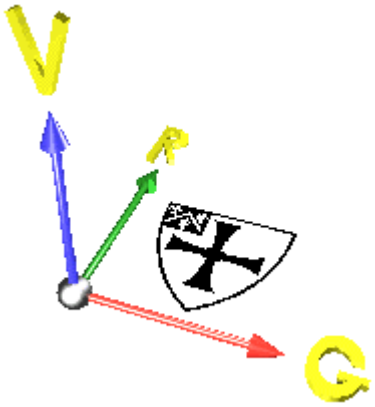
Three Dimensional Information Visualisation



Peter Young
Visualisation Research Group
Research Institute in Software Evolution

Department of Computer Science
University of Durham
Durham
DH1 3LE

http://vrg.dur.ac.uk/

November 1, 1996

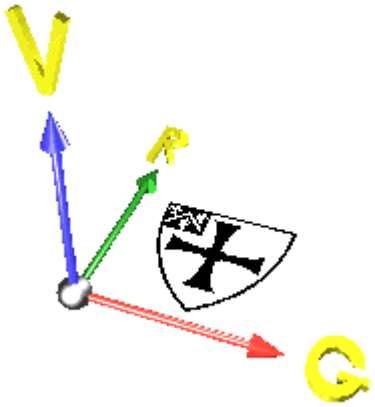
Computer Science Technical Report : 12/96

This document is also available on-line from:
http://vrg.dur.ac.uk/misc/PeterYoung/pages/work/documents/

Please note that many of the images in this document have been colour reversed in order to make the document more suitable for printing. The on-line version found above contains the original screen images.

For more information please http://vrg.dur.ac.uk/, Peter Young left VRG in 1999, PDF created by Claire Knight, 11th March 2002

Three Dimensional Information Visualisation



Peter Young
Visualisation Research Group
Centre for Software Maintenance

Department of Computer Science
University of Durham
Durham
DH1 3LE

E-mail : *peter.young@durham.ac.uk*

November 1, 1996

Computer Science Technical Report : 12/96

This document is also available on-line from:
http://www.dur.ac.uk/~dcs3py/pages/work/documents

Please note that many of the images in this document have been colour reversed in order to make the document more suitable for printing. The on-line version found above contains the original screen images.

Table of Contents

Table of Figures

1. Introduction

The rapid expansion of technology and dwindling hardware costs have resulted in an explosion in the computing power to price ratio. Computer systems are increasing both in processing power and storage capacity at an incredible rate. At the same time the price for this hardware is dropping rapidly resulting in high performance computing being available to the general public. This proliferation in computer availability coupled with the advances in network technology and the advent of the Internet and World Wide Web has resulted in an information big-bang. Unfortunately information is not the correct term to describe this abundance of data which can be described more aptly as 'noise'. The amount of data available now has grown to gargantuan proportions whereas the corresponding growth in actual information is far lower. The problems to overcome in finding relevant information have shifted, the difficulty previously being that information was not easily accessible or searchable. The problem no longer lies in getting the information, more in finding it and sorting out the one useful record from the hundred similar items. The problem at hand is that of *data mining*.

As the information super-highway thunders on towards a global data pile-up, researchers are investigating various tools and techniques to give users the most comfortable and economic ride ensuring they get to their destination safely and promptly. Algorithms and methods for data mining are being developed to automate information filtering, hopefully producing 'intelligent' search systems which can return more relevant information to the needs of the user. On the other hand research into visualisation techniques is receiving much attention, the onus of retrieving relevant data being put more on the user. Information visualisation systems will allow the user to explore the data universe at a more abstract level, harvesting interesting looking items as they are found. These systems will allow users to make use of their cognitive, perceptual and intuitive skills to find data which may be of interest but could be missed by search algorithms because they are not directly relevant to the query.

This report describes a variety of 3D information visualisation techniques and research systems which have been developed to aid the human comprehension of large information systems. The following section describes the techniques currently available for displaying various forms of information. Section 3 then goes on to describe a number of prototype or research visualisation systems and give brief summaries of their abilities. Finally, section 4 attempts to draw some conclusions on the systems and techniques described and also to explain the application of such technology to software visualisation.

2. Visualisation techniques

This section describes a variety of information visualisation techniques which are used in many systems. These techniques range from the familiar data presentation of surface plots and 3D bar charts through to the creation of abstract data spaces and the behaviour of objects within them.

The techniques described here can be classified as belonging roughly to one of a number of groups. Surface plots, cityscapes, Benediktine space and spatial arrangement can all be classified as *mappings* from the data domain to the

visualisation space. These techniques all use some aspect, property or value of the data items to produce a mapping to objects within the visualisation. Perspective walls, cone trees, cam trees and rooms may be classed as information *presentation* techniques. These visualisations concentrate on the appearance, accessibility and usability of the data and aim to provide a user friendly and intuitive interface. Finally, fish-eye views, emotional icons and self organising graphs may all be described as *dynamic* information visualisation techniques. These techniques endow the visualisations with behaviour and dynamic properties, allowing the visualisations to respond automatically to changes in the data or to the actions of the user. These classification are only rough and there is some degree of overlap, for example perspective walls could quite easily be classified under dynamic information visualisation.

The following sections describe each of these techniques in detail.

2.1. Surface plots

One of the most familiar extensions from standard 2D graphs has to be the 3D surface plot. Surface plots are constructed by plotting data triples onto the three co-ordinate axes X, Y and Z. Typically the data will consist of two standard sets which have a regular structure, e.g. days of the week and time of day, and one actual data value for example wind strength. The two regular sets are normally plotted on the horizontal axes X and Z with the variable data being plotted as height in the Y axis. The set of points thus formed are netted into a mesh or surface which is often colour coded to indicate height variations. The resulting visualisation resembles a landscape which can be easily interpreted to identify features such as patterns or irregularities.

2.2. Cityscapes

Cityscapes are basically an extension to 3D bar charts and a variation to surface plots. Cityscapes are created in a similar fashion to the surface plots by mapping scalar data values onto the height of 3D vertical bars or blocks, the blocks being placed on a uniform 2D horizontal plane. The resulting visualisation is a more granular representation of the surface plot. The cityscape demonstration [Walker93] developed at BT included additional features to aid graph comprehension and allow simple comparison of results. One feature projected the minimum, average and maximum values for each row and column in the 2D plane (i.e. X and Z axes) onto the end walls of the cityscape. Another feature allowed the addition of a variable transparency ‘sheet’ to placed at a specified height in the cityscape. The transparent sheet highlighted blocks or data values which exceeded the set height allowing instant identification and comparison between these data values.

2.3. Fish-eye views

The name given to this particular type of view is taken from the similar effect produced by a very wide angle ‘fish-eye’ lens. The fish-eye lens distorts the view so that objects close to the centre of the lens are magnified greatly, this magnification drops rapidly the further from the centre of the lens that the objects are. This view results in objects which are the centre of attention being shown in greatest detail, whereas objects on the periphery are shown in lesser detail. This allows objects of interest to be studied in detail, while still maintaining a view of the context or position of the focus with respect to other objects.

Fisheye views were originally investigated by Furnas [Furnas86] but have since received more attention and widespread use. The original views have been extended to give more control over the layout and to take into account the overall information structure [Sakar92].

The use of fish-eye techniques could prove to be useful in visualising large graphs. The technique will allow nodes of interest to be brought 'closer' to the viewer thus displaying greater detail while other nodes are moved further from the viewer and displayed in lesser detail. This allows the viewer to concentrate on the interesting nodes while still maintaining a picture of these node's position within the whole structure.

2.4. Benediktine space

Benediktine space is a term which arose from Michael Benedikt's research into the structure of Cyberspace [Benedikt91] which in turn was coined by William Gibson in his science fiction short story, *Burning Chrome* [Gibson93]. Benedikt put forward the notion that attributes of an object may be mapped onto intrinsic and extrinsic spatial dimensions. Extrinsic dimensions specify a point within space, for example a set of Cartesian co-ordinates. Intrinsic dimensions specify object attributes such as size, shape, colour, texture, etc. An example of a Benediktine space could be to map an attribute such as student names to the x-axis and their exam marks to the y-axis. The degree which a student received could then be mapped onto an intrinsic dimension such as shape.

Benedikt also proposed two rules for Cyberspace, the principles of *exclusion* and *maximal exclusion*. These rules attempt to clarify the positioning of data and in particular to avoid 'crowding' of data items. The principle of exclusion essentially ensures that no two data items can exist in the same location within space, i.e. their extrinsic dimensions must be different. This is stated as:

> *"Two non-identical objects having the same extrinsic dimensions and dimension values, whether at the same time, or including time as an extrinsic dimension from the outset, is forbidden, no matter what other comparisons may be made between their intrinsic dimensions and values."*

The Principle of Maximal Exclusion expands upon the Principle of Exclusion by ensuring different data items are separated as much as possible, thus avoiding confusion produced by cluttering of objects. This is stated as:

> *"Given any N-dimensional state of a phenomenon, and all the values - actual and possible - on those N-dimensions, choose as extrinsic dimensions - as "space and time" - that set of (two, three, or four) dimensions that will minimise the number of violations of the Principle of Exclusion."*

2.5. Perspective walls

Perspective walls [Mackinlay91] are a technique for viewing and navigating large, linearly structured information, allowing the viewer to focus on a particular area while still maintaining some degree of location or context. Perspective walls pay some similarity to the fisheye views in that they allow a particular area of information to be viewed in detail while information close to this is still visible in lesser detail thus giving an idea of position and orientation within the data.

The two strategies previously used to display large volumes of information were the *space strategy* and the *time strategy*. The space strategy used layout techniques and graphical design to maximise a single display area, presenting as much information as possible. This technique suffered from information overload in that so much information was presented that extracting any detail became harder. The time strategy took the principle of breaking the information structure into a number of separate views, only one of which could be displayed at any one time. This allows the viewer to switch between the views and focus on particular information as needed. The problem suffered by this view was that it was easy to become lost within the data as no overall view or cues to the current location are provided.

The perspective wall addresses the above problems by effectively extending the time strategy with contextual cues. The perspective wall folds the linear structure in 3D space, for example forming a cylindrical shell with the data mapped onto the interior surface (other configurations are possible). A single section of the data may be viewed in detail at any one time, with adjacent sections being folded back on either side of the view to give cues to the position of the current section. On moving between sections the wall would rotate smoothly to bring the next section to the centre of the view.

The perspective wall, cone trees, cam trees and 3D-Rooms (described below) are the result of research into an integrated *information visualizer* [Card91, Clarkson91] at the Xerox Palo Alto Research Center.

2.6. Cone trees and cam trees

Cone trees [Robertson91] are a three-dimensional extension to the more familiar 2D hierarchical tree structures. Cam trees are identical to cone trees except they grow horizontally as opposed to vertically. The aim of cone trees is to allow a greater amount of information to be navigated and displayed in an intuitive manner and also to shift some of the cognitive load of comprehending the structure to the human perceptual system.

Cone trees are constructed by placing the root node at the apex of a translucent cone near the top of the display. All child nodes are then distributed at equal distances along the base of the cone. This process is repeated for every node in the hierarchy, with the base diameter of the cones reducing at each level as the hierarchy descends to ensure sufficient space to accommodate all leaf nodes.

The original cone and cam tree visualisations produced at Xerox PARC enabled the tree to be rotated smoothly to bring any particular node into focus. The smooth animation was found to be critical in maintaining the viewer's cognitive model of the structure. Sudden changes in orientation of the tree would require a significant amount of time to reorientate the user's cognitive model.

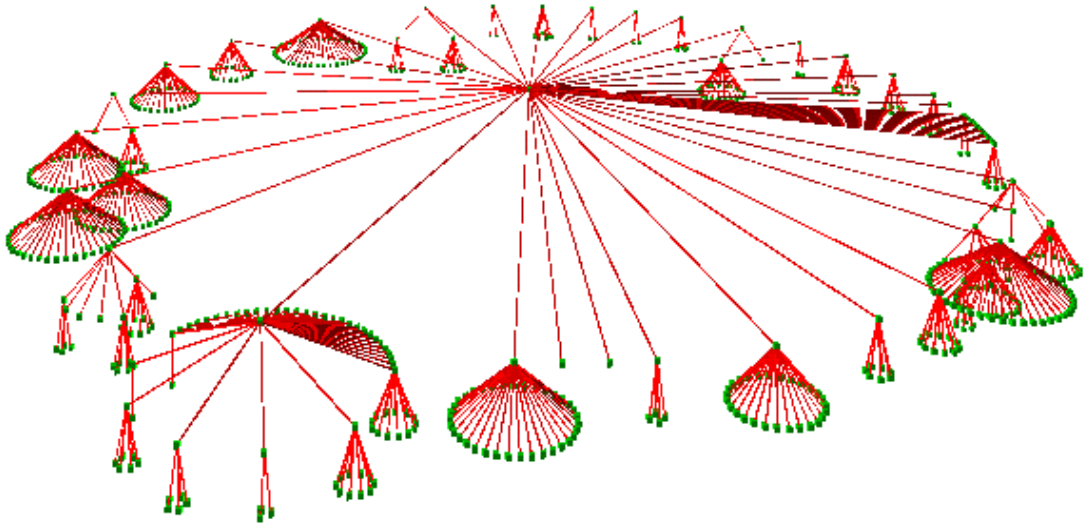


Figure 1. Example visualisation of a cone tree representing a directory structure.
Image courtesy of Dave Snowdon, Nottingham University.
http://www.crg.cs.nott.ac.uk/crg/Research/pits/pits.html

2.7. Sphere visualisation

The sphere visualisation is described by Fairchild *et al.* [Fairchild93] as a 3D version of the predominantly 2D perspective wall (see section 2.5.). The sphere visualisation is used within the VizNet visualisation system to view associative relationships between multimedia objects and a selected *object of interest* (OOI). Objects are mapped onto the surface of a sphere with highly related objects placed close to the OOI. Unrelated objects are displayed further from the OOI and thus become less visible as they move round to the opposite side of the sphere. This provides a natural fisheye view which emphasises objects of interest and de-emphasises less related objects.

The information is presented on the surface of a number of nested spheres. This provides a mechanism for representing different levels of information. The OOI is displayed on the outermost sphere with objects directly related to it fanning out around the surface of the sphere. Objects which are indirectly related to the OOI are considered lower level objects and are displayed on spheres nested within the outer sphere. The colour of the spheres becomes darker with increasing depth of nesting to give the user visual cues to their current location within the visualisation. Navigation through the visualisation is facilitated by rotating the sphere to bring nodes of interest into view and by traversing links to lower level spheres.

2.8. Rooms

The '3D-Rooms' metaphor is a three dimensional counterpart to the desktop metaphor commonly encountered in computing today. 3D-Rooms were developed at Xerox PARC as part of their integrated *information workspace* and are a 3D extension to the original concept of 2D rooms. 2D Rooms [Card87, Henderson86] built upon the notion of a multiple desktop workspace by adding features such as the ability to share the same objects between different workspaces, overview the workspaces and also to load and save workspaces. Rooms allows another way for users to structure and organise their work by allocating certain tasks to certain rooms and moving between

rooms as needed. Within each room their will be a variety of information sources depending on the type of task allocated to a particular room. 3D Objects may be present which represent, for example, documents or applications. The walls of the room may also contain information in a more conventional 2D manner, effectively using the walls as 2D displays.

The various rooms in an information complex such as this are connected via a number of doors. Doors lead from one room directly to another and leaving a room by the back door will return you to the previous room you were in. A necessary navigational aid for these structures is the provision of an overview or floor plan of the rooms, including some indication to the content or tasks of the individual rooms. This allows the user to quickly move between unrelated rooms or tasks without having to go via a number of other rooms. One further useful addition to the rooms metaphor is the notion of *pockets* which allow the user to 'carry' information or objects between rooms or keep important items close at hand.

Figure 2. Sphere visualisation produced using VizNet.
Image courtesy of Kim Fairchild, Institute of Systems Science, National University of Singapore.
http://panda.iss.nus.sg:8000/kids/fair/webdocs/

2.9. Emotional icons

Emotional icons [Walker95] taken within the context of a 3D data world, are objects which perform varying behaviour in response to the presence of a user or possibly other icons. The aim of emotional icons is to make the use of the data world a more interactive and dynamic experience. Emotional icons may respond to the presence or proximity of a user within the environment, their behaviour possibly being dependent on the profile of the user or their current activities and interests. Icons could possibly advance towards or retreat from the user, grow, shrink, animate or change their appearance all dependent on the relevance or importance of the data they represent to the current user. Icons could also respond to the proximity of other icons, those

representing information of a similar nature may move together whereas dissimilar icons may move further apart. Emotional icons could provide a big step towards creating a 'living' data environment.

2.10. Self organising graphs

Self organising graphs typically refer to a technique used in automatically laying out graphs. Conventional layout techniques involve a function or routine which attempts to perform a suitable layout on a given graph while attempting to satisfy a number of aesthetic criteria or heuristics. Self organising graphs allow the graph itself to perform the layout by modelling it as an initially unstable physical system and allowing the system to settle into a stable equilibrium. The use of computers removes the need for the graph behaviour to accurately model an actual physical system, this allows the implementation of models geared towards different criteria such as efficiency, speed, accuracy and aesthetics.

The first application of this method to graph layout was by Eades [Eades84] and was named the *spring-embedder*. Eades' work evolved from a VLSI technique termed *force-directed placement* [Quinn79] and has since seen a large number of improvements, modifications and similar algorithms. Graph layouts produced using such methods are often very satisfactory and aesthetically pleasing. Examples of the application of a force-directed placement algorithm to a large number of graphs can be found in [Fruchterman91].

Eades' system was modelled around a network of rings and springs. The vertices of the graph represented the rings and the edges represented the springs. The springs modelled did not obey Hooke's law as you would expect from a physical spring. When extended past their natural length they exerted an attractive force proportional to the length of extension. When compressed past their natural length they exerted a repulsive force, again proportional to the length of the compression. This network begins in an unstable or high-energy state and over a number of iterations the forces within the system will attempt to reach an equilibrium or minimum energy level. Once this more stable state is achieved the layout is complete.

This technique is often used in the display of large hyperstructures which are highly connected graphs of interacting components. Examples of such structures are program call graphs, control-flow graphs and hypertext document relationships. The technique is easily expanded into three dimensional space producing very interesting and often optimal graph structures. Examples of the practical application of such algorithms in 3D information visualisation can be found in [Chalmers95, Hendley95a, Kings95].

2.11. Spatial arrangement of data

A key problem in creating information terrains, for example PITs (below) is in creating a useful mapping from the data itself to a corresponding representation and location within the virtual environment. The requirement of this process is to create a spatial configuration from which the properties of data items within the information terrain and the relationships between them can be readily interpreted simply from their position and presentation. Within the research of Populated Information Terrains (PITs), the following four approaches were investigated [Colebourne94, Benford94a, Benford94b].

2.11.1. Benediktine Cyberspace

As discussed above, Benediktine Cyberspace is created by mapping data attributes onto the corresponding 3D object's intrinsic and extrinsic dimensions. This method would involve generating a schema for the database to be visualised in order to determine which data attributes are mapped to the intrinsic and extrinsic dimensions of the visual representation. An example of such an approach is Q-PIT which is described in section 3.3.1.

2.11.2. Statistical clustering and proximity measures

Statistical methods can be used to analyse database contents in an attempt to group items according to some measure of their semantic closeness. For example the contents of a document store could be grouped corresponding to matching keywords. Analysis performed on these information stores typically result in a number of 'scores' for documents which can then be used to create a suitable mapping into Benediktine space. The closer objects are semantically then the closer they will be within the data environment. Systems which adopt this approach include VIBE [Olsen93] and BEAD [Chalmers92], though the original idea of VIBE has been developed further and extended into three dimensions to produce VR-VIBE [Benford95].

2.11.3. Hyper-structures

Hyper-structures are created from information stores consisting of a number of data objects with any number or arrangement of explicit relationships between them. An example of a hyper-structure could be formed from the document structure within hypertext systems such as the World Wide Web. Another example is the dependency graph formed from components within a software system, such as functions. These structures are typically viewed using 3D extensions to standard graph drawing algorithms, or by more inherently 3D visualisations such as fish-eye views, cone trees or perspective walls.

2.11.4. Human centred approaches

This approach is not suitable for automatic generation of visualisations and relies on the user or system designer to create appropriate representations of the data to be visualised. This method lends itself well to the notion of abstract real-world metaphors such as cities, buildings and rooms within which the user navigates the data through more familiar surroundings. The problem with this approach is that the generation of such metaphors is time-consuming and difficult. Creating an abstract environment such as this which matches both the user's conception of these real-world surroundings *and* appropriate structure and representation of the data being visualised is not an easy task.

2.12. The information cube

The information cube is a technique developed by Rekimoto and Green [Rekimoto93] to visualise hierarchical information using nested translucent cubes. The information cube is loosely based upon the 2D tree-map visualisations of Johnson and Shneiderman [Johnson91, Shneiderman91]. Tree maps visualise hierarchical information in a rectangular 2D display using a space-filling approach to maximise display usage. The available display space is partitioned into a number of rectangular bounding boxes which represent the tree structure. Parent items are subdivided into

boxes which represent their children, and so on. Facilities are provided for the user to interact with the visualisation and specify the presentation of both structural and content information, for example varying the 'depth' of information or it's colour. Information within the display can be given a degree of interest or measure of importance which will result in it being allocated a greater proportion of available display space.

The information cube technique effectively extends 2D tree maps into 3D. Hierarchical information is presented as nested translucent cubes, with the level of transparency varied to control the 'depth' of the visualisation and hence the amount of information presented. Transparency and shading is the main technique used to control the information content of the cube visualisations. This transparency allows the user to view the contents of the cubes and their children, while hiding inner information gradually. Without this reduction in presented information the visualisation would become too complex to understand. A title is attached to the surface of each cube and leaf nodes, which contain no further nested cubes, are represented as 2D tiles with the title presented on the surface. Cubes may contain arbitrary information and are not restricted to containing only further cubes, other 3D visualisations or information may also be presented within.

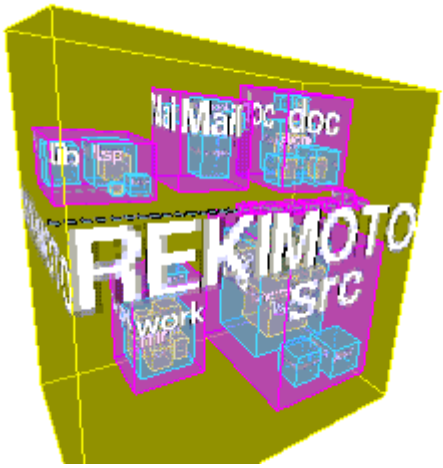


Figure 3. An information cube visualisation.
Image obtained from Jun Rekimoto's web page, Sony Computer Science Laboratory Inc.
http://www.csl.sony.co.jp/person/rekimoto/cube.html

3. Research visualisation systems

This section describes a number of information visualisation systems which make use of the techniques described in the previous section. The majority of these systems are prototypes or concept demonstrators and as such their merits or operation can only be assessed from the associated publications. This section aims to serve as a summary or synopsis of the publications describing these systems.

3.1. Telephone exchange network visualisation

A concept demonstrator developed at British Telecom [Walker93] made use of virtual reality techniques to visualise a telephone exchange network. This virtual environment allowed engineers or managers to fly through the network and expand

upon detail as and when needed, thus allowing the smooth integration of a higher level overview and lower level detail. One advantage which a visualisation such as this has over other information visualisation systems as described below, is the inherent spatial organisation of the structure being visualised. In the case of a telephone exchange network, the actual network structure is a physically real and geographically distributed entity. Structures such as these provide a good basis for visualisations as they already have some concrete form and thus tend towards less vague abstractions.

The actual visualisation itself is based on the structure of BT's synchronous digital hierarchy comprising of three logical levels. The network is visualised as a hierarchical graph overlaid over a two dimensional representation of the United Kingdom mainland with the hierarchy structured vertically above it. Again, this visualisation benefits from the geographical nature of the lower level with each 'ring' in this level approximating the corresponding geographical position on the UK map. Each of the levels can be selectively displayed or removed and additional information can be gained by approaching the nodes of the network.

Faults or alarms within the network are highlighted and the user is able to move into a particular node to investigate the problem. Within a node the user is confronted by a number of *rooms*, each offering a variety of information regarding the status of the node. This information is displayed as *writing on the wall*, i.e. pie charts and network data is displayed on the walls of the rooms.

3.2. Database visualisations

3.2.1. AMAZE

AMAZE [Boyle93, Benford94] is a three-dimensional visual query language created at Aberdeen University specifically for the complex analysis of protein structural data. AMAZE has developed as the next logical step in a series of advances to the existing database interface. AMAZE allows users to visually construct a query within the 3D environment then explore the resulting data sets in the same environment. The user constructs a query by interacting with a three dimensional representation of the schema. The completed query is passed to the remote database for evaluation and the resulting data is displayed as a set of 3D objects which the user can investigate further for additional information.

The AMAZE interface uses a mixture 2D and 3D to provide an extension rather than a replacement for the more typical and often more appropriate 2D windowing interface. 3D environments used in the system are supported within separate windows and navigated with a conventional mouse.

A session using the AMAZE interface would begin with the user navigating the 3D schema then attaching queries to relevant entity classes and adding constraints by way of a dialogue box. The user can build up a more complex query by navigating to a number of interesting entity classes and placing queries on each. Results from the query are represented within the 'Result Maze' which the user can explore through the 3D interface. The result maze is constructed by grouping resulting instances of the same entity class into sets then colour coding these sets. Each instance within a set is represented as a single 3D cube except for the case in which there are a very large number of instances, these being represented as an abstract group object. Finally, to

view a particular instance the user selects that instance causing an appropriate external viewer to display the data.

3.2.2. WINONA

WINONA, **WIN**dows **O**bject **N**avigation **A**pplication, provides a three dimensional visualisation of both the structure and content of an object orientated database. WINONA primarily uses variations on the cone tree and perspective wall to display the database structure. Representation and manipulation of these 3D structures is provided by a standard 2D windowing system and a viewpoint control toolkit. WINONA allows the user to investigate and interact with the contents of the database through these two 3D visualisations, both of which show the same information only in a different perspective.

The hierarchical visualisation shows the relationships between classes in the databases, each class being represented by a 2D square plane perpendicular to the plane of the tree. Within each class the individual objects are represented as cubes, their size, colour and level of detail increasing when selected with the option to view the data they represent. The relationships between classes and data are shown as links between the corresponding objects. Relationships are highlighted further as objects are selected. The perspective wall shows the same information in a similar form with the class planes becoming sections of the wall.

3.3. Populated information terrains (PITs)

The concept of a Populated Information Terrain (PIT) aims to provide a useful database or information visualisation by taking key ideas from the fields of Computer Supported Co-operative Work (CSCW), Virtual Reality and Database technology. The definition of a PIT is taken as a virtual data space that may be inhabited by multiple users [Colebourne94, Benford94b]. The philosophy behind PITs is that they should support people to work co-operatively *within* data as opposed to merely *with* data. Possible applications of PITs can apply to any form of information store for example library catalogues, large hypertext systems such as the World Wide Web or even large software systems through an appropriate set of visual abstractions.

PITs were developed in response to a lacking in current information stores to provide sufficient support for co-operative work and information browsing. The lacking of support for information browsing is addressed by the use of Virtual Reality to create a rich data environment in which the user can navigate and explore freely. VR provides the capabilities for the user to become immersed within the data and to explore and interact with it while making full use of their perceptual and intuitive skills. Co-operative work is supported in PITs by populating the virtual data environment with users. This allows users of the PIT to be aware of the presence and actions of other users within the same PIT and the ability to communicate with them, irrespective of their physical location. This awareness of other users enables a true sharing of information. A user interested in a particular subject area may question other users browsing the same information or even ask directions to relevant data.

3.3.1. Q-PIT

Q-PIT [Benford94b] is a prototype implementation of a populated information terrain which follows the Benediktine approach. Q-PIT is implemented using the World ToolKit virtual reality library and is capable of processing and displaying a simple

database of named tuples. Q-PIT allows the display, querying and manipulation of the database within the three dimensional environment, in addition to allowing this environment to be shared by multiple users.

Within Q-PIT, attributes of the data to be visualised are mapped onto the extrinsic and intrinsic dimensions of the their 3D representations. The extrinsic dimensions used are the X, Y and Z co-ordinate axes, while intrinsic dimensions supported are shape, height and angular velocity. Users within Q-PIT are embodied as monoliths, i.e. tall, thin cuboids each allocated a different colour to represent their identity. Users are aware of the presence of other users within the same PIT and can obtain information on them or open a communication channel with them.

Manipulation of the underlying data in Q-PIT is provided via a number of methods. The user may either select a single data item via it's 3D representation or they may issue a query resulting in all matching data items being highlighted. The actual data in the tuple can be modified by selecting an object then using a textual interface to modify the fields. Once altered, changes to either intrinsic or extrinsic attributes are reflected immediately in the visualisation, the latter of which results in a smooth transition between positions.

3.3.2. BEAD

BEAD [Chalmers92] is a virtual world document visualisation system which makes use of the DIVE toolkit and Visualizer to produce a multi-participant environment. BEAD makes use of a variant of the simulated annealing technique to arrange document representations within 3D space to form an information terrain. The layout algorithm is driven with a document similarity metric and strives to produce a terrain in which closely related or similar documents are grouped physically close together. This metric is based on word co-occurance, i.e. a measure of the number of words which occur in both documents. Similar documents are attracted towards each other, whereas dissimilar documents will repel each other if they are too close together.

An important point of note in the development of BEAD was that the original system attempted to match the physical distances between documents as closely as possibly to the similarity metric. This was facilitated by allowing the document nodes to move freely within three dimensions, thus creating a ball or cloud-like shape. It was found that the visual complexity of such structures and the inherent problems of navigation within them were too great. Development of BEAD then took on a more two-dimensional aspect, with the document nodes being forced to form a flatter terrain, though still with some degree of vertical movement. The document nodes are then meshed together by polygonal shapes, the resulting visualisation resembling more an island with an undulating terrain.

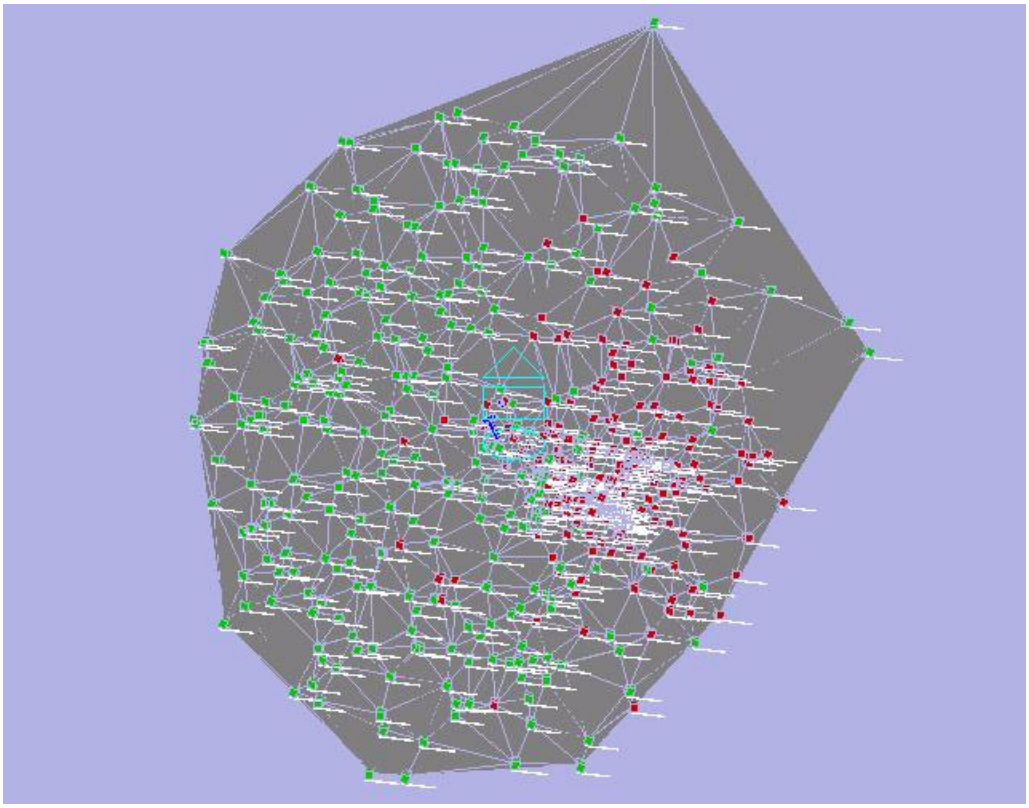


Figure 4. Screenshot of BEAD showing an overview (looking down) of a data landscape constructed from over 500 bibliographic references.
Image courtesy of Mathew Chalmers, Union Bank of Switzerland.
http://www.ubs.com/research/ubilab/Projects/hci/viz.html

The virtual environment or data landscape produced by BEAD can be populated by multiple users. Users are represented as T-shaped figures constructed of two cuboids, with a further two cuboids to represent eyes. This allows other users to see exactly how a user is orientated and in which direction they are looking. Users can interact with the data by either selecting specific objects or by performing searches which result in matching documents being highlighted. By browsing the overall data terrain certain features may become apparent such as dense areas and rough terrain, again these providing additional information on the similarity between documents. If a document is highlighted in a search then it is possible that documents close to it, but not highlighted may also be of some interest due to their high similarity. Inherent in the layout algorithm is a tendency for documents which are only slightly related to the main theme to be pushed towards the shores of the island. Such geographical features as the shoreline, dense clustering and rough areas allow the users to build up a notion of how to navigate the data and where to find data of most value.

3.3.3. VR-VIBE

VR-VIBE is a virtual reality extension to the original 2D VIBE system and enables the 3D visualisation of large document collections. The original VIBE system [Olsen93] was developed at the University of Pittsburgh and provided users with a 2D visualisation of a collection of documents. This visualisation gives users an overview of the documents they are interested in with respect to the whole collection. The visualisations in VIBE are constructed by defining a set of 'Points of Interest' (POIs) containing keywords which will be used as a basis for the query. A full text such is

performed on the document collection and each document is scored for relevance to each POI. The visualisation is constructed by distributing the POIs evenly on the 2D plane then placing the document icons between the POIs with respect to their relevance to each POI. For example, with two POIs A and B, a document may have a relevance score of 4 to POI A and a score of 3 to POI B. This document could then be placed 4/7 of the distance along the line joining A and B, thus showing the proportion of relevance attributed to each POI.

VR-VIBE exploits 3D visualisation techniques to tackle the problems of display crowding and also allow the possibility of larger numbers of POIs to be used. The document positioning technique used in VIBE is simply extended into three dimensions with VR-VIBE and is essentially the same. VR-VIBE offers two slightly different layout methods. One method is very similar to the VIBE layout and constrains the nodes to a 2D plane when organising their position between POIs. A vertical displacement is then introduced to each object to represent their degree of relevance as described below. The second method (figure 5) allows the objects to be positioned at any point in 3D space, within the confines of the POIs. The use of objects' intrinsic dimensions such as colour is exploited to indicate selection and also to emphasise relevance. The colour of a document object serves as an indication as to the level of relevance to POIs. This allows distinction between two documents which are both equally relevant to two POIs, however the actual scores may be different. For example a document with scores 3 and 3 would be coloured differently to one with scores 8 and 8. Shape is also used to distinguish between the documents, POIs and the users. These are represented as cuboids, octahedrons and 'blockies' (as in Q- PIT) respectively.

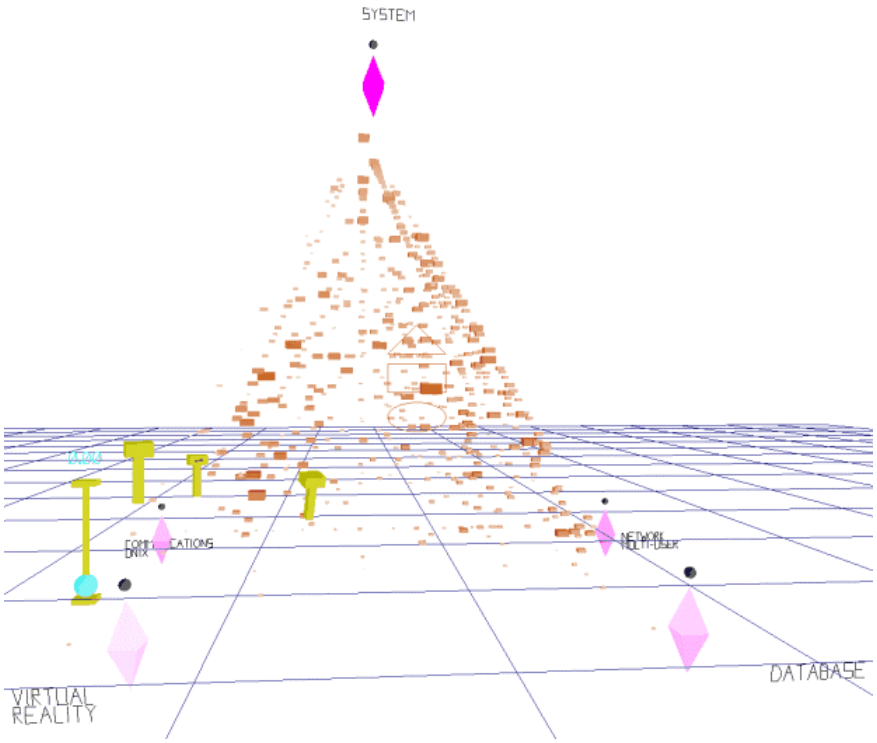


Figure 5. Screenshot of VR-VIBE showing a PIT containing 5 POIs and a number of users.
Image courtesy of Dave Snowdon, Nottingham University.
http://www.crg.cs.nott.ac.uk/crg/Research/pits/pits.html

VR-VIBE allows users to navigate freely through the information structure, select documents, perform queries, apply filtering or request additional information. A high degree of interaction is supported through the 3D interface with users able to add POIs, switch POIs on or off to try different configurations and even move POIs to see which documents get pulled after them. Users may also define a number of viewpoints within the visualisation and switch between them at will. This allows users to compare different views of the information and also aids in navigation by providing a set of reference points which can be quickly returned to. Finally, as with Q-PIT, VR-VIBE supports multiple users within the same visualisation, each aware of the presence and actions of each other.

3.3.4. LyberWorld

LyberWorld [Hemmje93, Hemmje94] is a prototype information retrieval user interface which aims to make full use of the advantages afforded by 3D graphics. LyberWorld concentrates on the visualisation of an abstract information space, *Fulltext*, and providing a user interface for the retrieval system *INQUERY*. The visualisations used in LyberWorld aim to provide an intuitive and natural interface to enable efficient searching and browsing within this abstract information space. The goal of LyberWorld is to provide a tight coupling between the graphical model of the information space (the visualisations) and the cognitive spatial model maintained by the user. If such a coupling exists then the visualisations should aid the user in exploring and querying the database, navigating and orientating themselves within the data, and judging the relevance and context of query results.

LyberWorld models the information space as a network of documents and terms. The relationships in this network are formed by a measure of the relevance between these document and term nodes. Two main visualisations are used within LyberWorld, the *NavigationCone* and the *RelevanceSphere*. NavigationCones provide a visualisation of the retrieval history, i.e. the extent to which the data space has been explored and the query paths travelled through the data. The RelevanceSphere provides a display of the retrieved documents and their relevance to each of the search terms in the query. Additionally the RelevanceSphere provides a visual indication of document clustering, highlighting documents of possibly closely related subject area.

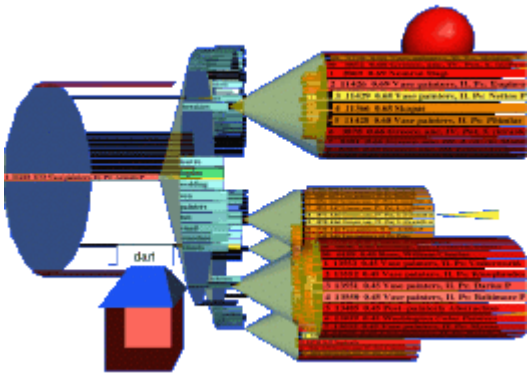


Figure 6. Screenshot of LyberWorld showing the *NavigationCones* visualisation.
Image courtesy of Matthias Hemmje,
German National Research Center for Computer Science (GMD).
http://www-cui.darmstadt.gmd.de:80/visit/Activities/Lyberworld/

The NavigationCones visualise the development of the retrieval dialogue and the extent to which the database has been explored. NavigationCones employ the Cone Tree visualisation [Robertson91] to present an expandable or collapsible display of the explored regions of the data space. One problem inherent with using the Cone Tree visualisation is that it can only visualise hierarchical data structures, whereas the complex network of documents and terms is often far from hierarchical. LyberWorld overcomes this problem by converting the document-term network into a hierarchical structure by introducing some redundancy. The network is transformed into a corresponding tree of alternating document and term nodes on each tier by duplicating nodes as necessary and introducing "hidden paths" between duplicate nodes.

Navigation of the information space is made by unfolding nodes of interest, alternating between document and term nodes. Nodes on the tree are colour coded to indicate their type (i.e. document or term) while duplicated nodes are additionally colour/shape coded to highlight a hidden path. Nodes on a particular tier of the tree are ordered corresponding to their relevance to their parent node. As with the cone trees of Robertson et al. [Robertson91], the NavigationCones rotate smoothly to position the current node of interest closest to the viewpoint. Similarly, as the tree is expanded or collapsed the radii of the various tiers are adjusted dynamically to maintain a compact structure while avoiding collisions.

LyberWorld incorporates the RelevanceSphere visualisation to aid in judging the relevance of items within the result space. The RelevanceSphere attempts to tackle the ambiguity problem associated with visualisations such as VIBE and VR-VIBE. This ambiguity occurs because a particular point in space within a VIBE visualisation can be attributed to a number of relevance ratios to different points of interest (POIs). This problem is described by Hemmje et al. [Hemmje94] and also by Benford et al. [Benford95]. The RelevanceSphere addresses this problem by transforming the network into a spatial visualisation which displays the relevance path lengths while still maintaining the numerical proportions. This approach allows path lengths to be easily compared with one another.

The RelevanceSphere visualisation operates on a similar principle to VIBE and VR-VIBE, that of attraction vectors and relevance paths. In the RelevanceSphere, all term nodes are positioned on the surface of a sphere and all document nodes are positioned within that sphere dependent on their relevance to each of the terms. The position of each document node is calculated by summing the attraction vectors between it and each of the term nodes. The resultant vector will determine it's position within the sphere, using the sphere centre as an origin. This visualisation allows the user to see both the relevance of documents with respect to the overall query but also their relative relevance to each of the terms. The former is implied in the distance between the document node and the centre or surface of the sphere. Documents closer to the surface are of a greater overall relevance. The relevance to each particular term is encoded in the document's distance from that term.

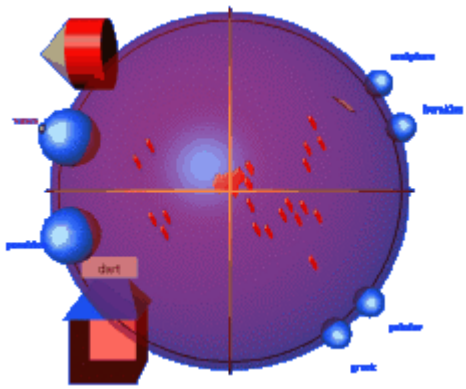


Figure 7. Screenshot of LyberWorld showing the *RelevanceSphere* visualisation.
Image courtesy of Matthias Hemmje,
German National Research Center for Computer Science (GMD).
http://www-cui.darmstadt.gmd.de:80/visit/Activities/Lyberworld/

The use of three dimensional space to contain the document and term nodes leads to both advantages and disadvantages in the case of the RelevanceSphere. The added display volume and thus increased spatial freedom reduces the probability for unclear positioning of document nodes, although it is still possible for ambiguity to arise. One problem encountered, typical in many 3D visualisations, is the reliance on the user's perception of depth. When looking at a static 2D image of a 3D structure it is often hard to judge depth, size and positioning. The RelevanceSphere deals with this problem by allowing the user to rotate the sphere freely and examine the visualisation from any angle. Additional features allowing the user to interact with the RelevanceSphere are provided. These facilities provide methods of investigating and manipulating the result space. User's can alter parameters such as the *document density*, *term attraction* and *scaling* of the visualisation [Hemmje94]. This interaction is an important contributor to increasing the user's understanding of the query results and the structure of the visualisation.

3.3.5. Vineta

Vineta is a prototype information visualisation system developed by Uwe Krohn [Krohn95, Krohn96]. Vineta allows the visualisation, browsing and querying of large bibliographic data without resorting to typing and revising keyword based queries. Similar to VR-VIBE, LyberWorld, and BEAD visualisations, Vineta presents documents and terms as graphical objects within a three dimensional space, the *navigation space* [Krohn96]. The positioning of these objects within that space encodes the semantic relevance between documents, terms and the user's interests. Vineta differs greatly from other visualisations in one distinct area, that is, the number of dimensions or terms it can present. Systems such as VR-VIBE are limited by the number of terms or points of interest (POIs) they can present simultaneously. Vineta applies techniques from multivariate analysis and numerical linear algebra for mapping documents and terms into the three-dimensional navigation space. This allows the visualisation of much larger term sets or POIs by providing a mapping from the multidimensional document space into the 3D navigation space.

Vineta is built upon the premise that navigation through an information space can be an effective means of retrieving information of interest. Krohn states that informational navigation is strongly connected with the human intuitive

comprehension of abstract facts by means of analogies with familiar concepts such as location or motion. Vineta uses spatial proximity to represent semantic similarity between objects (i.e. documents). As with previously described systems, the notion of similar documents is emphasised by placing them physically close together within the information space, whereas dissimilar documents are placed further apart.

Vineta has employed two main metaphors for producing visualisations, the galaxy metaphor and the landscape metaphor. The landscape metaphor has superseded the galaxy metaphor in the final implementation of Vineta, proving more intuitive and easier to comprehend.

The galaxy metaphor represents the navigation space as a galaxy of stars, documents are presented as fixed stars while terms are presented as 'shooting stars'. Semantic similarity is encoded in the galaxy metaphor by the proximity of the stars, i.e. similar documents and terms are grouped closely together.

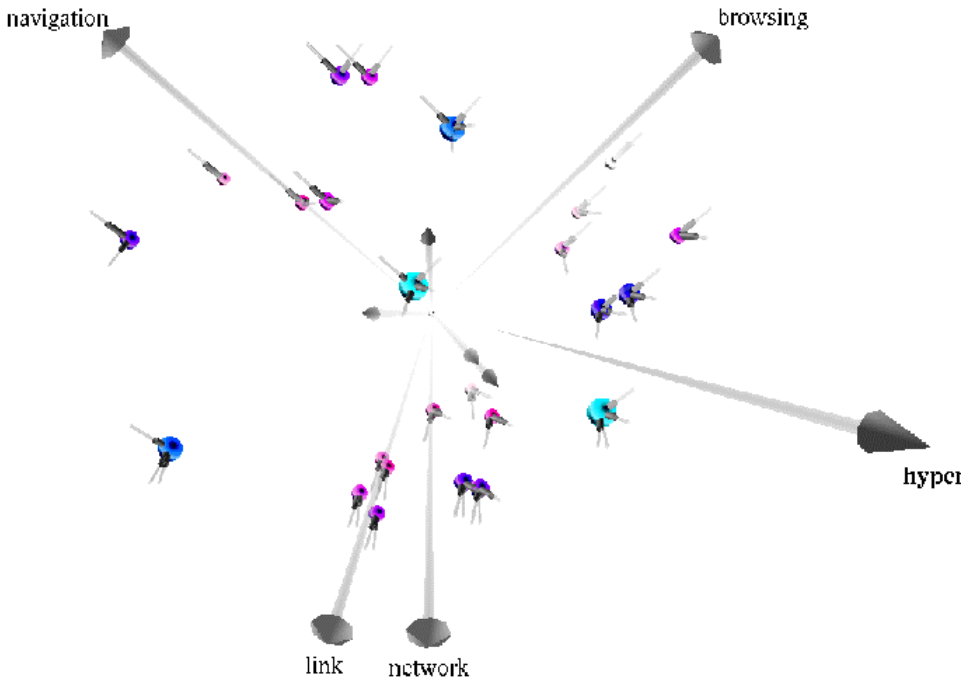


Figure 8. Screenshot of Vineta showing the galaxy visualisation.
Image courtesy of Uwe Krohn, University of Durham.

The landscape metaphor represents the navigation space as a flat, textured surface containing flowers with stems and petals. The textured surface of the landscape helps to give the viewer an indication of the depth or distance of the documents (flowers). The inclusion of this ground plane was encouraged by the study of ecological optics which emphasizes that perception of objects should never be considered apart from a textured ground surface [Krohn96]. Flowers nearer to the user's viewpoint are of more relevance to the initial query than flowers further from the viewpoint. The stems of the flowers aid the user by extrapolating the flower's position onto the ground plane. The direction and colour of petals on the flowers represent the search terms and their relevance to each document.

Figure 8 shows an image of the Vineta galaxy display. Terms are represented as arrows while documents are presented as spheres. Figure 9 shows an image of the

Vineta prototype, demonstrating the landscape metaphor. It can be seen from this image that the introduction of the ground plane helps greatly in determining the relative positions of the documents (i.e. the flowers). In both visualisations, documents closer to the user's viewpoint are of more relevance to their overall interests.

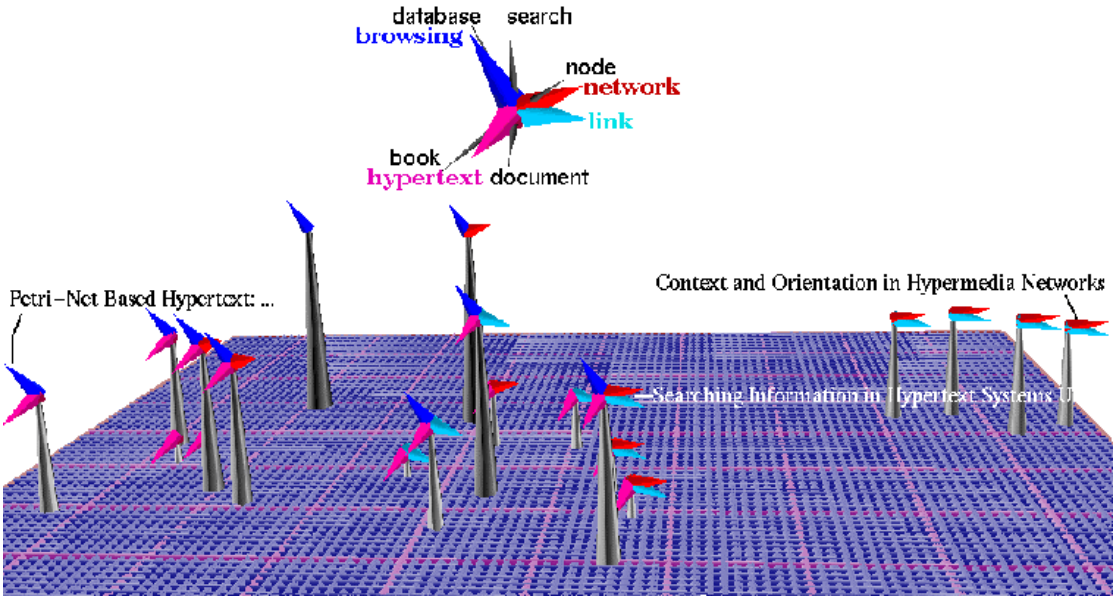


Figure 9. Screenshot of Vineta showing the landscape visualisation.
Image courtesy of Uwe Krohn, University of Durham.

3.4. Legibility enhancement

Research into legibility enhancement attempts to aid navigation through large virtual information spaces by applying principles from the more physical area of city planning. The legibility of an urban environment is judged by the ease with which inhabitants can develop a cognitive model of their surroundings over time, thus aiding their navigation through the environment. It has been argued that the legibility of urban environments can be improved greatly by careful design of certain key features. This section will concentrate on research performed at Nottingham University into the application of legibility features to 3D information visualisations [Ingram95a, Ingram95b].

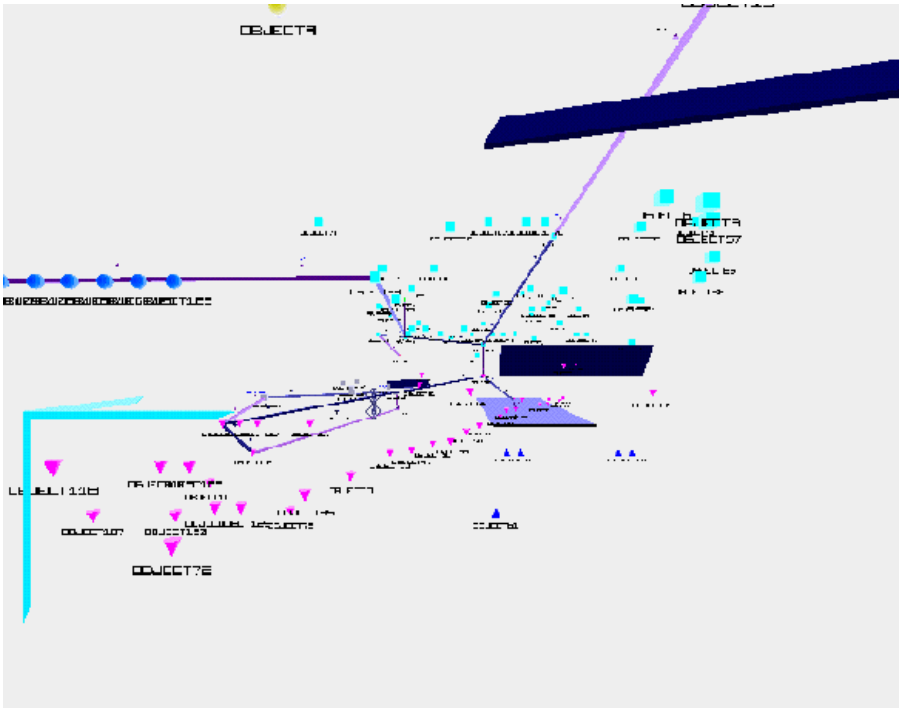


Figure 10. Screenshot of a Q-PIT visualisation after application of LEADS.
Image courtesy of Rob Ingram, Nottingham University.
http://www.crg.cs.nott.ac.uk/crg/Research/leads

The legibility of an environment relies on the ease with which an inhabitant can construct a cognitive model or internal representation of the environment. These cognitive models can be classified into two categories, linear and spatial. Linear models are based upon movement through the environment and observation of features on the route. Spatial models represent an overall model of some area of the environment which can be mentally manipulated and updated with information gained from linear models.

In "The Image of the City" [Lynch60], Lynch identifies five major elements which are used in constructing cognitive models of an urban environment. These elements were identified as a result of an experiment in which inhabitants of various cities were questioned on the city they resided in. Information was gained using various techniques such as interviews, written descriptions of routes through the city or by drawing maps. These five features are described here in addition to how they are applied to information visualisations by Ingram and Benford [Ingram95a]. The system developed at Nottingham is named LEADS (**LE**gibility for **A**bstract **D**ata **S**paces) and implements a number of algorithms to structure data according to these key features.

3.4.1. Districts

Districts are distinct areas within an environment which can be abstracted into a single entity. Such areas must usually contain some form of commonality or character in order for them to be considered as a whole. Districts can be identifiable in a real world sense, for example by the nature of the buildings within them. An example of two districts could be a residential district and a commercial district, both are distinct abstractions of the buildings within each district.

Identifying districts within an abstract data space is a process of determining similarity and dissimilarity between data items. This process is fairly complicated

when attempting to create a general method applicable to a wide range of data spaces. LEADS employs cluster analysis techniques to identify areas of similarity then groups all data items belonging to a particular area. The actual display of districts within the virtual environment is fairly arbitrary, the requirements being that the data items within a district appear similar and that districts appear sufficiently different. One possible representation could be to colour all data items within a district with a particular colour, each district then having a different colour. A similar approach could be taken using the shape of data representations.

3.4.2. Edges

Edges are features of an environment which provide distinctive borders to districts or linear divisions. Examples of edges are rivers and motorways though the latter may serve a dual nature, as an edge to pedestrians or as a path to motorists. As edges serve as boundary features then it makes sense to position them at intersections between districts within the abstract data space. Research done in developing the LEADS system identified three possible methods for defining appropriate edges, these will be briefly described here.

The first method, also the selected method due to complexity and efficiency reasons, is to identify the two nearest data items between districts. An edge can then be defined between these two points and orientated accordingly. This method is extremely primitive but also allows very fast identification of edges. The second method is to identify the hull or bounding region which completely encloses a district. This hull can then be rendered as the edge. Finally, the third method involves identifying the hull surrounding two districts then creating an edge by interpolating the points along adjoining edges of the districts. Unfortunately the latter two methods are computationally expensive.

3.4.3. Landmarks

Landmarks are static and easily recognisable features of the environment which can be used to give a sense of location and bearing. Within urban environments examples of landmarks could include distinctive buildings or structures. One pre-requisite of landmarks, both within the urban environment and the data space is that they remain relatively static in position. This is not a simple task when considering a possibly changing data environment, so it is reasonable to assume that the position of landmarks may vary slightly from time to time.

Again three methods were evaluated for inclusion in LEADS these will be described here briefly. The first method defines the landmarks at the centroid or geometric centre of districts. While simple to implement, this method is rejected as it does not cater for the size or density of districts. Landmarks could be easily obscured by dense regions of data and landmarks may possibly be more meaningful if placed external to districts.

The second method places landmarks at intersections between three or more districts by simply identifying the midpoint of the closest data items between adjacent districts. These landmarks would then give an indication to clusters of districts and would probably serve as a better navigation aid. Finally, the third method is a slight variant of the second in that the landmarks are placed by triangulation between the centroids of any groups of three adjacent districts. This method produces roughly the same number of landmarks as the second method, though will probably result in a

more stable positioning as it relies on the average position of data within a district and not just any one data item.

3.4.4. Nodes and paths

It is proposed by Ingram and Benford that paths should be composed of links between nodes which represent individual objects within the visualisation. The construction of these nodes and paths will rely on two stages. The first stage will assume no initial access to the data space and will attempt to define nodes and paths as data items which are likely to be accessed frequently. This will work on the basic assumption that more frequently accessed data will possibly be items at intersections between districts and also at the centre of districts. The former assumes that these items will be seen as the most similar items between two districts. The latter assumes that the centre items will be most typical of the information content of a district.

The second stage relies on the recording of usage information within the data space, such as which nodes are accessed and in which order they are accessed. This information can then be used to gradually construct new nodes and paths, with old and disused nodes and paths possibly fading over time. Metrics such as the frequency of access of a data item could be used to identify nodes, whereas the frequency with which two data items are accessed successively could be used to define paths. Finally, the aim is to produce two forms of path, major and minor, which will indicate the popularity of various routes.

3.5. Hyperstructure visualisation

As mentioned in 2.11.3 above, hyper-structures are formed by typically large information stores with an arbitrary number and configuration of explicit relationships or links between data items. Examples of such structures are extremely common within software visualisation, in particular dependency graphs. A more immediate and recognisable example is the graph formed from document stores in hypermedia systems where the documents and the links between them form the hyper- structure. The inherent complexity and arbitrary arrangement of such structures makes visualisation an extremely difficult task. Hyper-structures representing even relatively simple data sets can quickly clutter the visualisation and rapidly deteriorate usefulness.

Several visualisation systems have been developed to offer some form of visualisation of these complex structures, the majority of which opting to use self-organising graph layout techniques as described in 2.10 above. A small number of these systems are described here.

3.5.1. Narcissus and Hyperspace

Narcissus [Hendley95b] is an information visualisation system developed at the University of Birmingham for creating 3D visualisations of hyperstructures with a particular application towards software visualisation. One of the proposed applications of Narcissus is in visualising software structure by displaying the dependencies between the various components within a system. Hyperspace [Wood95] is an extension of the research into the Narcissus information visualisation system and concentrates on the visualisation of hypertext document stores, specifically HTML WWW pages readable using NCSA Mosaic. The system makes use of the Mosaic WWW browser API for data collection, i.e. fetching pages from remote

locations. The Mosaic browser is also used to display pages selected from within the visualisation.

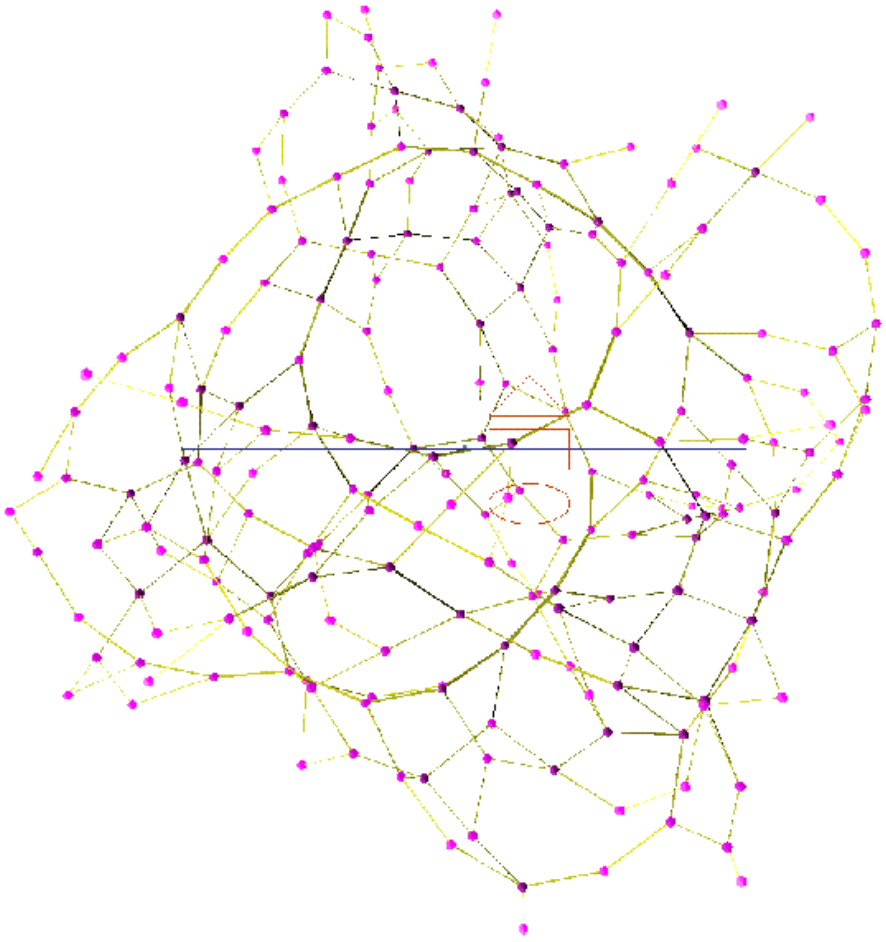


Figure 11. Example of a hyperstructure.
Image courtesy of Dave Snowdon, Nottingham University.
http://www.crg.cs.nott.ac.uk/crg/Research/pits/pits.html

The visualisations in both Hyperspace and Narcissus are created as wireframe 3D graphs with the pages represented as spherical nodes and the links as directed edges. Layout of the graph is performed using a self-organising technique as described in 2.10 above, with new nodes introduced to the structure being placed randomly. Problems with this form of visualisation are that the structure and appearance of the model produced by the graph may change greatly when a new node or link is introduced. This is compounded by the fact that visualisations of the same structure may be inconsistent between runs. This effectively destroys the user's cognitive model of the structure thus requiring frequent re-orientation as the graph is updated.

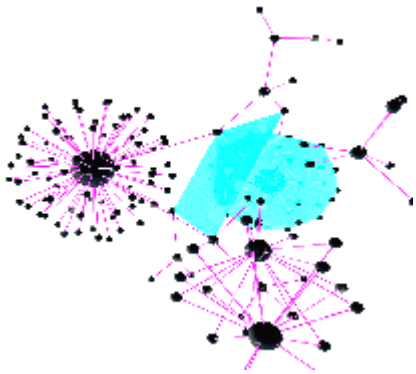


Figure 12. Screenshot of ray-traced output from Narcissus.
Image obtained from Birmingham University WWW presentation,
http://www.cs.bham.ac.uk/~nsd/Research/

3.5.2. SHriMP Views

SHriMP (**S**imple **H**ie**r**archi**c**al **M**ulti- **P**erspective views) [Storey95] is a tool for visualising large graphs and hyperstructures and is geared towards such graphs within software systems. SHriMP is incorporated within the Rigi reverse engineering system and aims to provide an overview of software structure allowing both detailed views while still maintaining a notion of positioning or context within the graph structure. SHriMP employs both nested graph and fisheye (see 2.3 above) techniques to provide both abstraction and focusing of detail as necessary.

Nested graphs [Harel88] are a technique used to abstract information from complex graphs in an attempt to clarify their structure yet also enable the detail to be viewed when necessary. Nested graphs are based on the notion that nodes and arcs within the graph may be either atomic or composite. Composite nodes are a single node abstraction of any number of child nodes, similarly composite arcs may represent any number of child arcs. The use of such a technique allows graph structures, particularly hierarchies to be simplified greatly, allowing the user to select additional detail on areas of interest.

3.5.3. SemNet

SemNet [Fairchild88] is a research prototype developed to advance understanding of the complex relationships and structures of large, arbitrary knowledge bases (Figure 13). The aim of SemNet was not to provide a solution for this general problem, but rather to explore the application of 3D visualisation techniques and to investigate the specifics of the problem. The main problem addressed by SemNet is how to present large knowledge bases to enable improved or more efficient comprehension. Fairchild and Poltrock hypothesise that for comprehension of a knowledge base, a user must recognise:

- the identity and meaning of individual elements;
- the relative position of elements within a hierarchical context;
- the explicit relationships between elements.

Knowledge bases represent information about relationships between objects or records. Fairchild and Poltrock believe that graphical representations offer an effective way of communicating such relationships. This coupled with the high level

of visual processing and spatial reasoning capable in the human brain, makes 3D representations a good candidate for such a task.

Fairchild and Poltrock are quick to point out that simply using a graphical representation does not immediately solve the problems inherent with exploring, manipulating, understanding and modifying large knowledge bases. Very large knowledge bases become very large directed graphs! This highlights an important need for reducing the visual complexity of the graphical representations, a need which is also addressed in SemNet.

Graph layout and information hiding within SemNet is debated by Fairchild and Poltrock. A number of methods are described with their advantages and disadvantages noted. The authors agree that the positioning of knowledge elements within the visualisation can greatly affect comprehension of the knowledge base structure. Layout of the visualisation in SemNet can be performed using one of two methods:

- the application providing the knowledge base assigns positions to the elements using semantic information which is not available to SemNet;
- alternatively, SemNet can perform layout using a number of heuristics designed to bring highly related elements closer together while moving unrelated elements further apart. As the semantic meaning of the elements is not available to SemNet, the layout algorithm uses the number of interconnections between nodes as a measure of their relatedness. Thus, highly interconnected nodes are placed much closer together than unconnected nodes.

Information hiding in SemNet is provided both by the inherent perspective based reduction in detail due to distancing in the 3D display, and also by application of additional fish-eye techniques. The former method is apparent if proximity-based positioning has been used, thus objects which are unrelated to the current interest will be further from the viewpoint, hence reduced in detail and size. This provides a fish-eye view based purely on three-dimensional perspective. This was combined with a cluster based fish-eye view which attempted to reduce the complexity of the graph while still maintaining the context of the overall structure. This technique assigns elements within similar areas of 3D space to clusters, with neighbouring clusters being assigned to higher level clusters. The technique is similar to the nested graphs of Harel [Harel88]. Using these clusters, only knowledge elements close the viewpoint are displayed individually, at further distances only the clusters are visible.

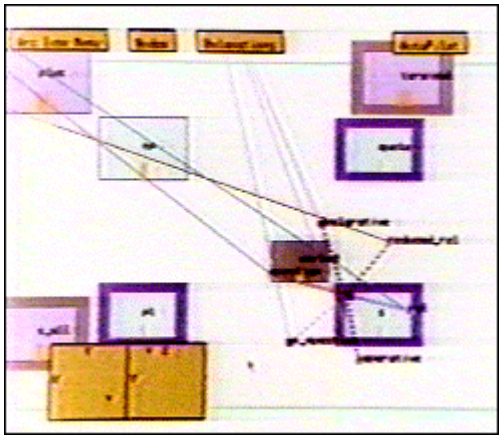


Figure 13. SemNet visualisation

Image courtesy of Kim Fairchild,
Institute of Systems Science, National University of Singapore.
http://panda.iss.nus.sg:8000/kids/fair/webdocs/viz/viz-1.html

3.5.4. GraphVisualizer3D

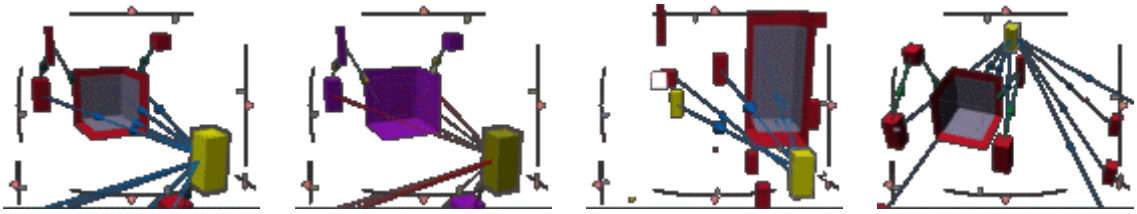
GraphVisualizer3D (GV3D) [Ware93, GV3D95] is a research project under development at the University of New Brunswick (Figure 14, a-d). The aim of GV3D is to utilise 3D visualisations and virtual environment technology to provide a more readable and comprehensible graph visualisation. The project is aimed at displaying predominantly directed graphs, though the focus of the development has been to visualise software structures. The structures in this case are component relationships within object oriented C++ programs, examples cited are: software module dependencies; software usage; and inheritance relations. Ware et al. [Ware93] describe the design issues and architecture of GV3D and include details of empirical evaluations performed on the system, which include an investigation into various interfaces and interaction techniques.

Ware et al. investigate the use of a number of different interaction techniques both for inclusion in the GV3D project [Ware93] and also as a separate study [Ware94]. The study evaluated the occurrence of errors in a graph comprehension task in which the subjects were restricted to using a number of different interaction techniques. The results of the study highlight an important decrease in errors, thus an increase in graph cognition, when a combination of stereoscopic display and viewpoint manipulation techniques are used. The study also confirmed previous research which indicated that the use of viewpoint manipulation alone was more effective than a stereoscopic display alone. These results indicate the importance of the virtual environment interface in producing useable and useful visualisations.

GV3D draws on a number of ideas from SemNet and is hinged upon the same concepts, however GV3D aims to improve upon SemNet in a number of ways. One area of divergence is a shift from the predominantly automatic node placement of SemNet to the more manual oriented placement of GV3D. Ware et. al believe that the most important heuristic for graph layout is semantic clustering, that is grouping nodes with respect to their semantic meaning as opposed to their position within the graph structure. The authors state that it is unreasonable to expect current layout algorithms to achieve more than a first approximation to the layout of software entities in space. The reason being that good layout is based greatly upon the high level semantics of the nodes, this cannot be formalised or automated at present. The graph layout used in GV3D is based upon elementary layout algorithms but supplemented by extensive manipulation and interaction techniques.

Similar to SemNet, GV3D also includes the notion of composite nodes and arcs [Harel88]. This technique allows reduction of the visual complexity of a graph structure by representing similar nodes and arcs as a single composite node or arc (the measure of similarity is arbitrary). These composite nodes can then be collapsed or expanded (opened or closed) as needed, depending on the focus of the viewer. GV3D copes with composite nodes by performing an identical layout on the sub-graph enclosed in the composite node. The composite node is then scaled to completely enclose the sub-graph, for this reason the node size cannot be used to display attributes. Opening or expanding a node results in the contents being displayed.

Input to GV3D is provided via a proprietary Graph Description Language (GDL). This language allows the permanent storage of graph structure, visual representation and layout information. Graph structures to be visualised, such as the C++ software components, must first be converted to the GDL format. Interaction with the GV3D visualisations is provided by two main methods, both of which use a standard high-resolution monitor for display. The viewer can use the monitor alone, making use of 3D Widgets to control position and velocity along each of the co-ordinate axes. Alternatively the viewer can also use a head tracked stereoscopic display. This provides the viewer with a stereo display and also allows head movement to manipulate the viewpoint thus providing essential perceptive cues which greatly aid depth perception and object recognition.



(a) (b) (c) (d)

Figure 14 (a-d). Various views of a network using GraphVisualiser3D
Images courtesy of Glenn Franck, University of New Brunswick.
http://www.omg.unb.ca/hci/projects/hci-gv3D.html

3.6. Information workspace

The *information workspace* or *information visualizer* is a concept demonstrator designed at the Xerox Palo Alto Research Centre (PARC) and is aimed at providing a more cost efficient workplace with respect to information access. The information visualizer is targeted as being the successor to the desktop metaphor and takes full advantage of 3D graphics and modern computing power to produce an intuitive and effective interface. Research at PARC has investigated the cost structure of information retrieval [Card91] within the workplace and oriented the information visualizer towards maximising effective organisation of information and minimising retrieval cost.

Research into the information visualizer has lead to a number of new representations and metaphors for data being developed. These new visualisations include cone trees, cam trees, perspective walls and 3D rooms all described above. The information visualizer is based primarily around the 3D rooms metaphor with rooms containing visualisations appropriate to the data contained within them. The information visualizer incorporates all of the features of rooms as described in 2.8 above, including the ability to display an overview of all rooms in use. The user is also permitted to manipulate the contents of the rooms from the overview, though in considerably less detail.

Navigation through the rooms is provided by the users viewpoint being embodied in a virtual human allowing independently controlled motion and viewpoint orientation, similar to a human walking while looking around. Interaction with objects within rooms is facilitated by either selecting an object to examine, or by giving a *gesture* indicating the operation to perform. Objects or the viewpoint may be moved towards

one another by selecting an object of interest and moving towards it or moving it towards you. The speed of movement is logarithmic with separation ensuring maximal control over positioning and also that the object and viewpoint do not collide. Many items within the information visualizer respond to gestures. Gestures are performed using rapid movement of the mouse in distinct paths, for example flicking the mouse in one direction or tracing a check mark or cross. Gestures allow an easy method of performing simple commands without releasing the mouse or requiring banks of function icons.

3.7. Other systems

3.7.1. VOGUE

VOGUE is a visualisation system developed at Koike Labs, which provides a framework for creating 3D software visualisations. VOGUE concentrates on the concept of integrating a number of 2D views to create a more powerful 3D visualisation. This system has been applied in a number of different visualisation scenarios:

- as a software visualisation of parallel Linda programs (VisuaLinda - described below);
- as a version control and module management visualisation;
- as a C++ class library browser.

VOGUE is used in creating a 3D visualisation which integrates version control and module management (Figure 15(a)). The visualisation creates a two dimensional tree from the module dependencies at each revision. This set of 2D trees are then placed in series along the Z axis of the display space, with the most recent revision placed at the front. Relationships between source code modules are visible both between different versions of the same module through time, and as compilation dependencies between modules for each revision.

The 3D class library browser implemented using VOGUE integrates two sets of information concerned with class inheritance hierarchies (Figure 15(b)). The first set concerns the actual inheritance hierarchy and depicts which classes are derived from other classes. The second set of information shows which methods belong to particular classes. These two sets are integrated into a single 3D visualisation to provide immediate indication of which method is actually executed when called from any particular class. By shifting viewpoint orientation the programmer can obtain a display of either data set, or view both sets and any dependencies between them. This clarifies the problem by reducing the need to mentally integrate both sets of information.

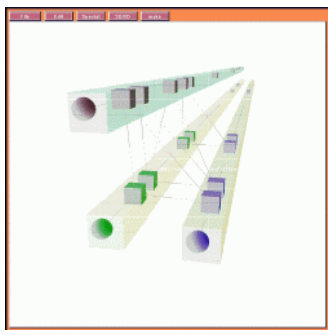


Figure 15(a). VOGUE version control management

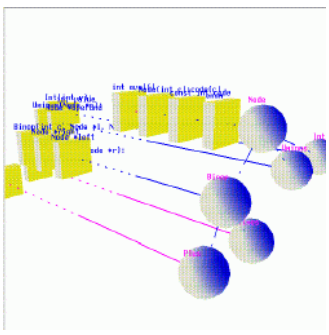


Figure 15(b). VOGUE C++ class browser

Images courtesy of Hideki Koike,
Koike Labs, University of Electro-Communications, Tokyo.
http://www.vogue.is.uec.ac.jp/

3.7.2. VisuaLinda

VisuaLinda is a program visualisation system developed at Koike Labs which uses 3D visualisations to clarify the operation of parallel programs [Koike95]. VisuaLinda is built upon the VOGUE visualisation system, also developed at Koike Labs.

VisuaLinda makes use of 3D graphics to visualise the execution of parallel processes (Figure 16). This allows the programmers to view both the relationships between processes and a time flow of the processes simultaneously.

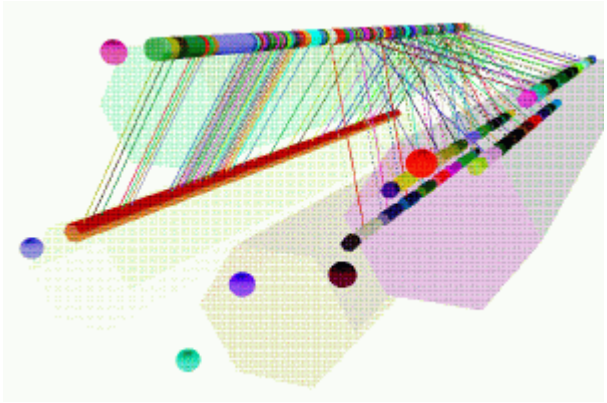


Figure 16. VisuaLinda implemented using VOGUE
Image courtesy of Hideki Koike,
Koike Labs, University of Electro-Communications, Tokyo.
http://www.vogue.is.uec.ac.jp/

3.7.3. VizNet

VizNet is an information visualisation system developed by Fairchild et al [Fairchild93] to enable the display of multimedia object collections. Two visualisations produced by VizNet are the cone tree (section 2.6) and the sphere visualisation (section 2.7), both make use of FishEye techniques to present the multimedia objects. VizNet supports five detail levels for displaying object icons. These range from not displaying anything, through increasing levels of detail, to displaying the full contents of the object that the icon represents. Size and resolution (or geometrical complexity in the case of 3D objects) is used to signify the level of interest or importance which a particular object possesses with respect to the users' current focus.

The type of graphical object to be attached to an icon is generated automatically using the *AutoIcon* technique also described by Fairchild et al. This assignment is based upon the nature and attributes of the object to represent. In the VizNet system three-dimensional objects are modelled by graphical 3D shapes which take the form of the object; images are mapped onto 3D spatial images; and text is represented as a text object. The level of detail affects each type of object differently, for example, the complexity (number of polygons) of the 3D objects or the resolution of the images is varied.

Figure 2 above shows an example visualisation produced using VizNet. This shows the sphere representation with the object of interest displayed in the centre and related objects fanning out from it.

3.7.4. FSN

FSN, pronounced 'Fusion', is file system navigator developed by Silicon Graphics as a fun and freely available utility to demonstrate their 3D graphics hardware. One claim to fame is that it starred in the extremely popular film *Jurassic Park*, making it possibly the most widely viewed piece of software. FSN makes use of the landscape metaphor to visualise the structure of a UNIX filestore and was originally intended as an investigation into information landscape navigation. The essentially hierarchical information is presented as a tree of nodes and paths on the surface of the landscape or ground plane. The root of the tree begins closest to the user with the branches receding into the distance, away from the user's viewpoint.

Each directory within the tree is represented as a pedestal, the height of this pedestal represents the combined size of the files contained within that directory. These directories are connected by paths or links which can be traversed to manoeuvre within the file structure. Files within each directory are represented by boxes, placed in a grid, on top of the directory pedestal. Each box is adorned with a graphical image mapped onto the top surface which represents the type of the file. The height of these file boxes represents their size, whereas the colour of the boxes represents the age of the file. The use of this information encoding, coupled with the landscape metaphor means it is easy to identify prominent features such as large files or directories. These features can then act as landmarks or points of reference within the visualisation.

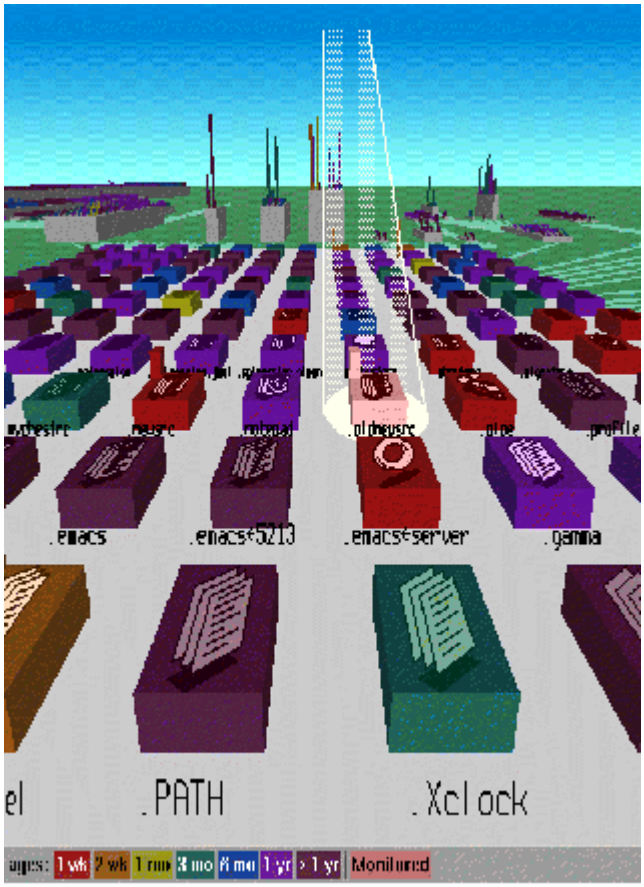


Figure 17. FSN visualisation of a UNIX file store

Image courtesy of Joel Tesler, Silicon Graphics.
http://www.sgi.com/Products/Mineset/products/vtools.html#TreeVisualizer

FSN is a fully fledged file manager it that it not only allows the visualisation of the file structure, but also allows manipulation of it's contents. Various operations may be performed on the files such as copying or moving them. Additionally, certain files may be edited or viewed using appropriate external applications depending on the file type. Selection of files is performed by an effective 'spot-light' facility which highlights the selected file, as shown in Figure 17.

4. Conclusions

This report has presented a number of information visualisation techniques and research systems. It is clear from the variety of these techniques that the problem of general information visualisation is very hard. One problem of considerable worth is in the generation of abstract representations which both hide the underlying complexity while offering a simple and recognisable graphical representation, i.e. a meaningful abstraction. This problem will prove a great hurdle in the acceptance and widespread use of information visualisation systems (in addition to the increased demand on computer performance). This can be circumvented to some extent by producing more specialised visualisations, meaningful only to their intended information domain. This helps by limiting the variety of information to be visualised, though the problem still exists of creating suitable abstractions for this information.

Information visualisation techniques should play a great role in the field of software visualisation. Software systems are not physical entities as such, they have no physical form and exist only as information; they are information artefacts. This highlights the problem mentioned above, that of creating meaningful visual representations or abstractions. Creating abstractions for software components and their dependencies will not be an easy task. The sheer size and complexity of the abstract structures created within software places them beyond the cognitive ability of a single human. The goal of software visualisation will thus be to allow users to browse and navigate these software structures freely and to identify and extract the information they require quickly. Co-operative work and interaction should also provide a major benefit to such systems, allowing teams of engineers to work on a particular software system and be aware of the actions of their colleagues.

In any case, it is clear that research into information visualisation will continue to be of great interest in the future. The need for techniques and systems developing from this research will increase as the amount of available on-line information grows increasingly rapidly. It is currently often hard enough to find files of relevance in one’s own disk space, buried within the myriad of directories, old documents and temporary files. Without regular cleaning and restructuring this problem will worsen over time, unfortunately globally available information does not often undergo such maintenance. Sometimes when it does, this maintenance itself causes problems due to users of the information not being notified of changes. This problem will only worsen, the task of maintaining or regulating this information is immense and possibly not feasible. Research into techniques to help cope with this problem rather than cure it will thus be of great relevance, information visualisation is one such technique.

References

[Benedikt91]
M. Benedikt,
Cyberspace: Some Proposals,
In *Cyberspace: First Steps*, MIT Press, pp. 273-302, 1991.

[Benford94a]
S. Benford et al,
Experience of using 3D graphics in database visualisation,
Computing Department, Lancaster University, October 1994.
ftp://ftp.comp.lancs.ac.uk/pub/vds/

[Benford94b]
S. Benford and **J. Mariani**,
Populated Information Terrains: Virtual Environments for Sharing Data,
Research report CSCW/4/1994, Centre for Research in CSCW,
Lancaster University, 1994.
ftp://ftp.comp.lancs.ac.uk/pub/reports/1994/CSCW.4.94.ps.Z

[Benford95]
Steve Benford, **Dave Snowdon**, **Chris Greenhalgh**, **Rob Ingram**, **Ian Knox** and **Chris Brown**,
VR-VIBE: A Virtual Environment for Co-operative Information Retrieval,
Eurographics'95, 30th August - 1st September, Maastricht,
The Netherlands, pp 349-360

[Boyle93]
J. Boyle, **S. Leishman**, **J. Fothergill** and **P. Gray**,
Development of a Visual Query Language,
Aberdeen University, 1993.
http://www.biochem.abdn.ac.uk/~john/vlq/vlq.html

[Card87]
S.K. Card and **A.H. Henderson Jr.**,
A multiple virtual workspace interface to support user task switching,
Proceedings of the CHI+GI 1987 (Toronto, April 5-7). ACM,
New York, pp. 53-59, 1987.

[Card91]
S.K. Card, **G.G. Robertson** and **J.D. Mackinlay**,
The Information Visualizer: An Information Workspace,
Xerox Palo Alto Research Center, Palo Alto, California 94304, 1991.

[Chalmers92]
M. Chalmers and **P. Chitson**,
Bead: Explorations in Information Visualisation,
Proceedings of SIGIR'92, published as a special issue of SIGIR forum,
ACM Press, pp. 330-337, June 1992.

[Chalmers95]
M. Chalmers,
Design perspectives in visualising complex information,
In Proceedings of IFIP 3rd Visual Databases Conference,
Lausanne, Switerland, March 1995.
http://www.ubs.com/research/ubilab/Publications/Cha95.html

[Clarkson91]
M.A. Clarkson,
An Easier Interface,
BYTE, February 1991.

[Colebourne94]
A. Colebourne et Al,
Populated Information Terrains: supporting the cooperative browsing of on-line information,
Research report CSCW/13/1994, Centre for Research in CSCW, Lancaster University, 1994.
ftp://ftp.comp.lancs.ac.uk/pub/reports/1994/CSCW.13.94.ps.Z

[Eades84]
P. Eades,
A hueristic for graph drawing,
Congressus Numerantium, 42, pp. 149-160, 1984.

[Fairchild88]
K.M. Fairchild, **S.E. Poltrock** and **G.W. Furnas**,
Semnet: Three-dimensional Graphic Representations of Large Knowledge Bases,
in "Cognitive Science and Its Applications for Human-Computer Interaction",
Fairlawn, NJ: Lawrence Erlbaum Associates, 1988
http://panda.iss.nus.sg:8000/kids/fair/webdocs/semnet/semnet-1.html

[Fairchild93]
K.M. Fairchild,
Information Management Using Virtual Reality-Based Visualizations,
in "Virtual Reality: Applications and Explorations", Alan Wexelblat (ed.), Academic Press Professional,
Cambridge, MA, ISBN 0-12-745045-9, pp. 45-74, 1993
http://panda.iss.nus.sg:8000/kids/fair/webdocs/viz/viz-1.html

[Fruchterman91]
T.M.J. Fruchterman and E.M. Reingold,
Graph Drawing by Force-Directed Placement,
Software Practice and Experience, Vol 2, Part 11, November 1991.

[Furnas86]
G.W. Furnas,
Generalized fisheye views,
In Proceedings of the ACM SIGCHI '86 Conference on Human Factors in
Computing Systems, pp. 16 - 23, 1986.

[Gibson93]
W. Gibson,
Burning Chrome,
reprinted in *Burning Chrome*, HarperCollins Science Fiction & Fantasy, 1993,
ISBN : 0-586-07461-9.

[GV3D95]
The GraphVisualizer3D Project,
On-Line project overview, University of New Brunswick, 1995
http://www.omg.unb.ca/hci/projects/hci-gv3D.html

[Henderson86]
D.A. Henderson and **S.K. Card**,
Rooms: The use of multiple virtual workspaces to reduce
spatial contention in a window-based graphical user interface,
ACM Transactions on Graphics 5, 3 July, 1986.

[Harel88]
D. Harel,
On visual formalisms,
Communications of the ACM, 31(5), May 1988.

[Hemmje93]
M. Hemmje,
A 3D Based User Interface for Information Retrieval Systems,
In: Proceedings of IEEE Visualization '93,
Workshop on Database Issues for Data Visualization, San Jose, California, October 25-29, 1993
ftp://ftp.darmstadt.gmd.de/pub/VISIT/papers/hemmje/IEEEVIS93.ps

[Hemmje94]
M. Hemmje, **C. Kunkel** and **A. Willet**,
LyberWorld - A Visualization User Interface Supporting Fulltext Retrieval,
In: Proceedings of ACM SIGIR '94, Dublin, July 3-6, 1994
ftp://ftp.darmstadt.gmd.de/pub/VISIT/papers/hemmje/SIGIR94.ps

[Hendley95a]
R. Hendley and **N. Drew**,
Visualisation of complex systems,
School of Computer Science, University of Birmingham, 1995.
http://www.cs.bham.ac.uk/~nsd/Research/Papers/HCI95/hci95.html

[Hendley95b]
R.J. Hendley, **N.S. Drew**, **A.M. Wood** and **R. Beale**,
Narcissus: Visualising Information,
University of Birmingham, 1995.
ftp://ftp.cs.bham.ac.uk/pub/authors/R.J.Hendley/ieeeviz.ps.Z

[Ingram95a]
R. Ingram and **S. Benford**,
Legibility Enhancement for Information Visualisation,
Proceedings of Visualization 1995, Atlanta, Georgia,
October 30 - November 3, 1995
http://www.crg.cs.nott.ac.uk/crg/Research/leads/viz95.html

[Ingram95b]
R.J. Ingram,
Legibility Enhancement For Information Visualisation,
PhD Thesis, Department of Computer Science,
University of Nottingham, September 1995

[Johnson91]
B. Johnson and **B. Shneiderman**,
Treemaps: A space-filling approach to the visualization of hierarchical information structures,
In Proceedings of the 2nd International IEEE Visualization Conference,
San Diego, pp. 284-291, October 1994
http://www.cs.umd.edu/Document/UMCP-CSD:CS-TR-2657

[Kings95]
N.J. Kings,
Software Visualisation,
Report for the Corporate Research Programme, British Telecommunications, 1995.

[Krohn96a]
U. Krohn,
VINETA: Navigation Through Virtual Information Spaces,
in "AVI: Advanced Visual Interfaces",
Ed. T. Cartaci, Gubbio, Italy, ACM, 1996

[Krohn96b]
U. Krohn,
Visualization for Retrieval of Scientific and Technical Information,
PhD. Dissertation, 1996

[Koike95]
Koike Labs,
VisuaLinda: 3D Visualisation of Parallel Linda Programs,
On-Line Project Abstract, Koike Labs,
University of Electro-Communications in Tokyo, 1995
http://www.vogue.is.uec.ac.jp/vlinda.html

[Lynch60]
K. Lynch,
The Image of the City,
M.I.T. Press 1960.

[Mackinlay91]
J.D. Mackinlay, **S. Card** and **G.G. Robertson**,
Perspective Wall: Detail and Context Smoothly Integrated,
In Proceedings of the ACM SIGCHI '91 Conference on Human Factors in
Computing Systems, pp. 173-179, New Orleans, LA, USA, April 1991.

[Olsen93]
K.A. Olsen, **R.R. Korfhage**, **K.M. Sochats**, **M.B. Spring** and **J.G. Williams**,
Visualisation of a Document Collection: The VIBE System,
Information Processing and Management, Vol. 29, No. 1, pp. 69-81,
Pergamon Press Ltd, 1993.

[Quinn79]
N. Quinn and **M. Breur**,
A force directed component placement procedure for printed circuit boards,
IEEE Transactions on Circuits and Systems, CAS-26, (6), pp. 377-388, 1979.

[Rekimoto93]
J. Rekimoto and **M. Green**,
The Information Cube: Using Transparency in 3D Information Visualization,
Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93), pp. 125-132, 1993
ftp://ftp.csl.sony.co.jp/CSL/CSL-Papers/95/SCSL-TR-95-012.ps.gz

[Robertson91]
G.G. Robertson, **J.D. Mackinlay** and **S. Card**,
Cone Trees: Animated 3D Visualizations of Hierarchical Information,
Proceedings of the ACM SIGCHI '91 Conference on Human Factors in Computing Systems, pp. 189 - 194, New Orleans, LA, USA, April 1991.

[Sakar92]
M. Sakar and **M.H. Brown**,
Graphical fisheye views of graphs,
In proceedings of the ACM CHI '92, pp. 83 - 91, May 3-7, 1992.

[Shneiderman91]
B. Shneiderman,
Tree Visualization with Tree-maps: A 2-d space-filling approach,
Department of Computer Science and HCI Laboratory,
University of Maryland, June 1991
http://www.cs.umd.edu/Document/UMCP-CSD:CS-TR-2645

[Storey95]
M-A.D. Storey and **H.A. Müller**,
Manipulating and Documenting Software Structures Using SHriMP Views,
In Proceedings of the ICSM '95 conference on Software Maintenance,
Opio (Nice), France, October 17-20, pp. 275 - 284, 1995.

[Walker93]
G.R. Walker, **P.A. Rea**, **S. Whalley**, **M. Hinds** and **N.J. Kings**,
Visualisation of telecommunications network data,
BT Technology Journal, Vol. 11, No. 4, October 1993, pp. 54 - 63.

[Walker95]
G. Walker,
Challenges in Information Visualisation,
British Telecommunications Engineering Journal, Vol. 14, pp. 17-25, April 1995.
http://www.labs.bt.com/innovate/informat/infovis/index.htm

[Ware93]
C. Ware, **D. Hui** and **G. Franck**,
Visualizing Object Oriented Software in Three Dimensions,
In proceedings of CASCON '93 (IBM Centre for Advanced Studies),
pp. 612-620, Toronto, Ontario, Canada, October 1993
http://www.omg.unb.ca/hci/papers/IBM-CAS93-visualizing.ps.gz

[Ware94]
C. Ware and **G. Franck**,
Viewing a Graph in a Virtual Reality Display is Three Times as Good as a 2D Diagram,
In 1994 IEEE Conference on Visual Languages,
pp. 182-183, St. Louis, Missouri, USA, October 1994

[Wood95]
A.M. Wood, **N.S. Drew**, **R. Beale** and **R.J. Hendley**,
Hyperspace: Web Browsing with Visualisation,
In Third International World-Wide Web Conference Poster Proceedings,
pp. 21-25, Darmstadt, Germany, April, 1995.
http://www.cs.bham.ac.uk/~amw/hyperspace/www95

Bibliography

[Card96]
S.K. Card,
Visualizing Retrieved Information: A Survey,
Special Report, Computer Graphics and Visualization in the Global Information Infrastructure,
CG&A, Vol. 16, No. 2, March 1996
http://www.computer.org:80/pubs/cg%26a/report/g20063.htm

[Gershon95]
N. Gershon and **S.G. Eick**,
Visualization's New Tack: making sense of information,
IEEE Spectrum 32(11), pp. 38-56, November 1995