To appear in Proc. of East-West International Conference on
Multimedia, Hypermedia, and Virtual Reality, Moscow, Sept. 94.

Hooking Up 3-Space: Three-Dimensional Models as Fully-Fledged Hypermedia Documents

Keith Andrews and Michael Pichler
Institute for Information Processing and Computer Supported New Media (IICM),
Graz University of Technology,
Schießstattgasse 4a, A-8010 Graz, Austria.
{kandrews,mpichler}@iicm.tu-graz.ac.at

Abstract

This paper examines the incorporation of three-dimensional models into hypermedia systems as fully-fledged documents. Their use provides hypermedia authors with an additional, powerful means of presentation. Display, linking, navigational, and authoring aspects of 3D hypermedia documents are discussed and are illustrated with examples taken from the Harmony Viewer for the Hyper-G hypermedia information system.

1 Introduction

Hypermedia evolved from the concept of *hypertext*. Unlike the typical printed book, which is read sequentially from beginning to end, hypertext is inherently non-linear: it comprises many interlinked chunks of self-contained text. Readers are not bound to a particular sequence, but can browse through the information naturally by association, following their interests by clicking on highlighted keywords or phrases in one text to bring up another associated chunk of text.

Hypermedia is the generalisation of hypertext to include other kinds of media as well as text. Individual chunks of information are usually referred to as *documents* or *nodes*, and the connections between them as *links* or *hyperlinks* – the so-called *node-link* hypermedia model. A *source anchor* is the starting point of a hyperlink and specifies the part of a document from which an outgoing link can be activated. Typically, the user is given visual cues as to where source anchors are located in a document (for example, a highlighted phrase in a text document). A *destination anchor* is the end point of a hyperlink and determines what part of a document should be on view upon arrival at that node. [Nie90] and [BD91] are good introductory texts on hypertext and hypermedia.

Whereas traditional hypermedia systems have supported document types such as text, image, drawing, sound, and film, the incorporation of three-dimensional models as hypermedia documents has only recently begun to attract the attention of the research community. A three-dimensional model is typically described geometrically as a set of connected polygonal faces positioned in three-dimensional space (3-space). In the context of hypermedia documents, two distinct kinds of 3D model can be distinguished: a model of an *object* used as a kind of three-dimensional illustration where the model itself is manipulated (rotated, scaled, moved, etc.) and a model of a *scene* through which the user navigates (walks, flies, etc.) in an exploratory manner. An example of the former is a hypermedia motor car handbook, an example of the latter an architectural walkthrough. Hyperlinks within a 3D model may be anchored to objects or groups of polygons in the model.

Such use of 3D models in hypermedia is the topic of this paper. Following a survey of related work, we examine the display, linking, navigational, and authoring aspects of 3D hypermedia documents and conclude with a look at future work.

Some research has also been done in the area of three-dimensional overview maps to help alleviate the problem of user disorientation ("lost in hyperspace" [EH89]) in hypermedia systems. We are ourselves working on 3D information landscapes for hypermedia [AK94], similar to the landscape representation of Unix file hierarchies pioneered by FSN [TS92]. However, this work goes beyond the scope of this paper and will not be discussed further.

2 Related Work

Perhaps the most well-known work in the area of three-dimensional hypermedia is that of Card et al [RCM93, CRM91] at Xerox PARC on the Information Visualiser and 3D/Rooms. They allow users to interactively explore workspaces modeled as three-dimensional rooms. Particular data sets (nodes) are visualised in rooms of their own, and doors lead from one room to another in a manner similar to hypermedia links.

Serra et al [SCT91] discuss the use of 3D object hierarchies with attached multimedia documents. Each component in the 3D object hierarchy (part-of relationships) may be combined into a *concept node* with text, image, and video documents. Links may be made from these text, image, and video documents to other concept nodes. However, there is no support for arbitrary links from (parts of) the 3D object as such, and the 3D object hierarchy itself forms the entire extent of the hyperstructure.

Smith and Wilson [SW93] describe a prototype system based on HyperCard and Virtus Walkthrough (a 3D visualisation system), in the context of an academic departmental information system. They enabled users to interactively explore a 3D model of the department: when they approached within a certain distance of a source anchor, it automatically triggered to display a corresponding text document.

In his closing address at ECHT'92, the European Conference on Hypertext and Hypermedia, Jay Bolter presented his vision of "hypertextualised virtual reality" [Bol92], combining the immediacy of virtual reality with the representative power of hypertext. He proposed "writing on the world", immersive three-dimensional hypermedia where textual and other information is portrayed directly in the virtual model in the form of posters, stone inscriptions, and the like. Links might be represented as some kind of flying device like a magic carpet, whisking the reader to another location, as a tunnel entrance, the tunnel leading to a new location, or as a "magic window" into another world (which has the advantage of giving some kind of preview as to where one would go).

The above systems are very much hand-crafted, proof-of-concept prototypes. In Graz, we have implemented a fully-integrated 3D viewer [And92, Pic93] as part of the Harmony browser [AKS94, And93] for the Hyper-G hypermedia information system [AK94, Kap93]. Model description files are stored in the hypermedia database just like any other document. Users arriving at such a node (for example, by following a link from some other node in the hyperweb) are presented with a 3D representation of the model. They are then free to explore the model interactively at their own leisure using a variety of metaphors. Hypermedia links are anchored to individual objects or subobjects within the model and may be clicked to activate the link. Such anchors may be optionally highlighted, to gain an impression of which objects have links attached.

Figure 1 shows the Harmony 3D Scene Viewer being used to provide quick point-and-click access to satellite images: a model of the globe contains hyperlinks to corresponding images from the European Space Agency's Nimbus 7 satellite. In this example, an image of the United Kingdom and its associated text are shown.

3 Displaying 3D Models

The quality and speed of display of 3D models depends largely on the available hardware and software resources. Current display quality ranges from simple wireframe models, through shaded polygons, to real-time texture mapping. Radiosity methods have the potential to further increase realism. Real-time ray tracing methods are, at present, unachievable.

To guarantee interactivity, around 20 frames per second must be generated. The well-known trade-off between display rate and model complexity thus places an upper limit on the complexity of a model for a given hardware configuration. Techniques for polygon reduction such as visibility preprocessing and multiple levels of detail [FST92, FS93] can push this limit higher, but not eliminate it completely.

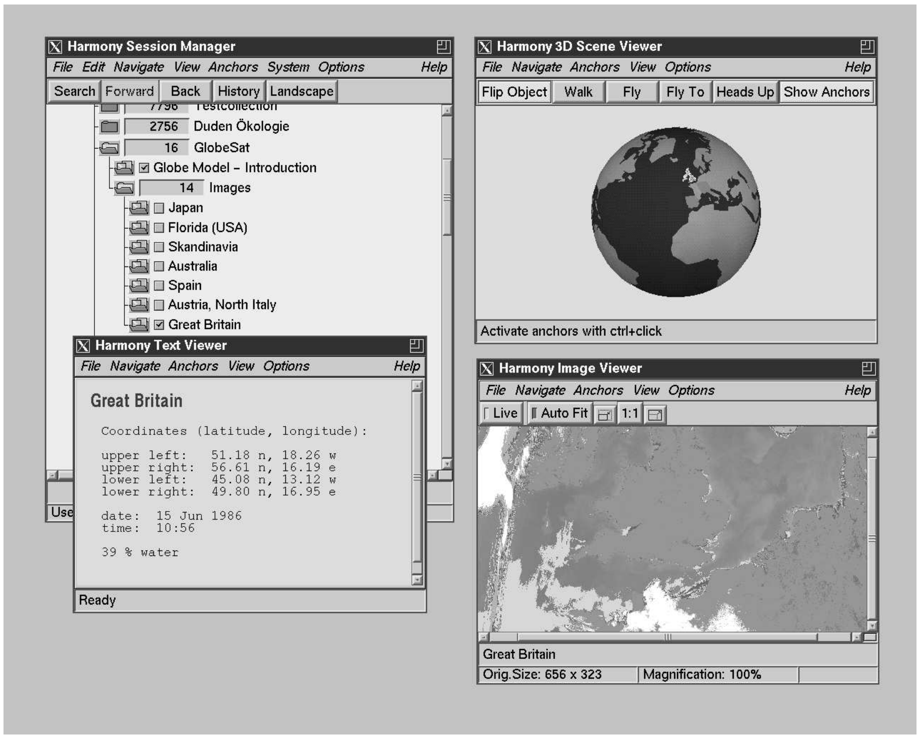

Figure 1: Harmony and its 3D Scene Viewer

4 Linking To and From 3D Hypermedia Documents

In conventional text documents, link anchors are generally words, phrases, or paragraphs in the text. The visual cue used to indicate source anchors is to highlight (reverse video, bold, different colour) the piece of text. On arrival at a text document, the text is scrolled so that the destination anchor is visible. In a raster image document, anchors might be defined as rectangular or polygonal areas of the image. Source anchors might be highlighted by displaying their boundaries. The image is panned so that destination anchors are positioned in the centre of the screen.

What are the possibilities for defining anchors in a 3D model? The most intuitive candidates to serve as source anchors are the individual objects in the model. This allows a natural mapping from an object in the model to more or related information about it. Anchoring links at the sub-object level, at individual polygons or surfaces of the object, is slightly more complicated but adds an extra level of flexibility. In order to activate a source anchor, the user must be able to select (pick with a mouse click) the corresponding object. This implies that only anchors currently in view can be activated, but since the user can freely explore the model, this imposes no great restriction.

An obvious candidate for a destination anchor is to use the position of the viewing camera, to transport the user to a particular view of the model.

Figure 2 shows a link being created from an object (the lamp) in a 3D model to another document (a text describing a virtual reality trip) in Harmony. The model is actually a (surprisingly tidy!) view of one of the authors' offices. The lamp has simply been selected and designated as source anchor. Harmony's Link Creator dialogue then pops up to allow specification of the destination, in this case the entire trip report document, and completes the link. Defining a destination anchor in a 3D model is equally intuitive: one navigates to the desired viewing position and then designates this position as the destination anchor.

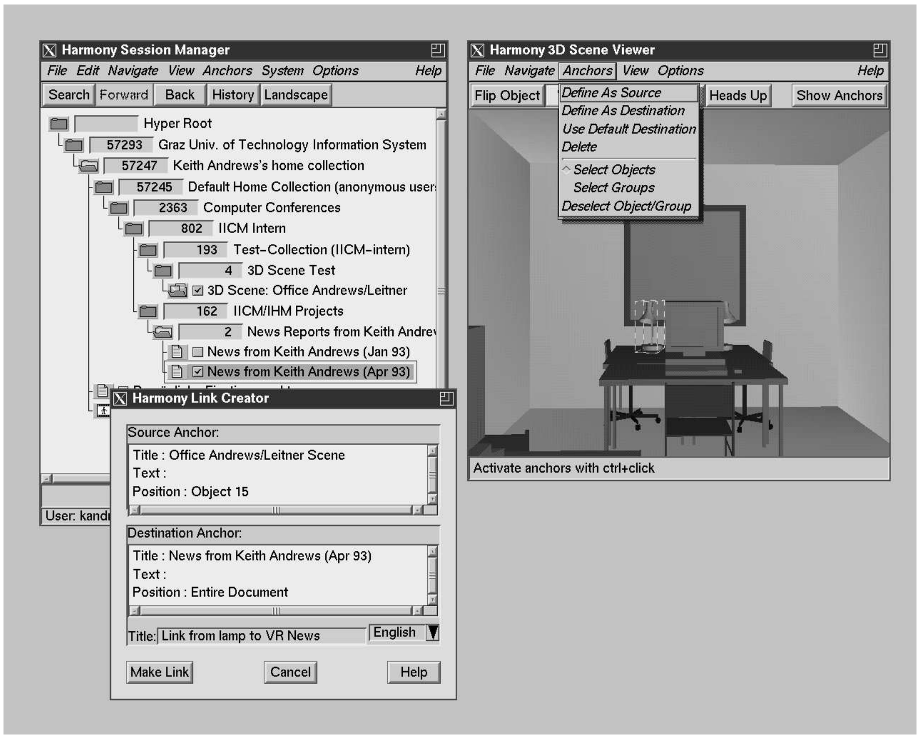

Figure 2: Creating a Link in Harmony

5 Highlighting Anchors in 3D Hypermedia Documents

Having agreed upon using objects and subobjects in the model as source anchors for hyperlinks, the question arises as to how to highlight anchor objects so that a user recognises them as being selectable. We decided that it should be possible to optionally turn anchor highlighting on and off and experimented [Pic93] with four methods for highlighting source anchors:

1. **Colour Code**: anchor objects have one colour, non-anchors another. User-configurable colour coding was implemented, with default colours of red-brown for source anchors and grey for the rest of the model.
2. **Bounding Cube**: source anchors are enclosed in their bounding wireframe cube. Thin bounding wireframe cubes of alternating yellow and red dashes were implemented.
3. **Brightness**: the natural colours of all objects in the scene are shifted so that anchors are relatively bright and non-anchors relatively dull. In our implementation, the HLS colour model [FvDFH90] was used, whereby the L (lightness) component was transformed to the range $[0.7, 1.0]$ for anchor objects and $[0.0, 0.3]$ for non-anchor objects.
4. **Edge Highlighting**: the edges of polygons belonging to source anchors are specially highlighted (the interior is drawn normally). In our implementation, thin lines, by default yellow in colour, were superimposed along the edges of source anchor polygons.

These techniques were evaluated both heuristically and with semi-formal usability tests. The evaluation indicated that colour coding and edge highlighting are the most usable of the techniques. The bounding cube method breaks down when a user gets very close to a source anchor object and the cube is no longer in the user's field of view (to some extent, the edge highlighting method exhibits similar behaviour). The brightness method feels somewhat unnatural: like being in a dark room containing luminescent objects. Colour coding makes anchors very obvious at the cost of destroying any illusion of reality. Edge highlighting is much less intrusive, while still adequately emphasising anchors.

Figure 3 shows the previously mentioned office scene. The lamp has been clicked to activate the link to the virtual reality trip report.

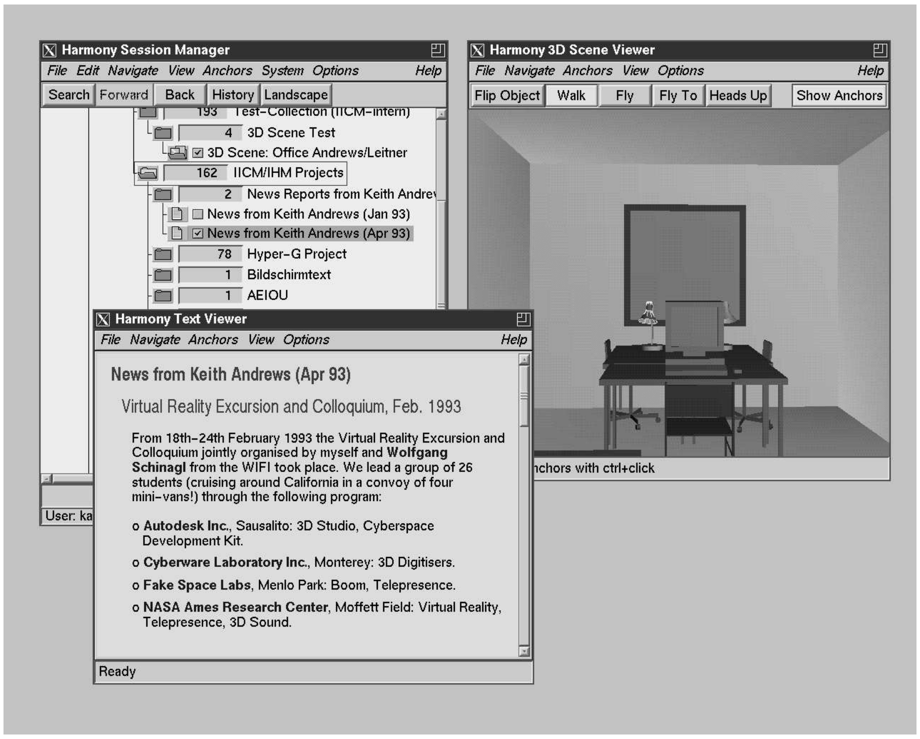

Figure 3: Anchor Highlighting Using Colour Coding

6 Navigation Within a 3D Hypermedia Document

In contrast to other kinds of hypermedia node (text, images, films, etc.), 3D hypermedia nodes require sophisticated mechanisms to be provided for navigation *within the node itself*. Whereas a text document may need to be scrolled, or an image panned or zoomed, a 3D model can be interactively explored by a user: object movement and/or viewpoint movement metaphors are essential.

Several authors (for example, [RCM93, BMB86, WO90, MCR90]) have published work on interactive movement in 3D models. These well-known techniques are easily transferable to 3D hypermedia documents and it suffices here to list the techniques implemented in the Harmony 3D Viewer:

1. **Flip Object**
   Three controls, translation, rotation, and zooming are provided to move the model itself rather than the user's viewpoint. These would typically be applied to models of an object.
2. **Walk**
   Forward and backward motion, sideways and vertical motion, and turning the head are assigned to the three mouse buttons.
3. **Fly**
   Flight direction is controlled by the position of the mouse cursor relative to the mid-point of the viewing window (denoted by a cross-hair). One mouse button activates flight, the other two control acceleration and deceleration.
4. **Heads-Up**
   Icons overlaid across the centre of the viewing window (like a pilot's heads-up display) denote eyes, a walking person, etc. Dragging from the eyes icon turns the viewer's head. Dragging from the body icon moves the viewer forwards or backwards in the horizontal plane, etc.
5. **Fly To**
   The user specifies a point of interest in the scene by clicking and flies toward it in a logarithmic fashion (see [MCR90]). Although this mode is not sufficient as a general navigation technique it is a useful addition to Heads-Up and Flip Object.

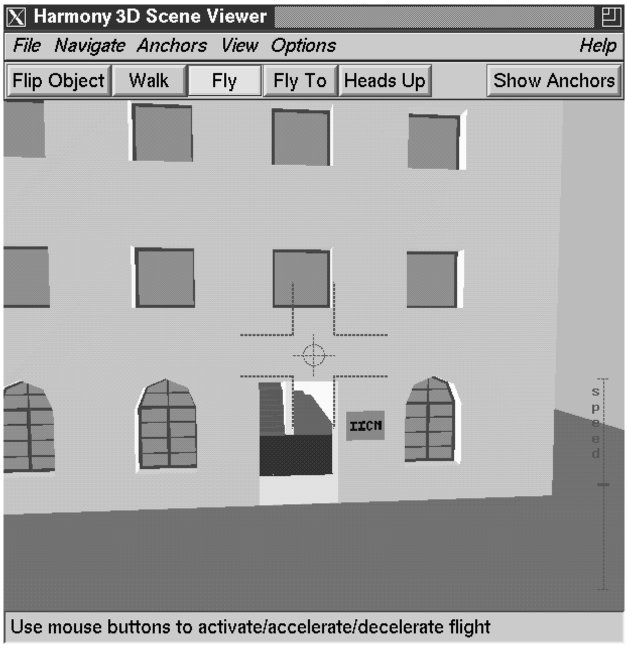

Figure 4: The Fly Navigation Mode

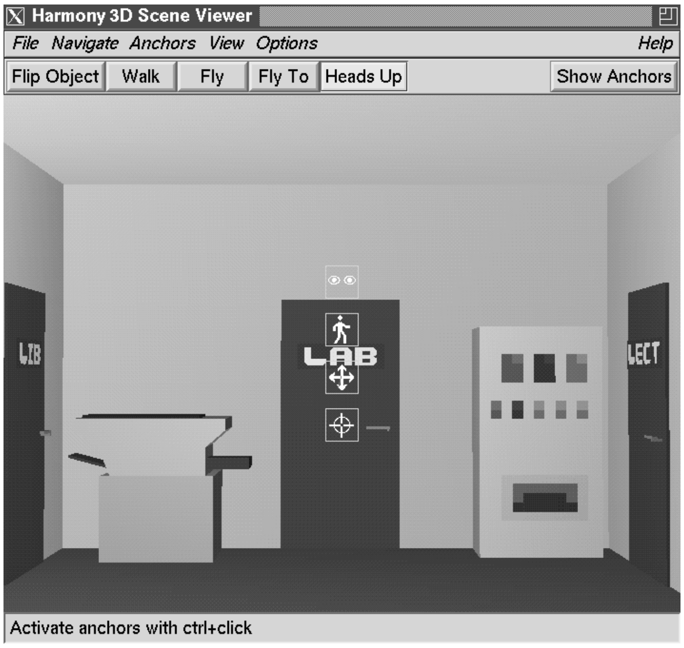

Figure 5: The Heads-Up Navigation Mode

Figure 4 illustrates the Fly navigation mode. The user is flying towards a model of the IICM building. Figure 5 shows an example of heads-up mode. The scene is a virtual model of the University of Auckland, introducing new students to the university's facilities.

7 Authoring 3D Hypermedia Documents

Powerful 3D modeling and scene composition programs (such as AutoCAD, 3D Studio, and Wavefront's Advanced Visualiser) are invaluable for modeling the geometry of a 3D document. The approach taken in Harmony is to upload finished model description files into the hypermedia database and provide interactive link editing tools as part of the 3D Viewer. As discussed previously, objects and subobjects in the model can be selected and designated as source anchors for hyperlinks. Similarly, the current camera position can be designated as a destination anchor.

8 Technical Details

Hyper-G is a large-scale, distributed, hypermedia information system based on the client-server model; Harmony is the Hyper-G Viewer (client) for Unix/X11. The Harmony 3D Viewer is implemented in C++ with the InterViews user interface toolkit for X Windows and uses Silicon Graphics' GL graphics library. Three-dimensional models are created with the Wavefront Advanced Visualiser and uploaded to a Hyper-G server.

9 Future Work

In the coming months, we intend to investigate polygon reduction techniques to improve the speed of the Harmony 3D Viewer and more realistic display techniques such as radiosity, real-time texture mapping, and stereoscopic viewing. We also intend tp extend the navigation techniques to cover glove input and other unorthodox methods.

As OpenGL, the new industry standard programming interface for 3D graphics, becomes more widely available, the Harmony 3D Viewer will be ported to it. We also intend to support further input formats for 3D models such as Autodesk's DXF.

10 Concluding Remarks

The incorporation of 3D scenes into a hypermedia system provides hypermedia authors with an additional, powerful means of presentation. Three-dimensional objects and scenes can be represented directly as such, complementing or replacing images or video clips, and users can interactively explore such models.

The problem of navigation in 3D scenes has been fairly well researched. The problem of how to embed hypermedia links into 3D scenes and how to indicate the presence of such links to the user has so far received little attention. The Harmony 3D Scene Viewer is a first step in this direction.

Further information about Harmony and Hyper-G is available by anonymous ftp from `iicm.tu-graz.ac.at` in directory `pub/Hyper-G`. The authors would like to thank the rest of the Harmony development team for their support.

References

[AK94] Keith Andrews and Frank Kappe. Soaring through hyperspace: A snapshot of Hyper-G and its Harmony client. In Wolfgang Herzner and Frank Kappe, editors, *Proc. of Eurographics Symposium on Multimedia/Hypermedia in Open Distributed Environments, Graz, Austria*, pages 181–191. Springer, June 1994.

[AKS94] Keith Andrews, Frank Kappe, and Jürgen Schipflinger. Harmony: A tool for navigating through deep hyperspace. To appear in Proc. of ED-MEDIA 94, Vancouver, Canada, June 1994.

[And92] Keith Andrews. Using 3D scenes as hypermedia nodes. Poster at ECHT'92, Milan, Italy, December 1992.

[And93] Keith Andrews. Constructing cyberspace: Virtual reality and hypermedia. Presented at Virtual Reality Vienna '93. Available by anonymous ftp from iicm.tu-graz.ac.at in directory pub/Hyper-G/papers, December 1993.

[BD91] Emily Berk and Joseph Devlin, editors. *Hypertext/Hypermedia Handbook*. Software Engineering Series. McGraw-Hill, New York, 1991.

[BMB86] Norman I. Badler, Kamran H. Manoochehri, and David Baraff. Multi-dimensional input techniques and articulated figure positioning by multiple constraints. In *Proc. 1986 Workshop on Interactive 3D Graphics*, pages 151–169, New York, October 1986. ACM.

[Bol92] Jay David Bolter. Virtual reality and the future of hypertext. In *Proc. ECHT'92*, page 2. ACM Press, December 1992.

[CRM91] Stuart K. Card, George G. Robertson, and Jock D. Mackinlay. The information visualiser, an information workspace. In *Proc. CHI '91: Human Factors in Computing Systems*, pages 181–188, New York, May 1991. ACM.

[EH89] Deborah M. Edwards and Lynda Hardman. Lost in hyperspace: Cognitive mapping and navigation in a hypertext environment. In R. McAleese, editor, *Hypertext: Theory into Practice*, pages 105–125. Blackwell Scientific Publications Ltd., 1989.

[FS93] Thomas A. Funkhouser and Carlo H. Séquin. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In *Proc. SIGGRAPH '93, Anaheim, California*, pages 247–254, New York, August 1993. ACM.

[FST92] Thomas A. Funkhouser, Carlo H. Séquin, and Seth J. Teller. Management of large amounts of data in interactive building walkthroughs. In *Proc. 1992 Symposium on Interactive 3D Graphics*, pages 11–20, New York, April 1992. ACM.

[FvDFH90] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, Massachusetts, second edition, 1990.

[Kap93] Frank Kappe. Hyper-G: A distributed hypermedia system. In Barry Leiner, editor, *Proc. INET '93, San Francisco, California*, pages DCC–1–DCC–9. Internet Society, August 1993.

[MCR90] Jock D. Mackinlay, Stuart K. Card, and George G. Robertson. Rapid controlled movement through a virtual 3D workspace. In *Proc. SIGGRAPH '90, Dallas, Texas*, pages 171–176, New York, August 1990. ACM.

[Nie90] Jakob Nielsen. *Hypertext & Hypermedia*. Academic Press, San Diego, CA, 1990.

[Pic93] Michael Pichler. Interactive browsing of 3d scenes in hypermedia: The hyper-g 3d viewer. Master's thesis, Graz University of Technology, Austria, October 1993.

[RCM93] George G. Robertson, Stuart K. Card, and Jock D. Mackinlay. Information visualization using 3D interactive animation. *Communications of the ACM*, 36(4):56–71, April 1993.

[SCT91] Luis Serra, Tat-Seng Chua, and Wei-Shoong Teh. A model for integrating multimedia information around 3d graphics hierarchies. *The Visual Computer*, 7(5-6):326–343, May/June 1991.

[SW93] Pauline A. Smith and John R. Wilson. Navigating in hypertext through virtual environments. *Applied Ergonomics*, 24(4):271–278, August 1993. Butterworth-Heinemann Ltd.

[TS92] Joel Tesler and Steve Strasnick. *FSN: The 3D File System Navigator*. Silicon Graphics, Inc., Mountain View, CA, 1992. Available by anonymous ftp from sgi.sgi.com in directory sgi/fsn.

[WO90] Colin Ware and Steven Osborne. Exploration and virtual camera control in virtual three dimensional environments. In *Proc. 1990 Symposium on Interactive 3D Graphics*, pages 175–183, New York, March 1990. ACM.