Solving the Occlusion Problem for Three-dimensional Distortion-oriented Displays

Donovan Winch
School of Informatics & Engineering
The Flinders University of South Australia
winch@ist.flinders.edu.au

Dr. Paul Calder
School of Informatics & Engineering
The Flinders University of South Australia
calder@ist.flinders.edu.au

Dr. Raymond Smith
Gippsland School of Computing and Information Technology
Monash University
ray.smith@infotech.monash.edu.au

Abstract

Recent research into distortion-oriented displays (DODs) and non-linear magnification techniques has considered extending their application to large three-dimensional datasets. Inherent properties of three-dimensional datasets introduce some difficulties that do not occur in 2D environments. This paper considers the Occlusion Problem - that of context data hiding, or occluding, some or all of the data within an area of focus. A novel solution to this problem is proposed, namely the use of non-geometric distortions combined with geometric distortions, producing a three-dimensional distortion-oriented display that reduces obfuscation of data within the area of focus. An implementation of these techniques is developed, and its application to a small number of 3D datasets is examined as proof of concept.

1. Introduction

Large datasets pose the difficulty of having too much detail to be viewed on-screen at once on a standard computer monitor. Any dataset in which there are more data points than the 1600*1200 pixels available on most modern PC desktops can not be displayed in its entirety if the original detail is to be maintained.

Traditional solutions to this problem have been numerous. A common technique is simple zooming, where the user simply zooms in on an area of interest, which enlarges to take up more screen space at the expense of other data. Also common is the concept of 'windowing', where the user has access to a large main window, which shows an area of the dataset in its original detail, and also a smaller overview window, which shows the location of the larger viewport with respect to the rest of the dataset.

1.1. The Focus + Context Problem

These traditional approaches and others like them, however, produce a different, but related problem: that of examining the area of *focus* while also maintaining *context* with the rest of the data. In simple zooming, for example, the user can either zoom in on an area of interest, or zoom out to look at the whole of the dataset in reduced detail. If there are multiple areas of the data that the user wishes to view simultaneously, perhaps for comparative purposes, this cannot be done. Likewise with the windowing interfaces: a user wanting to compare some point of interest in one area of the dataset with another similar point in a separate area of the data must navigate somehow through the rest of the data. Focus and context cannot both be maintained in displays using these approaches. For this reason, this problem has often been referred to as the *Focus + Context Problem* (e.g. [11]).

1.2. Distortion-oriented Displays

The Focus + Context problem is well known, and a number of solutions have been proposed that aim to produce displays in which the whole of the dataset can be viewed in decreased detail on-screen, with one or many regions of interest being displayed at an increased magnification. These techniques can generally be referred to as *non-linear magnification* techniques, and the displays they produce are often dubbed *Distortion-Oriented Displays*, or *DODs*.

Since the development of the first computer-generated distortion-oriented display, most of the research in the area has been directed towards different properties of the geometric distortion and transformation functions that define exactly how they distort the datasets they are applied to. Furthermore, these datasets have been inherently one- or two- dimensional for the most part. Recently, in keeping with the general trend in computer graphics in moving towards applications for visualisation

0-7695-0969-X/01 $10.00 © 2001 IEEE

and manipulation of three-dimensional datasets, some research has been directed towards extending these distortion-oriented displays into three dimensions [4, 11].

1.3. Properties / Problems of Three-Dimensional Displays

With this incursion into the realm of three-dimensional graphics, this new generation of DODs is beginning to encounter some of the problems involved in viewing data that is inherently multi-dimensional on a two-dimensional computer screen. Apart from the obvious difficulties in the sheer volume of data to calculate before display, and the performance deficits resulting from these necessary calculations, other difficulties not previously encountered in two-dimensional displays are arising. Models and methods for displaying three-dimensional data on a two-dimensional display surface in ways that present the data in some sort of expected and natural way are obviously important (after all, a three-dimensional model of a human being that doesn't look like a human wouldn't be very useful). Furthermore, navigation within large three-dimensional datasets using traditional two-dimensional navigation tools (mouse, keyboard, etc.), displayed on a (two-dimensional) computer screen presents useability issues of a greater magnitude than their 2D counterparts. Finally, in a three-dimensional dataset, it is possible (indeed, quite likely, if the dataset is fairly *dense*) that there will be data points between the viewpoint and any particular area of interest to the user of the application. Depending on the nature of the data, the area of interest to the viewer can be *occluded* by these intervening data points. We term this problem the *Occlusion Problem.*

These difficulties with the display, navigation and manipulation of three-dimensional datasets are all important, and must be taken into account when any 3D application is designed. The first problem mentioned, that of performance, is always an issue. As faster hardware and algorithms are developed to enable more rapid calculations and display of the data, larger and more detailed datasets are created. The second problem, that of actually displaying the data on the screen, is addressed by many *application programming interfaces* (APIs) that enable application developers to create and display, somewhat realistically, three-dimensional data objects. Most recent 3D APIs provide the programmer with access to methods for simulating lighting and other advanced effects, which also increase the performance problem. The interface problem is also being addressed in traditional displays, and can be directly extended into the realm of three-dimensional distortion-oriented displays. Specialised 3D input devices, including various gloves, joysticks and 3D mice, have been developed in the past, and further efforts to enhance them still continue. The final issue mentioned above, that of the Occlusion Problem, is the focus of the research described in this paper.

1.4. Paper Outline

The remainder of this paper is structured as follows: Section 2 examines some of the previous work related to past solutions to the Focus + Context problem. In particular, it looks at distortion-oriented displays or non-linear magnification, including recent work in extending the concepts behind these ideas into three dimensions. Section 3 proposes a solution to the Occlusion Problem and examines a practical implementation of it. Finally, Section 4 discusses some conclusions drawn from the current work and suggests possible avenues for further research.

2. Related Work

This section briefly examines some previous work related to the current project. Section 2.1 looks at 'traditional' distortion-oriented displays and techniques, which use geometric distortions to provide the user with a *focus area* of the screen at high magnification, with the remainder of the visualised data displayed at reduced magnification for context. Section 2.2 discusses more recent extensions of this research into the realm of three-dimensional graphical visualisations, also concerned almost exclusively with geometric distortions.

2.1. Traditional Distortion-oriented Displays

The *Bifocal Display*, developed by Spence & Apperley in 1982, is often recognised as the first computerised interactive distortion-oriented display technique [10]. Across the screen are three regions, the centre region being the area of focus with the two side regions providing contextual information. The central region is magnified for close inspection of the data, while the context regions contain suitably de-magnified data. Distortion occurs in only one dimension (across the width of the screen). An extension of the Bifocal Display was proposed by Leung in 1989, in which the display is divided into nine regions [5]. The central region is once again the area of focus, with the eight surrounding regions providing context, de-magnified in either one or both dimensions, depending on their location.

MacKinlay, *et. al.*, developed the *Perspective Wall* in 1991, so named because its appearance is based on what the user would see if the dataset being visualised was mapped onto a wall, with edges leading back into the distance [6]. The focus region is positioned in the centre of the display, with the surrounding contextual data being distorted based on its distance from the focal region. In

1992, based on the earlier *Fisheye View* of Furnas [2] and its descendants, the *Graphical Fisheye* display was developed by Sarkar and Brown [8]. This was the first distortion-oriented display to be widely useable for the visualisation of two-dimensional graphical datasets, as smoothly changing magnifications of sections of the data could be displayed, unlike the customary magnification *jumps* of the earlier displays. Unfortunately, though, the information in the context area tends to become useless at focus point magnifications greater than 10x [9].

Solutions to this and other problems were proposed by Smith [9]. He developed a frustum model of distortion, where the magnified area of focus is displayed on the top plane of the frustum, with the remaining contextual data distorted along the edges of the frustum to the edges of the display screen. The technique allowed focus area magnifications of up to 100x, while still retaining useful contextual information. Furthermore, Smith improved the interactivity of applications, maintaining high frame rates while distorting even very large datasets, by using display re-use and Anderson's *Self-Tuning Architecture* (*STAR*) [1].

Keahey introduced the term *non-linear magnification* to describe the effects common to all of these distortion-oriented displays, and described a general formulation of the *detail-in-context problem* (or *Focus + Context Problem*), along with general-purpose methods for dealing with it [3].

2.2. Extension to the Third Dimension

Recent research has been focused on the extension of these established distortion-oriented display techniques to provide support for three-dimensional visualisations. For example, Keahey investigated the visualisation of sparse multi-dimensional datasets in a three-dimensional space, and how non-linear magnification could be extended to three-dimensional visualisations [4]. Traditional two-dimensional distortion techniques were extended to include calculations for the third dimension, and a useable visualisation was produced for the given, sparse dataset. Keahey recognised that the nature of three-dimensional visualisations inherently produces the possibility of occlusion of some of the data, but found that for the given data the occlusion was minimal. Non-geometric distortion in the form of transparency was applied to the visualisation for the simple purpose of highlighting outliers in the data. Those data points that were abnormal in some way were completely opaque, with the more normal or expected values becoming more transparent.

Recent work by the authors also developed three-dimensional displays similar in look and feel to the Graphical Fisheye displays of Sarkar & Brown [8], but with non-continuous rather than continuous distortion functions as their basis [11]. The application of the *Three-dimensional Polar Fisheye Display* and the *Three-dimensional Cartesian Fisheye Display* to two simple test datasets was examined, and it was proposed that the simple extensions of traditional distortion-oriented display techniques into three dimensions was a useful direction for further research. The datasets examined in both this previous work by the authors, and Keahey's *High-dimensional Cluster Visualization* [4] were relatively simple, however, based only on points (or clusters of points) and vectors in 3D space.

3. Non-geometric Distortions

This section examines the Occlusion Problem in more detail, and offers a proposed solution for use in three-dimensional distortion-oriented displays. An implementation of the proposed methodologies is examined.

3.1. The Occlusion Problem

Using a typical API for three-dimensional graphics programming (such as SGI's OpenGL [7]), an application programmer can position objects anywhere within a three-dimensional cartesian coordinate system. The location of each point in such a *world* is specified in three-dimensional coordinates, *(X, Y, Z)*. The programmer can also assign a position from which the user of the application can view the world (the *viewpoint*), the location of which will typically be dynamic, enabling the user to move around within the world. This viewpoint again has a given location in 3D cartesian coordinates. Finally, the programmer can define the *field of view* for the given *viewport* (the window through which the user looks into the world), which determines how wide an angle will be viewable through that viewport (this is essentially the peripheral vision of the viewport).

With any given combination of viewpoint coordinates, viewport angle and datapoint coordinates, only some of the points or objects in any given three-dimensional dataset will be visible. Depending on the size and density of the data being viewed, this may be the whole dataset, or it may be only a small part of it. Three-dimensional distortion-oriented displays have been developed in order to allow the complete dataset to be viewed through the viewport (e.g. [4], [11]), in much the same way as their two-dimensional counterparts have done for 2D applications in the past (e.g. [8], [9]).

In an interactive visualisation application, the user should be able to move the viewpoint anywhere throughout the world that is created by the representation of the dataset, so that the data can be viewed from anywhere within the coordinate space. Furthermore, the viewport should be able to be rotated throughout any angle on any of the three cartesian axes, allowing any point in the coordinate space to be viewed from any location within it. Even with a small dataset made up only of points in the world, it is easy to see that looking at the dataset from any one of many possible positions and angles would cause some of the data points to be hidden behind others.

The problem quickly becomes severe when large datasets are viewed, and when the worlds created from those datasets are made up of larger geometric primitives than simple points in space, such as lines and polygons. Modern 3D-accelerated computer graphics hardware is extremely adept at handling the calculations required to display polygons on the screen, and solid-looking three-dimensional objects are typically created from surfaces made of such polygons (mostly triangles). Figure 1 shows three separate views of a simple modelled human head surface dataset, demonstrating clearly the severity of the occlusion problem in such a situation.

Figure 1. Three views of a three-dimensional, hollow head model, with a teapot positioned neatly inside. The first view shows the solid face with the teapot completely hidden. The second view shows scant traces of teapot visible at the back of the head, with the remainder occluded by the face. The third view, from behind and slightly below the head, shows the whole teapot clearly visible inside the hollow head, through the opening at the back. (Note: The head model was originally acquired from 3dcafe.com before subsequent modification.)

3.2. The Occlusion Problem in Distortion-oriented Displays

DODs that use a non-continuous geometric magnification function for distorting the data typically have a region which is magnified, or viewed in its original detail, upon which the user's attention is presumably focused. In DODs based on continuous magnification functions, this *focus region* is reduced to a *focal point*, at which the magnification is maximal. These points or regions of focus are what the user is assumed to be most concerned with, the remainder of the screen display serving to maintain context with the remainder of the dataset.

The occlusion problem thus has special significance with respect to three-dimensional distortion-oriented displays. If the point or region of focus is some distance away from the viewpoint's position within the 3D world created by the data, then there is an opportunity for other data objects to be positioned between the viewpoint and the area of focus. These data objects, which are displayed only for the sake of maintaining context, can thus occlude part or all of the area of focus, detracting severely from the usefulness of the display. This problem is demonstrated in the central image of Figure 1, above. If the focal region is centred on the teapot inside the head, the context area objects (the face/head in this example) occlude the focal region from many angles.

3.3. A Solution to the Occlusion Problem

A number of further geometric distortions are possible in an attempt to reduce or eliminate the problem of context area data objects occluding focus area data objects. Unfortunately, however, if the distortion function in one section of the context area (that which is likely to occlude the area of focus) is inconsistent with that in the rest of the display, the value of the contextual information that can be gained from the context area is reduced, as it will no longer be consistent across the display. In particular, it is difficult to maintain geometric *feature matching* across the boundary between the focus and context areas, if one section of the context area must be distorted more than another section.

However, the application of non-geometric distortion techniques to properties of the data visualisation has not previously been investigated. APIs for three-dimensional graphics programming provide the developer with the opportunity to specify colour, material properties (reflectance to different types of lighting, emission, etc.), transparency and display method (solid/wireframe) separately for each geometric primitive within the world. Furthermore, these properties can be dynamically altered, as, for example, the position and direction of the viewport change. If these properties of each data object are made dependent upon the location of the data object with respect to the focus area and the position and orientation of the viewport, then a useful three-dimensional visualisation can be generated with distortions based on non-geometric properties of the data. By distorting some of these properties based on the location of the data in comparison with the viewport and focus region positions, a DOD can be developed in which there is no occlusion of focus area objects by those objects outside of the focus area.

3.4. Occlusion Culling

The simplest method for distorting the context area to remove its obfuscating effect upon the focus area is to simply remove those objects from the display entirely. This technique is termed *Occlusion Culling*. Obviously this is not a continuous distortion function, as it has only two states, displayed or not-displayed. Nevertheless, it can be usefully used within three-dimensional distortion-oriented displays to completely remove the effects of the Occlusion Problem from the visualisation. An example of the use of this technique when applied to the simple head dataset is given in Figure 2.

3.5. Wireframe Conversion

Rather than removing polygons that occlude the data within the focus region, we can render them as wire-frame objects instead, and thus still retain some information about their general size and shape. This *Wireframe Conversion* method is again dichotomous, as was the Occlusion Culling method above, as context area objects are rendered as either solid or wire-frame polygons. Depending on the dataset being visualised, the wire-frame objects may still occlude important parts of the data objects inside the focus region, especially in cases where the width of the lines in the wire-frame model is relatively large with respect to the focus region objects. However, this technique can be useful for retaining some useful context with the data in front of the focus region, while still allowing some visualisation of the focus area data objects. Figure 3 shows an example of this technique when applied to the head dataset.

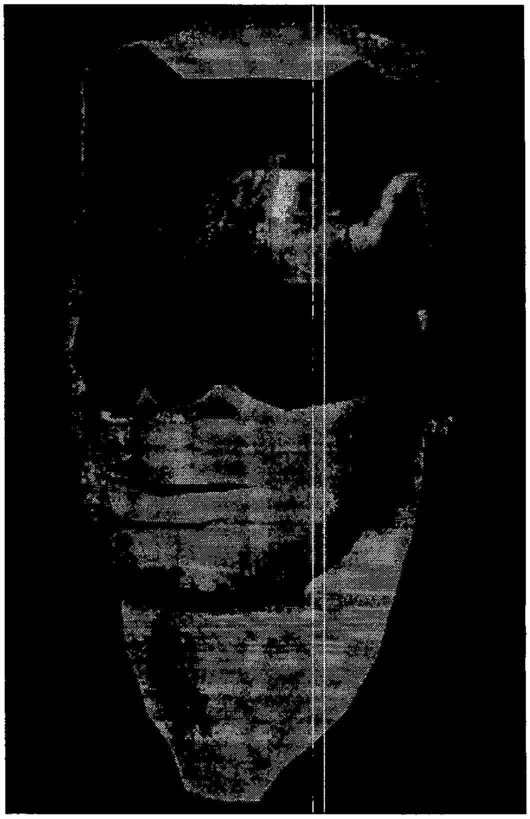

Figure 2. Culling of the polygons occluding the area around the focus region allows the teapot inside the head to be seen.

An obvious extension to the Wireframe Conversion and Occlusion Culling techniques is to combine both methods to produce a three-value non-continuous distortion function. Objects close to the centre of the region obscuring the focus area are completely removed from the visualisation, while objects slightly further away, but still impinging on part of the focus area are merely drawn in wire-frame. In this way, the benefits of both methods are at least partially retained, while the disadvantages of both are reduced. Figure 4 presents the head dataset again, this time zoomed-in on the focal region, with the combined techniques applied.

Figure 3. Conversion of occluding polygons to wire-frame objects allows the teapot inside the focus region to be seen, while retaining some context with the data objects of the face.

3.6. Variable Transparency

Many APIs for programming with three-dimensional graphics include a 32-bit colour mode, which allows the specification of eight bits each of red, green and blue tints to colour each pixel, as well as an eight-bit *alpha* value that specifies how the final colour of each pixel on the screen will be affected by the different colours of the objects *underneath* that pixel in the three-dimensional space represented on the screen. This alpha value can be used to produce various effects that can make visualisations look more realistic, including *transparency* effects to simulate objects made of glass or other transparent materials.

If the transparency of data objects within a 3D DOD is based on their position relative to the viewport and a specific area of focus, a visualisation can be produced in which those context area objects that would otherwise have occluded the focus area are instead transparent. The degree of transparency can be dynamically altered, allowing the user to place more or less emphasis on either the partially occluded objects inside the focus area (making them more visible), or the semi-transparent context area objects (thus hiding the focus area more). This functionality allows for a great deal of freedom with the visualisation, as seen in Figure 5, in which the focus area is only slightly occluded, and Figure 6, in which it is significantly more occluded.

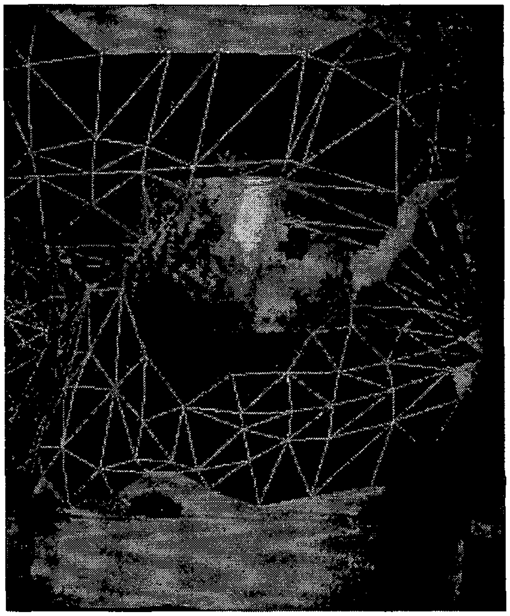

Figure 4. Wire-frame conversion combined with occlusion culling allows better observation of the teapot, while still retaining some context.

This non-geometric distortion function can then be extended to produce a distortion that is continuous across some of the display, by associating the transparency of the occluding objects with the distance from the centre of the area of occlusion. In this way, a visualisation is produced in which those objects occluding the central region of the focus area are almost totally transparent, but the degree of transparency decreases (*opacity* increases) for objects further away. Figure 7 shows the head dataset once again, with the transparency of the occluding context area objects inversely proportional to their distance from the centre of the occluded area.

Finally, further interesting and potentially useful visualisations can be created from combining the transparency distortions with the culling and/or wire-frame distortions of the two previous sections. Figure 8 shows just one possible configuration, with wire-frame rendering using a semi-continuous transparency function for the rendering of the polygon outlines. Many other configurations of these and other distortion techniques are possible, based on the properties of the particular dataset and the desired visualisation.

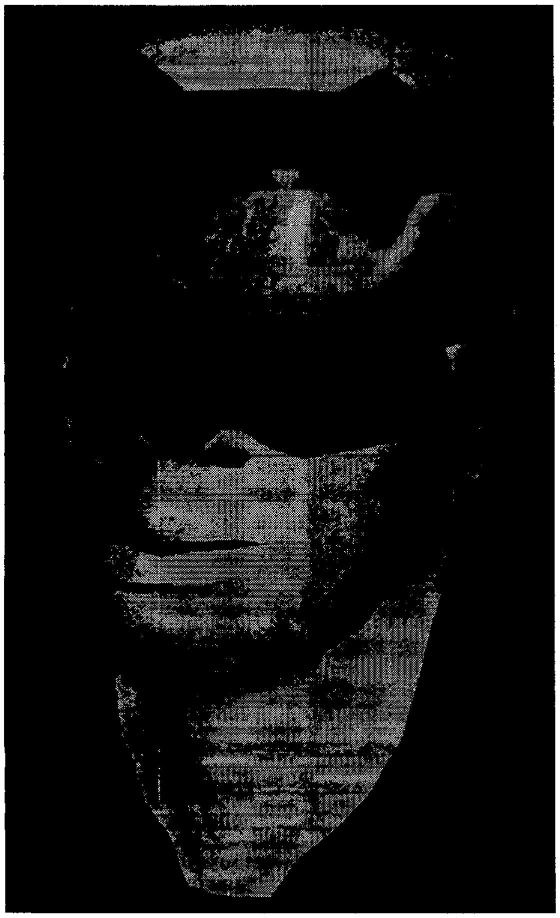

Figure 5. The head dataset with high transparency.

4. Conclusions & Future Work

It has been shown that non-geometric distortions can be used in three-dimensional distortion-oriented displays to reduce or eliminate the problem of occlusion of focus area data by data in the context region. These non-geometric distortions can further be utilised in DODs outside of their application to the occlusion problem, to produce some interesting and potentially useful visualisations, beyond those proposed by Keahey [4]. Further research into non-geometric distortions is underway, including investigations into the further use of alpha values for effects other than transparency, lighting, and material properties of the visualised data objects.

The research described in this paper investigated the use of non-geometric distortions in 3D DODs with only one area of focus, and tied the use of these new distortions to that of the geometric focus area to solve the Occlusion Problem. Ongoing research is examining how these and other non-geometric distortions might usefully interact with multiple geometric distortion focus areas. A three-dimensional distortion-oriented display with multiple independent points of focus, both geometric and non-geometric, is being developed to demonstrate these interactions.

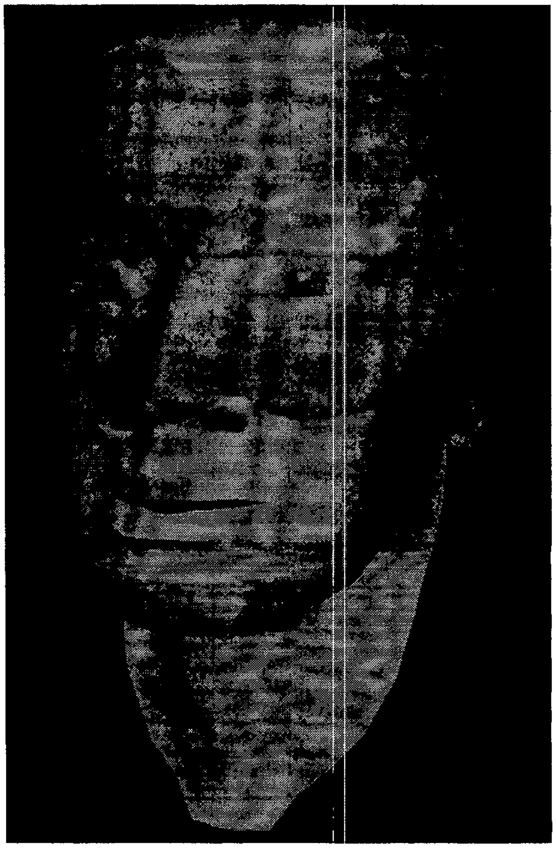

Figure 6. The head dataset with low transparency (or high opacity).

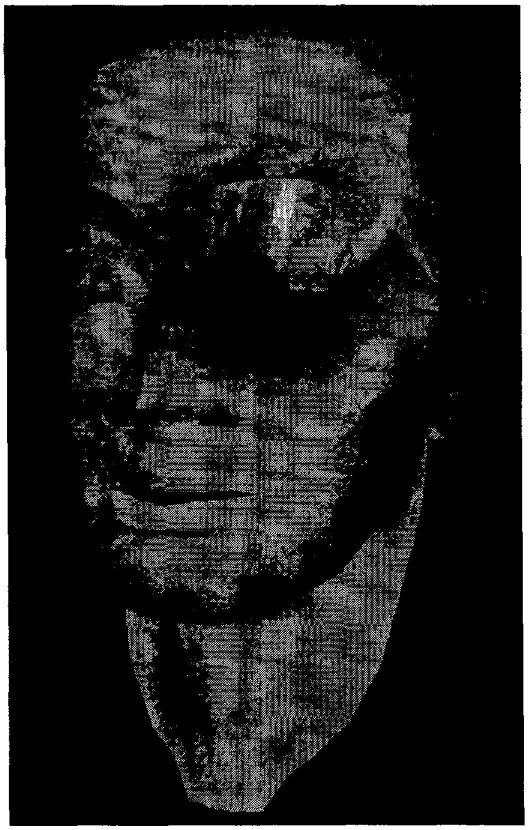

Figure 7. The transparency of occluding polygons is inversely proportional to their distance from the centre of focus.

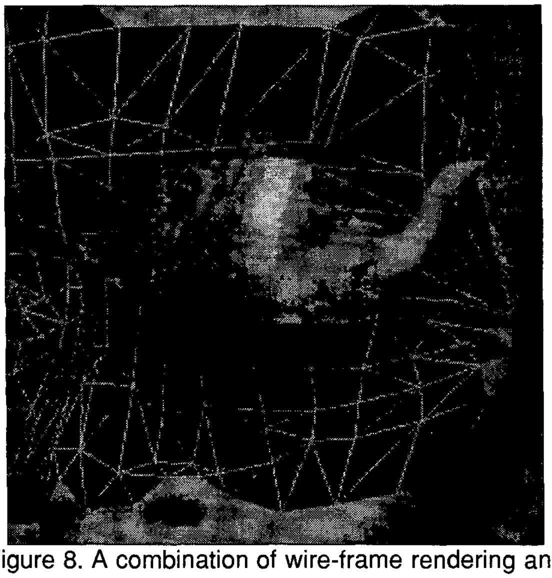

Figure 8. A combination of wire-frame rendering and transparency.

5. References

[1] Anderson, P., Smith, R. and Zhang, Z. (1995). STAR: A general architecture for the support of Distortion Oriented Displays. In *Proceedings of the KDD'95 conference*, Montreal Canada, August 1995, pp. 15-20.

[2] Furnas, G.W. (1986). Generalized fisheye views. In *Proceedings of CHI'86*. ACM, New York, pp. 16-23.

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

[4] Keahey, T.A. (1999). Visualization of High-Dimensional Clusters Using Nonlinear Magnification. In *Visual Data Exploration and Analysis VI*, Vol 3643 of SPIE, Jan 1999.

[5] Leung, Y.K. (1989). Human-computer interface techniques for map based diagrams. In *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*. G. Salvendy and M.J. Smith (eds), Elsevier Science Publishers Amsterdam, pp. 361-368.

[6] MacKinlay, J.D., Robertson, G.G. and Card, S.K. (1991). The perspective wall: Detail and context smoothly integrated. In *Proceedings of CHI'91*, ACM, New York, pp. 173-179.

[7] OpenGL Architecture Review Board (1993). *The OpenGL Reference Manual*. Addison-Wesley Publishing Co, Mass.

[8] Sarkar, M. and Brown, M.H. (1992). Graphical fisheye views of graphs. *Proceedings of ACM CHI'92*. 3-7 May 1992, pp. 83-91.

[9] Smith, R. (1997). *Distortion oriented displays for demanding applications*. PhD Thesis, Gippsland School of Computing and Information Technology, Monash University.

[10] Spence, R. and Apperley, M. (1982). Database navigation: an office environment for the professional. *Behaviour and Information Technology*, pp. 43-54.

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