amount of activity in the corresponding hour of the day (e.g. 3–4 a.m. Monday).

Figure 3(a) is a composite showing data for the second week of four consecutive months. Individual weeks are shown in different colours. August 1999 is in the foreground and the corresponding weeks for later months stretch off into the background. The viewpoint shown is looking from midnight towards earlier times. The leftmost of the two surfaces shown represents the number of visitors (unique client IP addresses) and the other represents the number of requests (accesses).

In a one hour period on a typical weekday accesses would peak at around 400–500 visitors and 4000–6000 requests. The heights for both surfaces have been scaled (by 0.1 for the number of visitor IP numbers and 0.01 for the number of requests) for purely aesthetic reasons. As we have argued elsewhere [3] a major advantage of virtual worlds is their ability to handle extreme values more naturally than other presentation techniques—the user need only "look up" to see the large peaks and is free to wander about on the flatter regions or fly up the steep faces at will.

Many of the visual attributes employed in visualisation, such as colour and texture, are less suitable for precise measurements but rather are designed to highlight trends and relationships. It is certainly possible to include geometry in the worlds to represent elements such as co-ordinate axes, and we use such techniques occasionally.

Combining surfaces for several weeks in this way allows trends to be examined. For example, the peaks become lower in November when the academic year is over. Other variants, such as periodic boundary conditions, are also interesting.

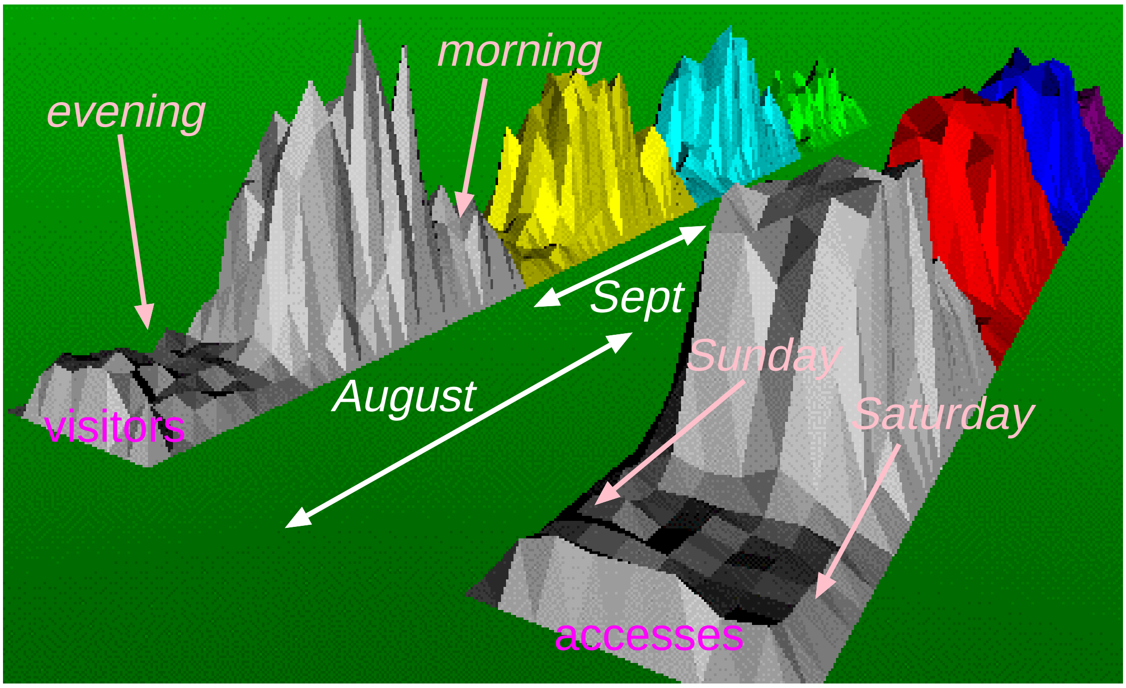
Variations on this simple visualisation may be used to highlight aspects of interest. Figure 3(b) includes a semi-opaque plane indicating the mean over the whole week with heights representing the differences above and below the mean value. Other possibilities include using colour to represent the deviation from the mean, further planes for quartiles and animation to show how the surface changes over time.

The data in Figures 3(a) and 3(b) indicate that peak access occurs, as might be expected, during the middle of the day during the working week. However, other features are also apparent. There is a small but noticeable increase in activity around Sunday lunch time; lower than average activity on Friday afternoons; a small peak around midnight might indicate regularly scheduled unattended activity.

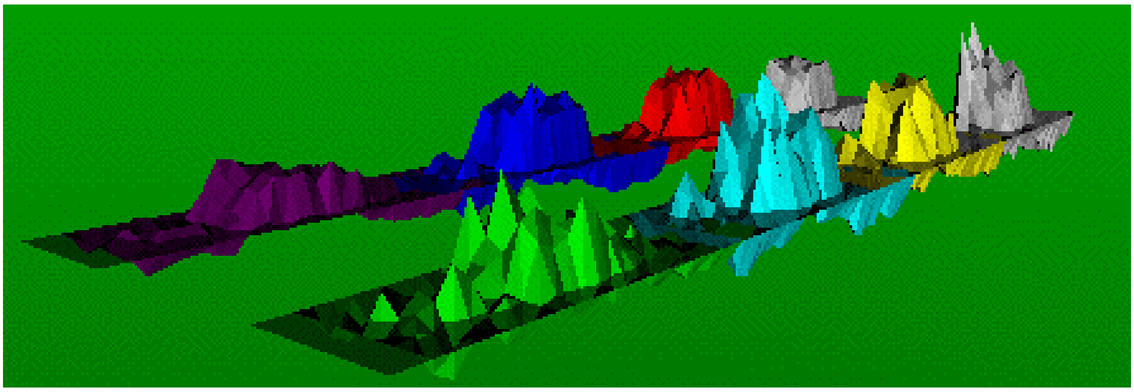
4.3 Spindicators and perspective wall

A rather different visualisation of the activity data presented in section 4.2 is shown in Figures 4 and 5. Here the emphasis is not on the static summary features but rather on visualisation of the dynamic variation of values with time (day of the week or hour of the day).

Each of the animated sphere/pointer combinations (which we term "spindicators") represents a particular slice through a surface such as the visitor surface of Figure 3(a). Those in Figures 4 and 5(b) each represent one day of the week while those of Figure 5(c) each represent an hour of the day interval.



(a) Visitor IP and resource accesses



(b) Deviations from global mean

Figure 3: Visitor IP and resource accesses from week 2 August–November 1999

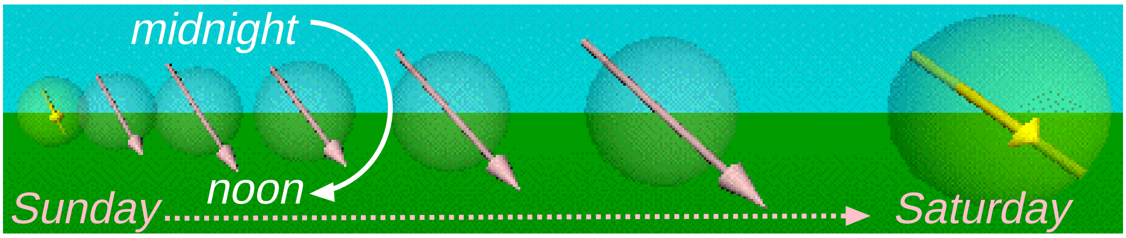


Figure 4: Spindicators (One per day: Visitor IP counts week 2, August 1999)

Each spindicator in Figure 4 is configured to represent aspects of the number of distinct IP numbers seen during one day of the week. The radius of the spheres indicates the daily average for the week, providing a reference level in much the same way as the plane incorporated in the world of Figure 3(b). The angles of the pointers represent the hour of the day (vertical at midnight) and the pointer lengths change to indicate the corresponding activity (number of distinct visitor IP numbers) for that hour. Figure 4 shows a snapshot at around 9:00 a.m. in the second week of August 1999. At this time of day there is little activity on Saturday and Sunday while slightly above average visitor numbers are observed during the working week.

The pointer lengths are interpolated smoothly as the spindicators rotate cyclically, allowing ready observation of changing patterns. Greater resolution can be obtained by aggregating in smaller time intervals.

We have combined our spindicators with the principle of the perspective wall [15] to provide a means of handling larger numbers of spindicators. The fundamental idea in the perspective wall is to show a region of interest in full detail with progressively less detail evident in the peripheral regions. This is a particular application of the more general idea of fisheye views [8].

In the original form of the perspective wall, as shown in Figure 5(a), the region of high detail (focus) is a rectangular region. Less detail of the function range is visible for domain values further from the focal region.

Our variation emphasises the periodic aspects of the data and the navigability of the VRML worlds. Individual spindicators are arranged in a circle around a central cylindrical plinth. There is no fixed focal region and no specific controls are needed as users simply move around the wall in exactly the same way that they perform other navigation. The perspective effects come "for free" without additional processing. The plinth serves to obscure components behind the wall as seen from the users current location in the ground plane. Changing location, or removing the plinth, reveals the full circle of components.

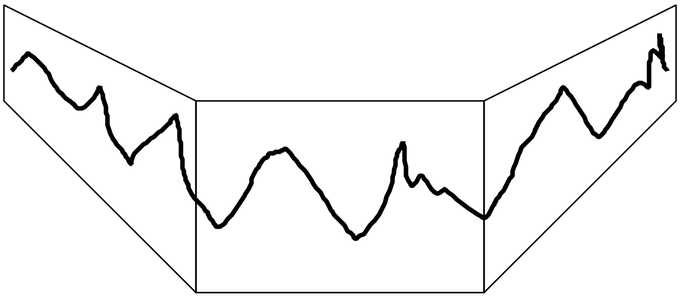
Figure 5(b) shows an overhead snapshot view of a world corresponding to that of Figure 4. The (yellow) spindicator at the left represents Saturday. This view illustrates the structure clearly. Although the time of day is not readily apparent in this snapshot, the animation provides a much firmer impression of this variable.

Figure 5(c) shows another perspective spindicator world containing 24 indicators, one for each hour of the day, whose rotation represents the weekly cycle. Vertical pointer orientation would correspond to Sunday—this snapshot is for Thursday. The viewpoint is looking towards noon data with morning on the left and afternoon on the right. Data for night times is behind the plinth and pointer colour is used as an additional identifier for spindicators showing nocturnal activity. The perspective effects are evident at the edges of the figure while more detail is visible in the (closer) central region.

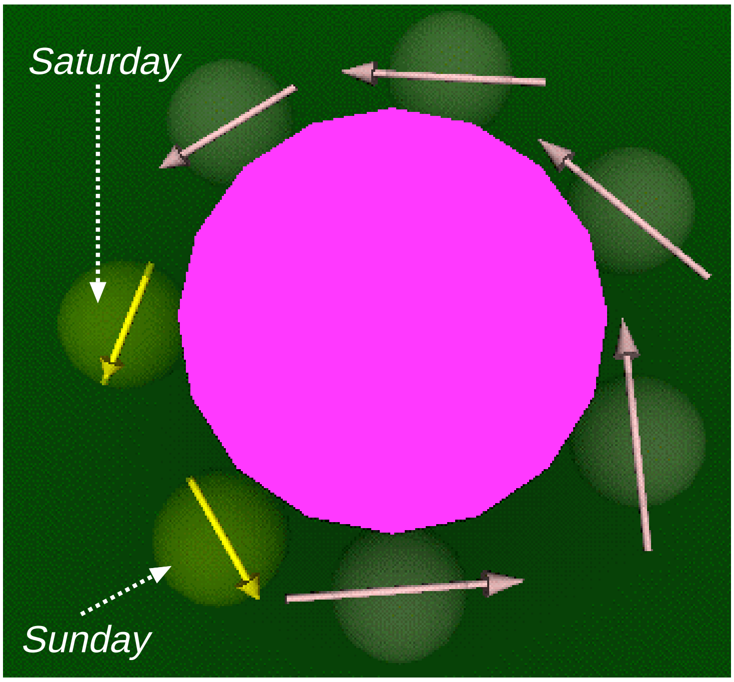
4.4 Visualising differences

This section describes an example of more focused visualisation. One reason suggested for the observed level of nocturnal activity was the presence of visitors from other time zones. In this section we show how partially transparent geometry can be used to highlight the differences between surfaces. Techniques such as that used in Figure 3(b) can also be used.

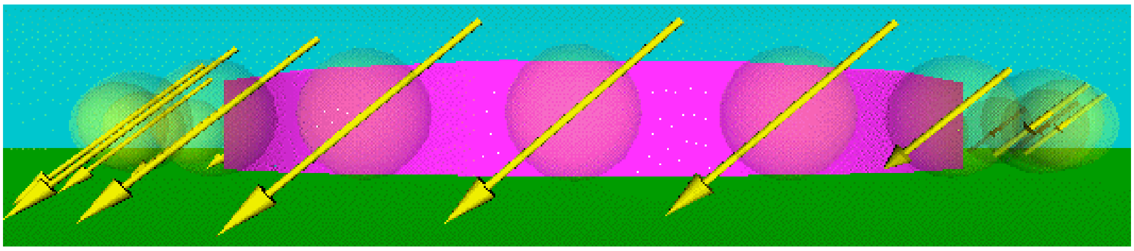
Figure 6(a) shows the aggregate activity surfaces from the first 4 weeks of October 1999,



(a) Perspective wall [15]

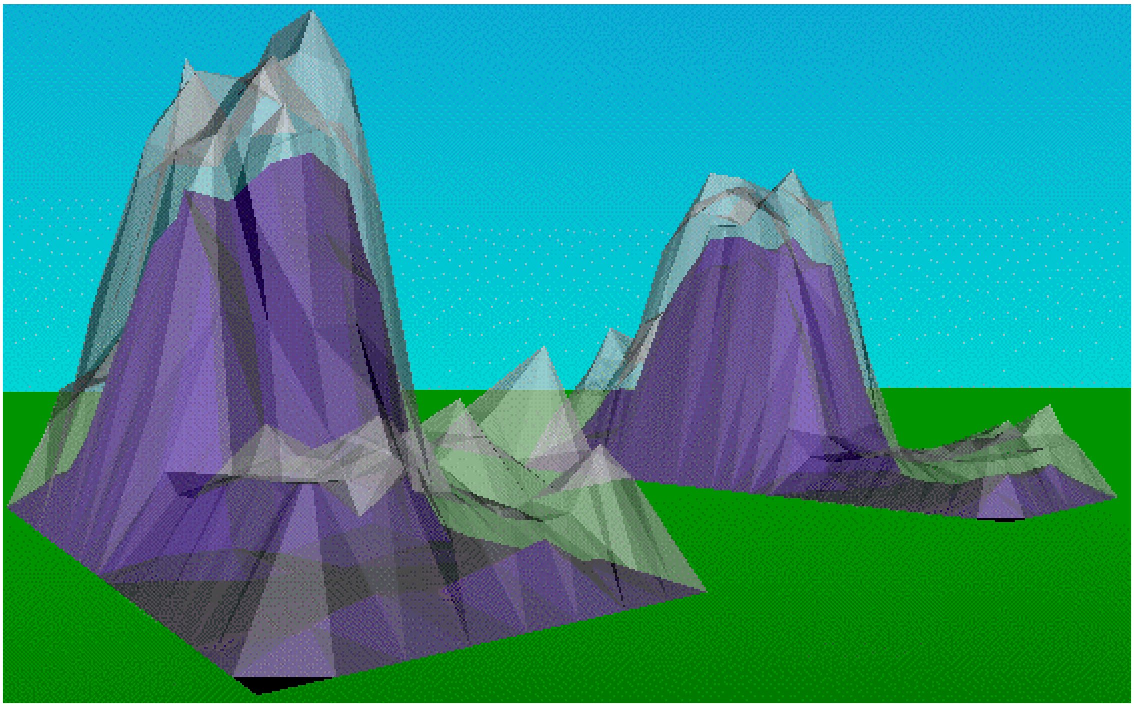


(b) Perspective version of Figure 4 viewed from above

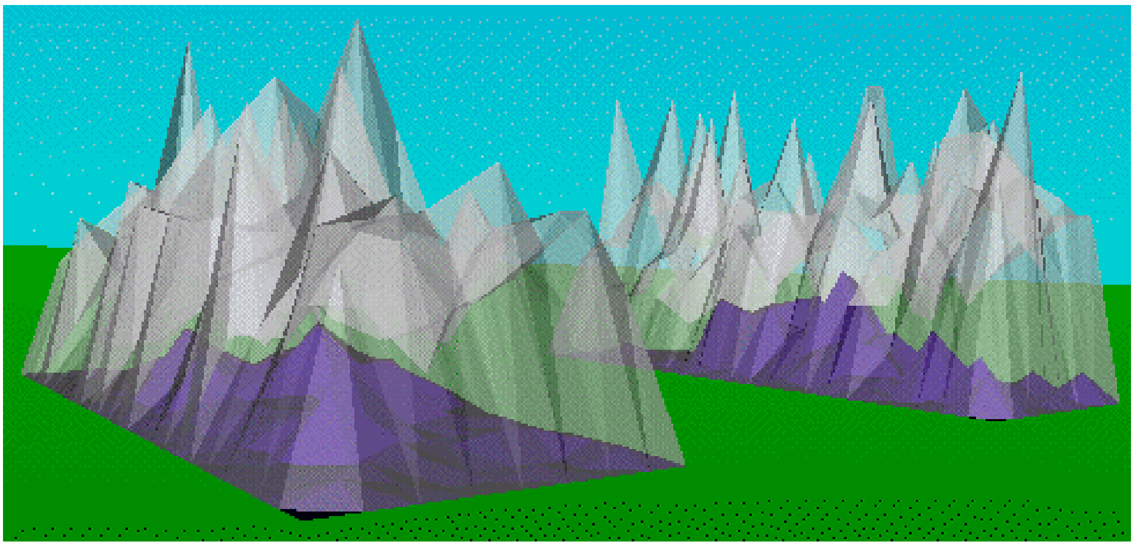


(c) One spindicator per hour

Figure 5: Visitor IP counts week 2, August 1999



(a) All resources



(b) International student pages

Figure 6: All visitor and on-campus visitor activity for weeks 1–4 October 1999

viewed from the late-night end of the visitor surface, as a transparent surface. The corresponding activity from visitors whose IP numbers are known to be on-campus is represented by the superimposed (purple) lower surfaces. Clearly, off-campus users account for much of the late night activity.

A further hypothesis was that off-campus visitors were more likely than on-campus visitors to access the part of the site which contains material about international students. Figure 6(b) suggests that this hypothesis is supported by the data. Such information about the peaks and trends in access to these pages is of interest to those who market Canterbury internationally.

4.5 User trails

It is difficult to obtain accurate data on the trails followed by individual visitors. One reason for this is the absence of a transaction model. There is no representation of the beginning or end of an interaction and features such as bookmarks and caching complicate matters. A referrer log, which provides specific details of the page and link traversals made by users, was not available on our site. However, by considering a large number of visitor trails we hope to emphasise genuine trends in our visualisation. The absence of the referrer log data is quite typical so we must expect to have to improvise. In our case, we assume that the sequence of pages visited results from link selection rather than bookmarks or directly entering URLs in the browser. It is not uncommon for links to be followed within the one second resolution of the timestamps. In such cases we infer the ordering from the order in the log. This order is preserved in our database.

Figure 7 shows a world representing data about the "news and events" section of the site as an undirected graph. One important but tangential issue we encountered concerns algorithms for 3D graph layout. The approach we used is a relatively crude implementation of the force-directed approach [7].

The 16 relevant pages (nodes) are represented by spheres whose radius indicates the number of times the corresponding page was accessed during an 8 week period. The node colour indicates the visitor origin: red if $> 65\%$ of accesses are from on-campus IP numbers, blue if $> 65\%$ of accesses are from off-campus IP numbers and green otherwise.

Edges in the graph represent the hyperlink connectivity of the pages. The coloured edges indicate the path taken by a single on-campus user between 2:16 and 2:41 p.m. on October 8th 1999. The thickness of the coloured edges indicates how many times the corresponding link was followed.

Care must be exercised in interpreting such graphs as there are a number of user actions which may not leave a log record. However, we believe that this representation is useful for many purposes. For example, it is interesting to follow the trails left by a number of visitors who enter the site at the same node.

5 Conclusions and further work

The sheer volume of data generated by logging web server traffic means that some form of visualisation is essential in order to allow administrators to be aware of trends and unusual events. Virtual worlds implemented in VRML are suitable for low-tech on-demand visualisations. Such worlds need to be combined with database, statistical or other applications where detailed numerical measurements are required. However, they are particularly effective in a variety of situations.

Our future research in this area has a number of directions. We wish to collect data for some additional sites and perform some comparative analyses. Further automating both the data processing and world update aspects of our software will deliver tools which are both more powerful and easier to use. New visualisations are under development. Anecdotal evidence suggests that our visualisations are useful for the administrators of our site. However, user trials are needed in order to determine the most effective worlds for particular tasks.

Ultimately, we hope that increased understanding of web sites will benefit software engineers in all aspects of their work.

References

[1] T. Ball and S.G. Eick. Software visualization in the large. *IEEE Computer*, 29(4):33–43, April 1996.

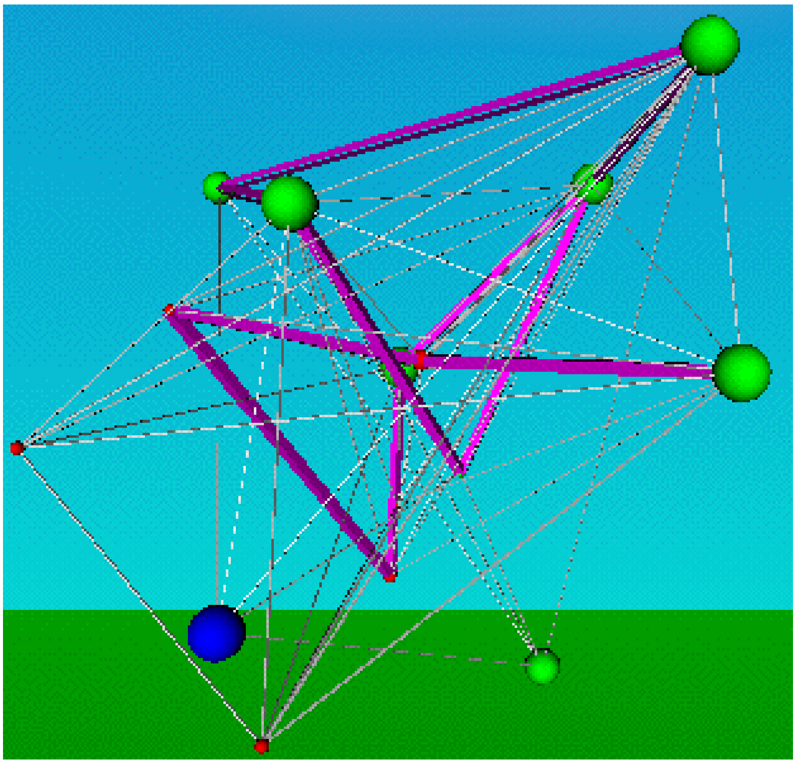


Figure 7: Visitor trails in the news and events pages

[2] R. Carey and G. Bell. *The Annotated VRML 2.0 Reference manual*. Addison-Wesley, 1997.

[3] N.I. Churcher, L.M. Leown, and W. Irwin. Virtual worlds for software visualisation. In A. Quigley, editor, *SoftVis99 Software Visualisation Workshop*, pages 9–16, University of Technology, Sydney, Australia, December 1999.

[4] P. Eades and K. Zhang, editors. *Software Visualisation*, volume 7 of *Series on Software Engineering and Knowledge Engineering*. World Scientific, 1996.

[5] R.A. Earnshaw, M.A. Gigante, and H. Jones, editors. *Virtual Reality Systems*. Academic Press, 1993.

[6] R.A. Earnshaw and N. Wiseman. *An Introductory Guide to Scientific Visualization*. Springer-Verlag, 1992.

[7] T.M.J. Fruchterman and E.M. Reingold. Graph drawing by force-directed placement. *Software—Practice and Experience*, 21:1129–1164, 1991.

[8] G.W. Furnas. Generalised fisheye views. In *Proc ACM SIGCHI '86 Conference on Human Factors in Computing Systems*, pages 16–23, 1986.

[9] R.S. Gallagher, editor. *Computer Visualization: Graphics Techniques for Scientific and Engineering Analysis*. CRC Press, 1995.

[10] W. Gibson. *Neuromancer*. Gollancz, London, 1984.

[11] F. Hamit. *Virtual Reality and the Exploration of Cyberspace*. SAMS Publishing, 1993.

[12] J.D. Hollan and B.B. Bederson. Information visualization. In M.G. Helander, T.K. Landauer, and P.V. Prabhu, editors, *Handbook of Human-Computer Interation*, chapter 2, pages 33–48. North-Holland, 2nd edition, 1997.

[13] C.L. Jeffery. *Program Monitoring and Visualisation: An Exploratory Approach*. Springer-Verlag, 1999.

[14] P.R. Keller and M.M. Keller. *Visual Cues: Practical Data Visualisation*. IEEE Press, 1993.

[15] J.D. Mackinlay, G.G. Robertson, and S.K. Card. The perspective wall: Detail and context smoothly integrated. In *Proc ACM SIGCHI '91 Conf. on Human Factors in Computing Systems*, pages 173–179, New Orleans, Louisiana, April 1991.

[16] R. Paton and I. Neilson, editors. *Visual Representations and Interpretations*. Springer-Verlag, 1999.

[17] B.A. Price, R.M. Baecker, and I.S Small. A principled taxonomy of sofware visualization. *Journal of Visual Languages and Computing*, 4(3):211–266, 1993.

[18] N. Ramsey. Literate programming simplified. *IEEE Software*, 11(5):97–105, September 1994.

[19] H. Rheingold. *Virtual Reality*. Summit Books, 1991.

[20] R. Schroeder. *Possible Worlds: the Social Dynamic of Virtual Reality Technology*. Westview Press, 1996.

[21] T. Strothotte. *Computational Visualisation: Graphics, Abstraction, Interactivity*. Springer-Verlag, 1998.

[22] J. Vince. *Virtual Reality Systems*. Addison-Wesley, 1995.

[23] A. Wexelblat, editor. *Virtual Reality: Applications and Explorations*. Academic Press, 1993.

[24] B. Woolley. *Virtual Worlds*. Blackwell, 1992.