

Hooking Up 3-Space: Adding the Third Dimension to Hypermedia

Keith Andrews

Institute for Information Processing and Computer Supported New Media (IICM),
Graz University of Technology,
Austria

ABSTRACT

This paper examines some of the issues involved in incorporating three-dimensional scenes as nodes in hypermedia systems. A general outline is followed by specific consideration of two user interface issues: the definition and representation of link anchors and the provision of appropriate navigation metaphors within a 3D hypermedia node. Preliminary results from a prototype implementation of 3D hypermedia nodes are presented, in particular: four methods of highlighting anchor objects and three navigational metaphors. The paper concludes with a short discussion of authoring 3D hypermedia documents.

1 Keywords

Hypermedia, 3D, Link Anchors, Navigation.

2 Introduction

Hypermedia systems typically combine integral nodes of information with navigable links between them – the so-called *node-link model*. The reader is referred to [5] and [9] for good introductions to hypertext and hypermedia. Nodes are documents of any of a number of types: text, image, drawing, sound, film, and so forth. The parts of a document to which links are attached are called *anchors*. Nodes containing graphical information usually take the form of either raster images (e.g. scanned pictures) or 2D object-oriented pictures (e.g. from a drawing editor). Film sequences of raster images have been used to give the impression of travelling through 3D space, but the incorporation of true 3D objects and scenes is still

uncommon, if not unknown.

A three-dimensional scene is typically described by geometric models of various objects, positioned somewhere in 3-space. Users are able to move freely around the scene in real time and their view is updated simultaneously. This is the main difference between a true 3D scene and presenting the user with a sequence of pre-calculated views. Current graphics workstations are capable of displaying quite complex scenes of smooth-shaded polygons at interactive speeds.

Hypermedia links can be anchored to a 3D scene as a whole, or to individual objects within the scene. A user arriving at such a scene (by following a link from some other node in the hypermedia web) is presented with a 3D representation of the scene. Various objects in the scene may be highlighted in some way to indicate that they are source anchors, i.e. have links to other nodes which may be 3D scenes or any other kind of hypermedia node. The user is free to look around the scene interactively at his/her own leisure using a variety of navigational metaphors. Selecting (clicking on) a highlighted object activates its link. Depending on the system, this may cause the new node to be displayed instead of the originating 3D node, or displayed simultaneously in a new window.

There is a wide range of possible applications for 3D scenes in hypermedia. Two primary uses have emerged from our work: *3D illustrations* where the 3D scene is a model of some object under consideration and *virtual worlds* where the user navigates around the 3D scene in an exploratory manner.

An example of the former is a hypermedia motor car handbook with 3D illustrations (see Figure 1). The first node might be a 3D scene showing a model

of the motor car. The viewer can freely manipulate the model, looking at the car from all angles. Clicking on the bonnet activates a link to another scene with the bonnet open, exposing the engine. Clicking on the oil filter leads to a text, a sound clip and a 3D illustration of that component, etc.

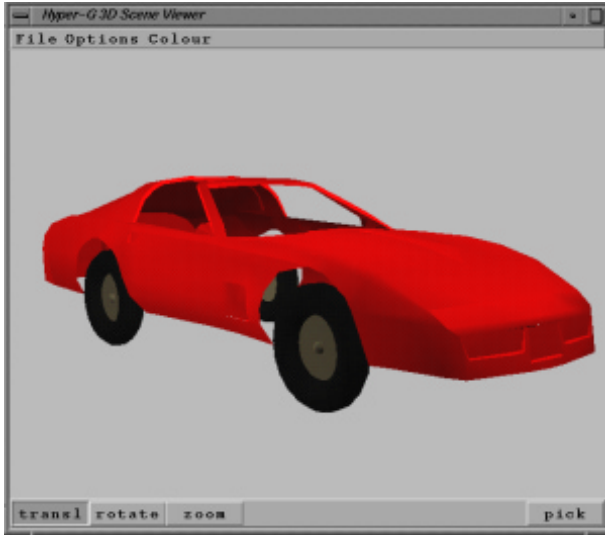


Figure 1: Example of 3D Illustration

An example of the latter is an architectural application (see Figure 2), where a CAD model of a building is augmented with hypermedia links to additional information. The user can freely move around the building, clicking on (selectable) objects of interest for further information.

3 Previous Work

Perhaps the closest published work in this area is that of Card et al ([10, 4]) at Xerox PARC on the Information Visualiser and 3D/Rooms. They allow users to interactively explore workspaces modeled as 3D rooms. Doors lead from one room to another, in a similar manner to hypermedia links. Several authors (e.g. [2, 8, 12]) have published work on interactive movement in 3D scenes.

To my knowledge, however, there exists no published research on the general problem of incorporating 3D scenes into hypermedia systems. In his closing address at ECHT'92, the European Conference on Hypertext and Hypermedia, Jay Bolter presented his vision of “hypertextualised virtual re-

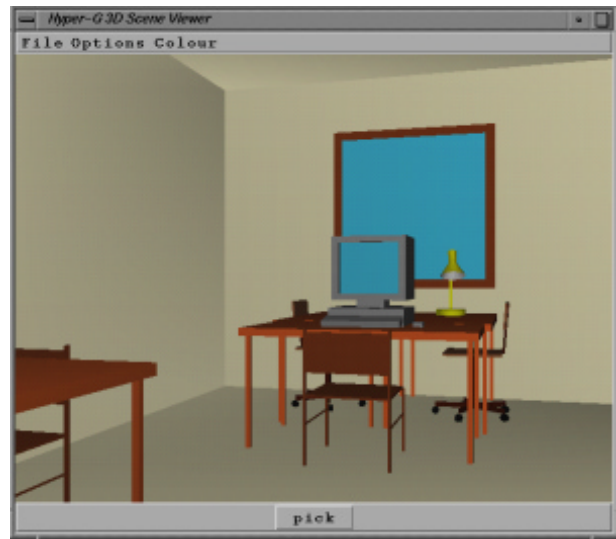


Figure 2: Example of Virtual World

ality” [3], combining the immediacy of virtual reality with the representative power of hypertext. This paper makes a first attempt to outline some of the issues, problems, and advantages of using 3D scenes in hypermedia.

4 Linking To and From 3D Hypermedia Nodes

4.1 Nodes, Links, and Anchors

Consider a generalised model of a hypermedia system: nodes are of one of many document types and anchors serve as source and destination for links between nodes. A *source anchor* 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. A *destination anchor* determines what part of a document should be on view upon arrival at that node. Sometimes, source and destination anchors are specified to cover the whole document.

In a conventional (hyper)text document, the 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 areas of

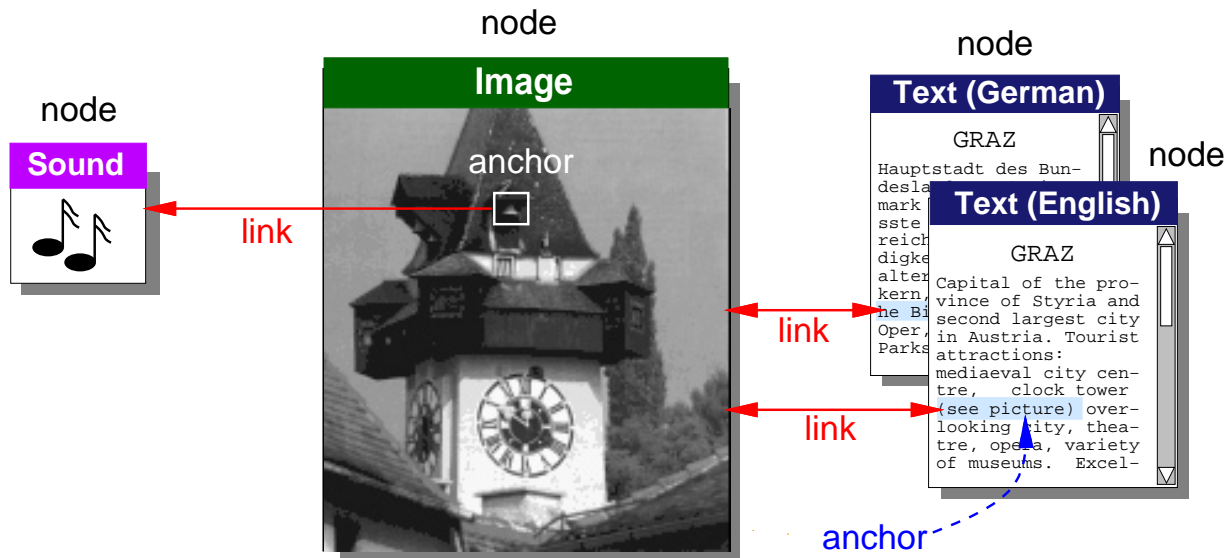


Figure 3: Nodes, Links, and Anchors in Typical Hypermedia Documents

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. Figure 3 illustrates nodes, links, and anchors in the context of some typical hypermedia documents.

4.2 Defining Anchors in 3D Scenes

What are the possibilities for defining anchors in a 3D scene? The most intuitive candidates to serve as anchors are the individual objects in the scene. In the case of source anchors, this allows a natural mapping from an object in the scene to more or related information about it. For destination anchors, the viewing camera can be positioned so that the destination anchor object is in the centre of the field of view. Anchoring links at the sub-object level, at individual polygons or surfaces of the object, is also a possibility, but introduces an unnecessary layer of complexity. In any case, such functionality can be simulated by breaking an object into several smaller objects. Note that, for the purposes of this paper, I assume a polygonal representation of objects in a scene.

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 move freely around the scene, this imposes no great restriction.

4.3 Highlighting Anchors in 3D Scenes

Having agreed on using objects in the scene as hypermedia anchors, the question arises as to how to highlight objects so that a user recognises them as being selectable (source anchors). Leading 3D modeling packages such as 3D Studio [1], TDI Explore [11], and the Wavefront Advanced Visualiser [13] use some kind of colour coding to indicate groups of selected objects during the editing process. This is also perhaps the most obvious method of highlighting source anchors in 3D scenes, but there are a number of other possibilities.

By extending a number of common techniques for highlighting objects in two dimensions, a tentative list of eight possible highlighting methods for 3D objects was drawn up:

1. **Colour Code**

Anchors objects have one colour, non-anchors another. See Figure 4. In our implementation, coding source anchors red and the rest of the scene grey produced aesthetically pleasing results. A commercial application would probably provide a sensible default and let the user

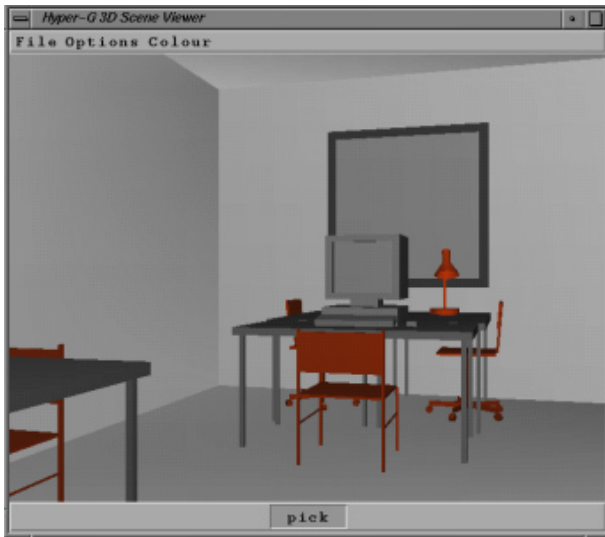


Figure 4: Colour Code Method

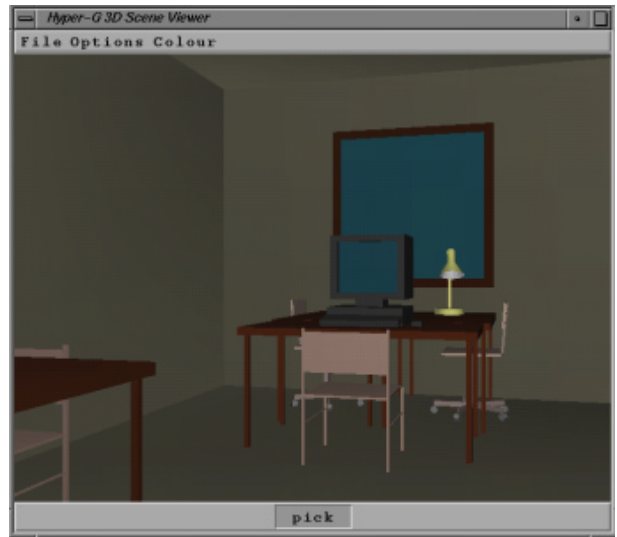


Figure 6: Brightness Method

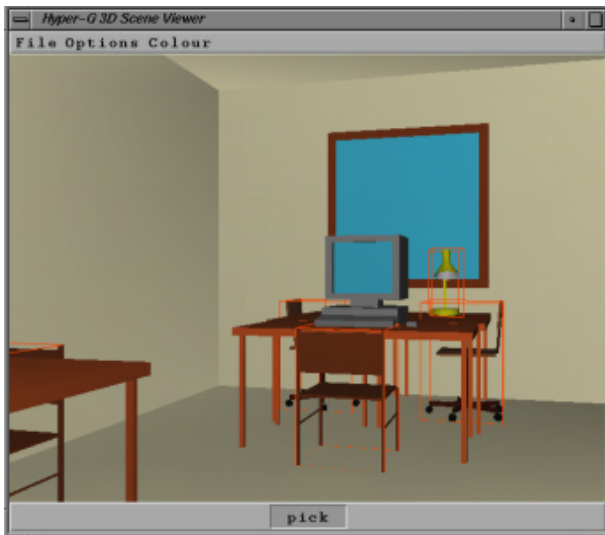


Figure 5: Bounding Cube Method

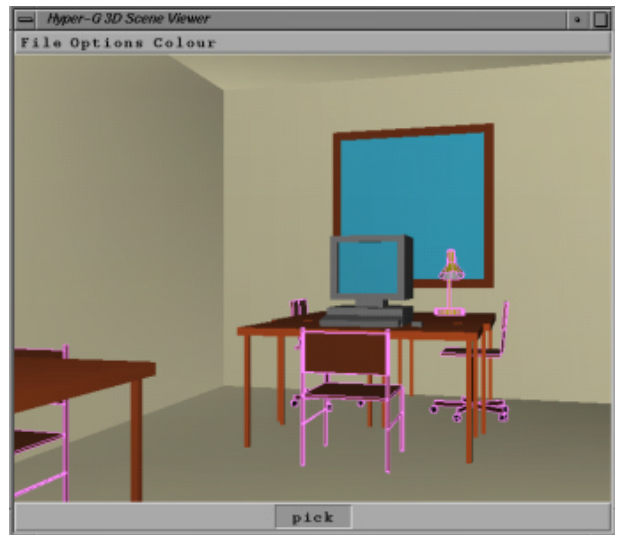


Figure 7: Edge Highlighting Method

configure his own colour code preference.

2. **Bounding Cube**

Source anchors are enclosed in their bounding wireframe cube. See Figure 5. We use thin red bounding cubes, but a commercial application should probably let the user set his own preference.

3. **Brightness**

The natural colours of all objects in the scene are shifted so that anchors are relatively bright and non-anchors relatively dull. See Figure 6. We do this by using the HLS colour model [6] and transforming the L (lightness) component for anchors to the range [0.6, 1.0] and for non-anchors to [0.0, 4.0]. This approach requires more computational work than the others, since all objects in the scene must be colour-transformed.

4. **Edge Highlighting**

The edges of polygons belonging to source anchors are specially highlighted (the interior is drawn normally). See Figure 7. Our prototype superimposes magenta lines along the edges of source anchor polygons. A more visually appealing variant might be to run lengths of virtual rubber tubing along the edges.

5. **Halo**

Source anchors are shrouded in some kind of halo, like the classical vision of a deity.

6. **Textures**

The surface texture of source anchors is modified in some way, for example by applying a bump map. This, however, assumes the availability of real-time texturing.

7. **Colour Pulse**

Source anchor objects pulsate (change their colouring over time in some regular sequence).

8. **Animated Edge**

The polygonal edges of source anchor objects are animated, for example by draping flashing Christmas tree lights along them.

The first four of these highlighting methods have been implemented in a prototype system (the

Hyper-G 3D Viewer prototype¹) running on Silicon Graphics machines under Unix with the GL graphics library. A number of 3D models created with the Wavefront Advanced Visualiser were used for testing, including a model of our institute building, a couple of offices, a motor car, and an engine. In our prototype, highlighting of anchors is not switched on by default; the user specifically activates highlighting by clicking the “Pick” button.

Initial informal usability evaluations indicate that colour coding and edge highlighting are the most usable of these methods. The bounding cube method breaks down when a user gets very close to a source anchor object and the bounding 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.

The use of auditory cues was also considered: when the cursor is located over an anchor, some kind of sound could be produced. However, apart from the computational overhead of continuously determining which object is currently targeted, it would result in the user scanning the scene with the cursor for anchors like a sapper in a minefield!

A final possibility would be to offer no cues at all, forcing the user to try clicking an object to know whether or not it is an anchor. In this case, a single-click should indicate whether or not an anchor is present and a double-click be required to activate it.

Different styles or colours of highlighting could be used to give the user a hint as to what kind of link is present. For example, using colour coding, links to other 3D nodes could be coloured red, links to text nodes green, and so forth. A common problem in hypermedia, that of being overwhelmed by excessive numbers of links is ameliorated in 3D hypermedia nodes – only those link anchors visible from the current viewpoint are active.

¹Hyper-G is a major hypermedia system under development at this institute. See [7] for details.

5 Navigation Within a 3D Hypermedia Node

In contrast to other kinds of hypermedia node (text, images, 2D drawings, 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 scene can be interactively explored by a user: object movement and/or viewpoint movement metaphors are essential.

Three navigational metaphors were implemented in our prototype system:

1. Flip

Three controls, translation, rotation, and zooming are provided which move the scene itself rather than the user's viewpoint.

2. Walk

Pressing the left mouse button and dragging the cursor up or down results in forward or backward motion; dragging left or right turns the viewer to the left or right. Combinations are allowed, for example walking forward while turning to the left.

Pressing the middle mouse button and dragging the cursor up or down results in vertical motion up or down; dragging left or right results in sideways motion to the left or right.

3. Fly

Clicking the left mouse button toggles flying on and off. The middle mouse button controls acceleration, the right mouse button deceleration. The position of the cursor relative to the mid-point of the screen (denoted by a cross-hair) modifies the direction of flight.

Preliminary test results show, as expected, that the flip metaphor (object movement) is most useful for viewing 3D models. The walk and fly metaphors (viewpoint movement) are most useful for exploration of virtual environments. Walking can become somewhat irritating during prolonged use, due to continual dragging of the mouse. While flying, test users experienced problems overshooting, which prompted us to build in an "emergency stop" feature to immediately come to a halt.

6 Authoring a 3D Scene Document

Powerful 3D modeling and scene composition programs are invaluable for the author of a 3D scene document. The main problem is that current packages (such as [1, 11, 13]) were not designed with hypermedia in mind and there is no provision for the interactive specification of anchors and links. One must either "bolt" this functionality onto an existing scene compositor, or else specify anchors and links manually after composing the scene (e.g. by editing a text file containing a list of anchor objects and their incoming and outgoing links). We resorted to the latter technique for our test models.

Ideally, a hypermedia system would provide interactive authoring tools for each supported document type. A link could then be hung between, say, an object in a 3D scene and a paragraph of a text document, simply by clicking on the appropriate anchors in the two documents (in a windowing system, each document would typically be displayed in a window of its own).

7 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, rather than as (sequences of) flat 2D views. Users can interactively explore such 3D 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 in the literature. This paper is a first step in that direction.

In the future, research will also be needed in the area of integrated authoring environments for 3D hypermedia nodes. Only then can Jay Bolter's vision be realised in practice.

8 Acknowledgements

I would like to thank Michael Pichler for implementing the methods presented here in the prototype Hyper-G 3D Viewer and my colleagues at the institute, especially Nick Sherbakov, for valuable critique and suggestions.

REFERENCES

- [1] Autodesk, Inc. *3D Studio*. Autodesk, Inc., Sausalito, California.
- [2] N. I. Badler, K. H. Manoochehri, and D. 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, Oct. 1986. ACM.
- [3] J. D. Bolter. Virtual reality and the future of hypertext. In *Proc. ECHT'92*, page 2, Milan, Italy, Dec. 1992. ACM Press.
- [4] S. K. Card, G. G. Robertson, and J. D. Mackinlay. The information visualiser, an information workspace. In *Proc. CHI'91: Human Factors in Computing Systems*, pages 181–188, New Orleans, Louisiana, Apr. 1991. ACM.
- [5] J. Conklin. Hypertext: An introduction and survey. *IEEE Computer*, 20(9):17–41, Sept. 1987.
- [6] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, Massachusetts, second edition, 1990.
- [7] F. Kappe, H. Maurer, and N. Scherbakov. Hyper-G – a universal hypermedia system. *Journal of Educational Multimedia and Hypermedia*, 2(1):39–66, 1993.
- [8] J. D. Mackinlay, S. K. Card, and G. G. Robertson. Rapid controlled movement through a virtual 3D workspace. In *Proc. SIGGRAPH '90, Dallas, Texas*, pages 171–176, New York, Aug. 1990. ACM.
- [9] J. Nielsen. *Hypertext & Hypermedia*. Academic Press, San Diego, CA, 1990.
- [10] G. G. Robertson, S. K. Card, and J. D. Mackinlay. Information visualization using 3D interactive animation. *Communications of the ACM*, 36(4):56–71, Apr. 1993.
- [11] Thomson Digital Image. *EXPLORE*. Thomson Digital Image.
- [12] C. Ware and S. Osborne. Exploration and virtual camera control in virtual three dimensional environments. In *Proc. 1990 Symposium on Interactive 3D Graphics*, pages 175–183, New York, Mar. 1990. ACM.
- [13] Wavefront Technologies. *Advanced Visualiser User's Guide*. Wavefront Technologies, Inc., Santa Barbara, California, 1991.