Context in 3D Planar Navigation

Scott Vallance
School of Informatics & Engineering
The Flinders University of South Australia
(vallance@infoeng.flinders.edu.au)

Paul Calder
School of Informatics & Engineering
The Flinders University of South Australia
(calder@infoeng.flinders.edu.au)

Keywords: 3D graphics, navigation techniques, distortion viewing.

Abstract

One of the most frustrating barriers to the widespread use of 3D visualisation is the additional complexity in navigating 3D data. This paper details a new approach to improving navigation in 3D environments where the navigation is mainly planar. Data at a distance from the viewpoint is distorted as if projected onto a partial cylinder to approximate a plan view, thereby exposing information that may have been obscured. Previous approaches are compared with this new technique and screenshots presented. Implementation details of the technique are discussed as well as possible performance and useability issues.

1. Introduction

Navigating a virtual 3D space armed with nothing more than a 2D mouse, a keyboard and 2D screen is a difficult task. Systems which simply allow a user 6-degrees-of-freedom movement do so at the risk of making a difficult to use, unintuitive interface. Metaphors are a powerful way of simplifying interactive tasks and making it intuitive. In 3D navigation the most obvious metaphor is that of a virtual person.

In a virtual world, constraining the view to that of a human raises a few undesirable features. Firstly, down at ground level it can be very difficult to see where to go. Secondly, it can be hard to make sense of the lie of the land as objects close to the viewpoint can obscure important detail. There are three different types of views traditionally used with a virtual person: first, second and third person. A first person view looks out from the eyes of the virtual person. A second person view follows or tracks the virtual person, but from outside. A third person view is independent of the virtual person. First person views are the most direct.

Successful navigation relies on a clear understanding of the current context. The context is the surrounding information that allows a greater understanding of the environment. In a word processor the amount of information that can be seen either side of the line currently being typed is the context. In a 3D navigation problem, the context is the amount of data seen around the current location or focus. Trivially, some types of context can be increased by a larger field of view, but this context comes at the expense of the detail and resolution of the objects surrounding the viewpoint. Attempts have been made in 2D and 3D visualisation to increase this context without losing detail, examples of this work can be seen in [WINCH2000], [KEAHEY1998].

The goal of the work reported in this paper is to increase the context for navigation without unduly decreasing the directness of the view. Section 2 details a task that exposes shortcomings of previous approaches and our solution to it. Section 3 presents a comparison of the new technique with previous ones. Section 4 explains the implementation of the technique, followed by the conclusion in Section 5. Future work arising from the paper is briefly outlined in Section 6.

2. Navigation tasks

This paper investigates techniques for improving navigation in 3D scenarios where the navigation is substantially planar. In particular we have investigated maze navigation scenarios where the task involves both navigation and interaction with the local environment.

Tasks involving just navigation might best be solved by providing a third person plan view. Tasks involving just interaction with the local environment might best be solved by providing a first person view. Tasks involving both navigation and interaction with the local environment require more complex views.

2.1 Techniques

Common approaches to tasks such as the one detailed above can involve single views or multiple views. Single views consist of a first, second or third person view. Multiple views traditionally involve a main view and secondary view to provide context. Examples of each of these techniques are:

- Single view: a first person view, such as in a flight simulator.
- Multiple views: a first person view, with a map (plan view) on top.

Another approach to the problem is World-in-Miniature [STOAKLEY1995]. Here a small third person view shows all of the data in a miniaturised form, and an representation of the user can be moved through the data. A plan view on a first person view is a particular type of World-in-Miniature visualisation.

While a combination of a first and third person view (for example a superimposed plan view) satisfies the navigation task by providing both context and first person directness, it contains an undesirable separation between the information. To navigate, the user need only focus upon the plan view. When interacting directly with the local environment the user must switch to focusing on the first person view. Although this takes only a small amount of time, it reduces the consistency of the interaction, and as a result it may negatively impact on the user's spatial understanding.

2.2 Removing the separation between viewpoints

To remove this separation, we rely on the fact that the information provided by each viewpoint is not needed equally for data at different distances to the user. Put simply, close data needs to be viewed in first person and far data needs to be viewed in either second or third person. To implement this, data is distorted according to its distance from the user, so that close data is the same as a normal first person view, while distorted data approximates looking at that data from a second person viewpoint above the maze. This can be seen in the following figure, which shows a distorted maze:

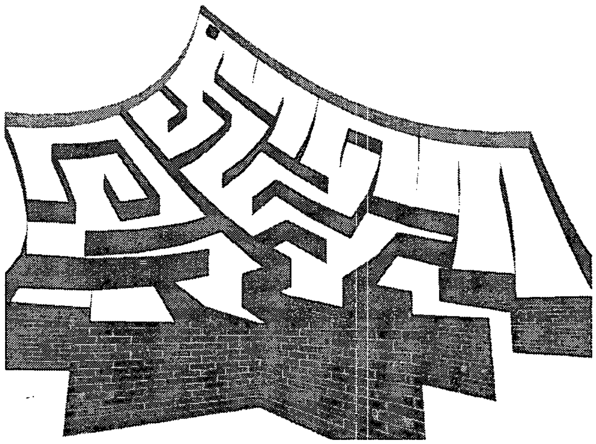

Figure 1: A maze distorted to show context

In this approach both context and directness of interaction are provided. The disadvantage of this approach is the loss of context immediately around the viewpoint. Detail cannot be seen over the walls very close to the viewpoint in the above figure.

3. Comparison

In this section screenshots of various techniques that might be used to combine navigation and local interaction are presented and discussed. The amount of context each technique provides can clearly be seen. Examples of single view solutions to the task will be presented first, followed by multiple views and finally distorted views.

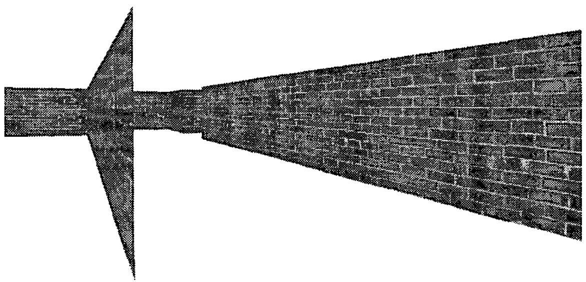

Figure 2: A first person view of the maze

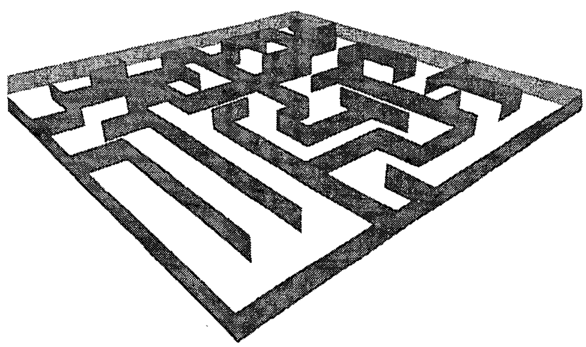

Figure 3: A third person view of the maze

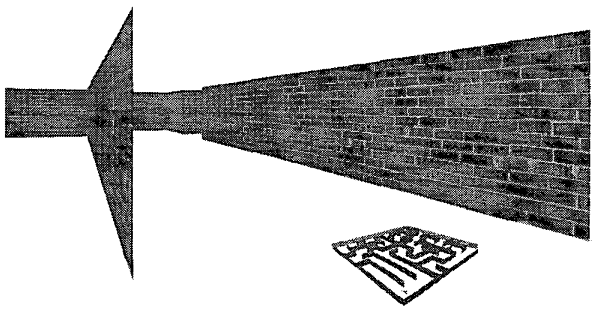

Figure 4: A World-in-Miniature view

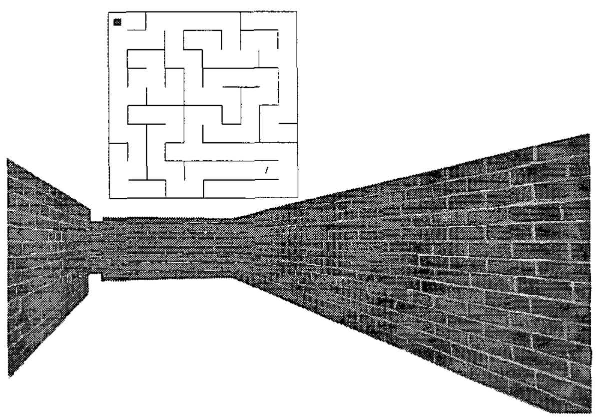

Figure 5: Plan view and first person view combined

Displaying only a single view limits the interaction. The view in Figure 2 allows easy local interaction but difficult navigation and the view in Figure 3 has easy navigation at the expense of directness. Navigation in Figure 2 is difficult because the choice of route is unclear without exploration. Route exploration can be time consuming, and is undesirable in most real world (non-entertainment) applications.

To alleviate the need to explore without removing the directness of a first person view, multiple viewpoints can be used. Figures 4 and 5 provide a discrete combination of viewpoints. Navigation is possible with a view that provides context (the smaller view in both figures), and local interaction is easy through the larger first person view.

Figures 4 and 5 suffer from a need to switch from one view to another, this can be quite disconcerting as is discussed in [PAUSCH1995]. In their paper they present a method of specifying navigation with World-in-Miniature views to alleviate this problem. Once the user specifies a path, the context view rotates around and zooms in to become the first person view. There still exists a fundamental separation in views with this technique, and rapid swapping between the context and first person view, may become confusing.

Separation of focus tends to lead to confusion when the user changes focus. The user is unsure of how the viewpoints relate to one another. In tasks where the user's spatial awareness must not be compromised, such as planning and architecture, this can be a serious concern.

The next series of figures show the bending technique, with varying bending *thresholds* (see Section 4).

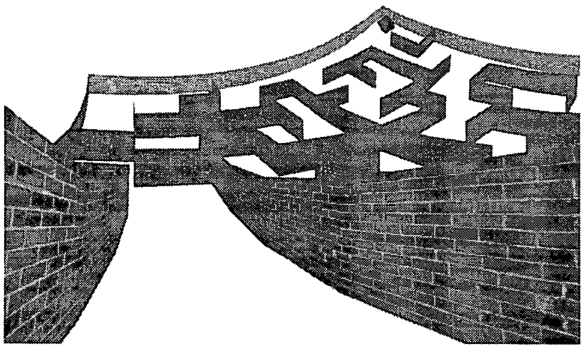

Figure 6: Bending navigation with a 'close' threshold.

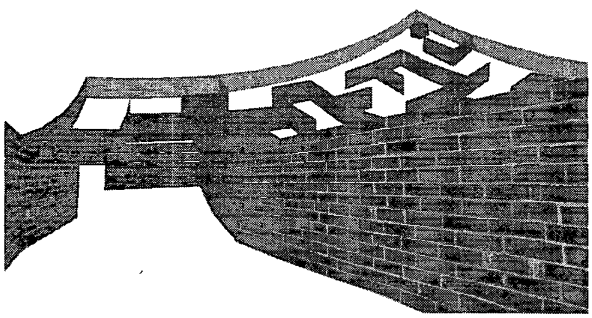

Figure 7: Bending with a 'medium' threshold.

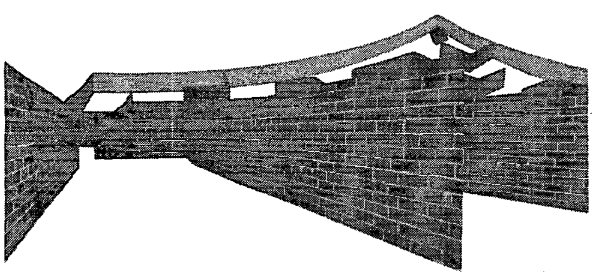

Figure 8: Bending with a 'far' threshold.

Unlike the plan view, the distorted view does not show all the maze. This means it is not a perfect tool for navigation, and may require, at some stages, the user to move around to gain the appropriate context. However continuity between navigation information and direct viewing is not compromised with this technique.

Scenarios in which navigation data and local environment data are intertwined present a problem for this technique. For instance, when seeing if a close object aligns with a couple of more distant objects, bending will make this impossible. A solution would be to have some areas distorted and others left alone, but specifying this would be difficult.

4. Implementing the technique

To make a smooth transition from the first person view, to the distorted view, each vertex is rotated separately according to its distance from the user. This distance is measured only in terms of the plane the data lies on, so the effect is similar to bending the plane itself. The plane of the maze is simply the *xz*-plane. This means that the vertical displacement of the vertex from the user has no effect on its rotation. Figure 9 shows the idea.

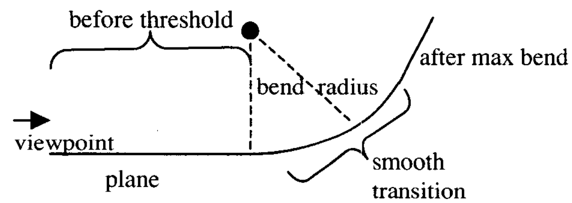

Figure 9: Schematic of the plane after bending

In general, to find the distance from the user in terms of some plane, first project both the user's position and the vertex onto that plane. This is easy in the specific case where the plane is the *xz* plane. All that needs to be done is to discard the *y* axis value. So the distance measure is:

$$d = \sqrt{(x_v - x_p)^2 + (z_v - z_p)^2}$$

Where:
d = distance
x_v = x coordinate of the viewpoint
z_v = z coordinate of the viewpoint
x_p = x coordinate of the vertex
z_p = z coordinate of the vertex

Once the distance is obtained it is checked against a threshold value. The threshold specifies where the bending effect should begin. Additionally there is a radius value for the bend, which effects how large the smooth transition area is. Once the point is further than the distance threshold plus the radius, it is simply rotated by the maximum amount. So the amount of rotation is:

$$\begin{cases} 0 & d < t \\ \dfrac{d - t}{r} & d < t + r \\ m & otherwise \end{cases}$$

Where:
d = distance
t = threshold
r = radius
m = maximum bend

The amount of rotation increases linearly over the *bend radius*. Non-linear relationships between rotation and distance have subtle effects on the technique. A non-linear function can be used to instigate a slow onset of rotation followed by a fast climax. This would look like:

$$\begin{cases} 0 & d < t \\ (\dfrac{d - t}{r})^{f} & d < t + r \\ m & otherwise \end{cases}$$

Where:
f = some exponent, larger values mean a quicker transition
d = distance
t = threshold
r = radius
m = maximum bend

The point and axis of rotation depend on the exact kind of bending effect desired. In the prototype system, vertexes are distorted in a fashion similar to if they were projected onto a portion of a cylinder. It differs slightly from a cylinder in that vertices are distorted more according to their distance, even if they are the same distance from the start of the cylinder. The cylinder runs perpendicular to the direction of the view. So for each vertex, the axis of rotation is also in the direction of the view vector. The point around which the rotation occurs is above the closest point from the vertex, on the line perpendicular to the user in front of them and *threshold* distance from them. This can be seen in figure 10.

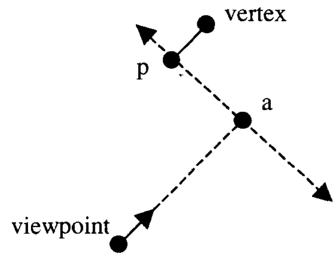

Figure 10: Determining the point of rotation for a cylindrical bend

In this diagram, a is the point *threshold* distance from the user, and p is the point closest to the vertex on the line perpendicular from the viewpoint. So the point of rotation is above p by the amount of the bend radius. In the case of the plane being the xz plane, above simply means setting the y value to being that of the bend radius. If the plane is arbitrary, then the point of rotation is above the plane (with respect to the viewer) in the direction perpendicular to the plane. For a spherical bend, the point of rotation would simply lie on the line between the viewpoint and the vertex, projected onto the plane.

Once the axis of rotation, the point of rotation and the amount of rotation are established it is simply a matter of using any maths library to implement the transformation. Unfortunately, the OpenGL API does not allow rotation transformations to be applied to vertices separately in a triangle [SEGAL1998]. If the rotation were to be applied at triangle level then splits would occur between the triangles of the scene that were previously flush against one another.

This technique is applied per frame on a per vertex basis. Consequently performance is a concern. The technique should be applied in conjunction with view frustum culling, so only a visible triangle's vertices will be modified.

4.1 Using the Technique

It is clear from this example that certain conditions need to be met for the technique to be helpful. Firstly, the data needs to be able to be seen from a plan view. If the data has a roof or covering, context is still going to be obscured. The technique is not applicable to arbitrary 3D data sets, and as with any technique should only be used when the benefits are clear. Secondly, data to be presented in a first person view must be close to the viewpoint. Some data may be distant from the user but not obscured. This data will be distorted, even though it would have been seen without distortion.

In the maze scenario the data is dense and regularly distributed. In a sparsely populated data set, the effect of the technique may be unclear. In these circumstances it may be advantageous to render a 2D grid on the plane of bending. In general, the effect of any technique on data should be directly visualized so that it is clear what is inherent in the data and what is a result of the technique.

The amount of context can be varied by the *threshold* before the bend and the angle of the bend. A steeper and closer bending section decreases the similarity to a normal first person view, but increases the context. This raises the question of user specification of such parameters. Varying the bend threshold interactively may well be an effective way of controlling the amount of context. However other parameters may be less intuitive to manipulate. The type of bend used, whether it is cylindrical or spherical, and the acceleration of rotation (whether the bend is smooth or not) may well change the efficacy of the technique. Allowing changing of these parameters may unnecessarily complicate the interface.

4.2 Applications

Many interactions that are constrained to traversal on a surface are likely to have the context obscured by detail that projects out from the surface. This is particularly prevalent where there is a human presence metaphor being used in the interaction. Furthermore constrained navigation in 3D can be much more effective than free flying as generally only 2D input devices are used. For example Hanson and Wernert [HANSON1997] present a system in which all 3D navigation is constrained to an arbitrary 2D guide manifold, which is more intuitive than free-flying.

Some examples of possible real applications for this technique are the navigation and display of Geographic Information Systems (GIS) data for mining, planning and resource management, and real world navigation aided by augmented reality.

GIS data tends to be very large and planar, and as such would lend itself well to navigation by bending. Augmented reality allows virtual data to be displayed over real vision. A map could be overlayed over the top of real scenery and then distorted in the distance to provide easy navigation. Essentially anywhere the data is mainly on a single plane, and the data in the distance provides information necessary for navigation, is a potential application area.

5. Conclusion

Every task has different requirements, and no one technique can provide all the solutions, but for certain problems the technique presented in this paper can provide context without separate viewpoints. This is particularly useful where distant data is to be used for navigation, and close data needs to be viewed to facilitate local interaction. In applications whose data mimics the real world this technique is most likely to be valuable.

This paper provides an exploration of concept and a guide to the implementation of a novel navigation technique. As with all new techniques, its value depends on the applications found for it. Considering the wealth of first person, virtual human, interactions, there is the potential for quite a number of applications that can benefit from the concepts explored herein.

6. Future Work

The task presented in this paper will be extended and presented to a series of users to gauge the distorting technique's effectiveness. It is expected that timing results for the task, as well as user feedback, should give a good indication of the relative merit of the techniques. Following this, the extension of the technique to a more realistic task will be considered.

Further areas of work raised by this paper stem mainly from the implementation and specification of different kinds of distortions. Arbitrary distortion of certain regions and not others can enable easier navigation in a wider range of applications. However, how these arbitrary distortions are specified, whether by the user or designer, is an untackled problem. In such complicated environments it would most likely be necessary to directly visualize the distortion itself. This may be done in the form of a grid, but other solutions would have to be investigated.

7. References

[HANSON1997] Hanson, A. J. and Wernert., E. Constrained 3D navigation with 2D controllers. In *Proceedings of Visualization '97*, pp. 175-182. IEEE Computer Society Press, 1997.

[KEAHEY1998] Keahey, T.A. The Generalized Detail-In-Context Problem. *Proceedings of the IEEE Symposium on Information Visualization.* IEEE. 1998.

[PAUSCH1995] Pausch, R. Burnette, R. Brockway, D. and Weiblen, M. Navigation and locomotion in virtual worlds via flight into hand-held miniatures. *Proceedings of the 22nd annual ACM conference on Computer graphics*, pp 399 – 400. 1995.

[SEGAL1998] Segal, M and Akeley, K. The OpenGL Graphics System: A Specification (Version 1.2). ftp://sgigate.sgi.com/pub/opengl/ doc/opengl1.2/opengl1.2.pdf). 1998

[STOAKLEY1995] Stoakley, R, Conway M. and Pausch, R. Virtual reality on a WIM: interactive worlds in miniature. *Conference proceedings on Human factors in computing systems.* pp.265 – 272. 1995.

[WINCH2000] Winch, D., Calder, P. and Smith, R. (Focus + Context)3: Distortion-orientated displays in three dimensions. In *Proceedings of the 1st Australasian User Interface Conference AUIC 2000,* Canberra, Australia, Jan-Feb 2000, pp. 126-133.