Visiview: A system for the visualization of multi-dimensional data

Shaun Bangay

Computer Science Department, Rhodes University
PO Box 94, Grahamstown, 6140, South Africa

ABSTRACT

Results generated by simulation of computer systems are often presented as a multi-dimensional data set, where the number of dimensions may be greater than 4 if sufficient system parameters are modelled. This paper describes a visualization system intended to assist in understanding the relationship between, and effect upon system behaviour of, the different values of the system parameters.

The system is applied to data that cannot be represented using a mesh or isosurface representation, and in general can only be represented as a cloud of points. The use of stereoscopic rendering and rapid interaction with the data are compared with regard to their value in providing insight into the nature of the data.

A number of techniques are implemented for displaying projections of the data set with up to 7 dimensions, and for allowing intuitive manipulation of the remaining dimensions. In this way the effect of changes in one variable in the presence of a number of others can be explored.

The use of these techniques, when applied to data from computer system simulation, results in an intuitive understanding of the effects of the system parameters on system behaviour.

Keywords: multi-dimensional, simulation, visualization

1. INTRODUCTION

Analysis of the performance of parallel and distributed virtual reality systems results in expressions relating performance metrics (frame rate, latency) to system parameters (computation and rendering time, network bandwidth, numbers of processors, etc.). Recent work on models of such systems[1] reveals some characteristics of the behaviour of such systems. For a range of the system parameters, performance is usually governed by a single expression dependent on the values of these parameters. These performance functions are usually continuous, allowing the visual identification of the region covered by a particular expression to be easily accomplished. The functional relationships are often easily recognizable, particularly since they are often linear combinations of the parameters. However they are not always amenable to automated analysis because they may involve unusual expressions such as that of the Highest Common Factor (HCF) function. Such functions can, in many cases, be identified visually by an experienced observer.

The primary data type to be visualized for the purposes of this paper are simulation results; performance characteristics of a system measured across a range of system parameters. The purpose of the visualization is to deduce the relationships that exist between the system parameters and the corresponding performance characteristics. The independent variables (system parameters) in this case are sampled at regular intervals and the dependent variables (performance values) are measured after simulation with these parameters. The purpose of the visualization is to assist in determining the nature of the relationship between the dependent and the independent variables.

The specific nature of this data is that it is multi-dimensional, due to the large number of possible system parameters that may be used as independent variables. For the purposes of this discussion multi-dimensional will be assumed to refer to data sets with three or more dimensions. The problem of visualization becomes particularly intriguing as the number of dimensions increases. An application which addresses this problem is the subject of this paper.

Email: cssb@cs.ru.ac.za

Depicting explicit connections between adjacent points can be misleading, or inappropriate at times. This is particularly true when applied to data with no physical interpretation. The geometric structure of the projection of the data set being rendered may not always be meaningful, and no additional relationships should be implied. No relationship between points may be introduced by the software. This discussion is thus limited to the visualization of multi-dimensional, unconnected data points.

While this document concentrates on visualizing simulation results, the application is by no means limited to this type of data only. The tool is used for a number of other purposes: visualization of mathematical functions for educational purposes, examination of medical data (extracts from CAT scans) and visualization of digital elevation data from satellite data sets.

The emphasis is on rapid intuitive understanding of the data set being examined. Thus the approach concentrates on showing the relationship between the points in the data set. Less emphasis is placed on realism of output and on data manipulation.

The techniques for visualization of data are discussed in the following sections, categorized by the complexity of data to which they are suitable. Some terminology specific to this document is defined as follows:

Definition: Dimension. Assuming the input data consists of a collection of k records, each record consisting of an n-tuple: $\{x_{1j}, x_{2j}, \ldots, x_{nj}\}, 1 \leq j \leq k$, the i^{th} dimension is the list $\{x_{ij} | \forall j \in \{1 \ldots k\}\}$. This is equivalent to the i^{th} column of a data file where the tuples are stored one per line.

Definition: Spatial Axis. The term spatial axis refers to one of the physical ranges used to represent the elements of a dimension visually. The spatial axes used in this paper are listed in Table 1.

2. MULTI-DIMENSIONAL VISUALIZATION

Scientific visualization is usually applied to data relating to physical phenomena, based on three spatial dimensions. Even including support for measurements taken over a time interval and for representing the value of an attribute at each point, the total number of dimensions is usually bounded at five. Visualization packages, such as KHOROS[2] and VIS5D,[3] provide comprehensive support for the manipulation and visualization of data sets that fall into this category.

Data mining and querying of multi-dimensional databases has produced a number of approaches to providing a visual representation of multi-dimensional data. These include the display of query relevance factors,[4] worlds within worlds[5] and parallel coordinates.[6] While many of these techniques require that the user understand the principles behind the representation, the visual appearance of the data provides an intuitive understanding of the nature of the data set. For example, when using parallel coordinates, there is a duality between the representation and the structure of the original data.[7] These approaches are homogeneous in the sense that scaling to more dimensions involves use of the same representation, but with increasing complexity.

The approach taken in this paper is to make use of a geometrical representation. Each parameter is mapped onto a spatial axis of some kind. The nature of these axes will be explored in the sections following. The final representation of the data will be a heterogenous mix of these different spatial axes, with the addition of each additional dimension adding only a limited amount of additional complexity. There is a limit to the number of spatial axes available, after which a more general technique must be applied to control additional dimensions. Limiting the onset of the complexity explosion allows simplified multi-dimensional visualization of data that would otherwise require the use of a more complex representation. The use of a spatial representation allows the physical shape to play a role in the identification of the data, whereas more abstract techniques do not always provide this correspondence.

Many of the techniques described rely on certain properties of the data set. In practice, finite data sets mean that the set of values in a dimension is finite. In many cases, the values in a dimension are integers (representing, for example, the number of processors). Some dimensions are related, where they represent the independent and dependent variables of a function.

3. VISIVIEW: THE APPLICATION

While the actual implementation of the visualization system is not of particular relevance to the discussion, there are several points with respect to the application which are relevant to the discussion of multi-dimensional visualization techniques in later sections. This section examines the principles under which the user interface is implemented. The details of the visualization mechanisms are given in sections 4, 5 and 6.

3.1. Data representation

The data files for the application are normal ASCII text files, containing multiple columns of space delimited data. This format is convenient for accepting data from most applications. Inclusion of a simple header in the file allows the system to associate a meaningful name with each column. Each column, the equivalent of a dimension, can be assigned to one or more of the spatial axes corresponding to the various visual representations.

The level at which data becomes interesting in terms of visualization is when it involves determining the relationship between three or more variables. Relationships between two variables can be examined using the large number of graphing techniques already commonly employed in everyday use. Three (and higher) dimensional visualization techniques become interesting because the amount of effort required to manually construct a representation of the data set can become excessive.

The nature of the data, specifically that there is no relationship between the points (other than that to be determined with the help of the visualization) means that it would be misleading to attempt to use a graphical representation that may imply such a relationship. The use of lines, polygons or splines connecting the points is inappropriate. A data point can best be represented by a single pixel. No hidden surface removal is required for monochrome pixels. This low overhead proves to be extremely convenient by allowing the application to provide reasonable visualization performance for large data sets, even on systems with limited graphical acceleration.

The software automatically calculates the bounds on the coordinates being used and creates the bounding box surrounding the entire dataset. This provides a visual reference outlining all points. Some optical illusions (such as confused depth perception) are removed in this way, as the user can rely on the knowledge that all points are within the bounding cube.

The software also allows automatic scaling of all values so that all sides of the bounding cube will be equal in length. This has the obvious benefit of making maximum use of the available viewing area. This has little influence on the effectiveness of visualization for datasets in which the units of measurement for each variable are independent, or where a three dimensional representation is uncommon. The distortion in the aspect ratio is easily noticed when viewing familiar objects, or objects where the different variables share a common scale. Generally, in these latter cases, the ranges of the different variables are comparable, and so a common scaling factor can readily be used.

3.2. GUI layout

The layout of the GUI provides a sequence of controls for the different visualization techniques in order of increasing complexity, as shown in Figure 1. A user can restrict himself to a given level of complexity by ignoring controls beyond a certain point. All controls can be used at any time or in any combination. The ordering is for convenience only.

The leftmost window provides the visual representation of the data. The user can interact with the representation displayed within the two dimensional window. Translating and rotating points about the three cartesian axes is achieved by moving the mouse. The six degrees of freedom required for three dimensional interaction are provided by combining the two degrees of freedom of the mouse with use of its three buttons. Translation along X and Y axis are combined on a single button, with rotations about these axes linked to another button. This allows a fairly intuitive correspondence between direction of mouse movement and corresponding motion. Translation along, and rotation about, the Z axis are activated by the third button.

The motion controls are duplicated on the side of the application window. These controls are intended only for novice use. The distraction of having to focus on controls while moving the object detracts from the use of the movement for purposes of visualization.

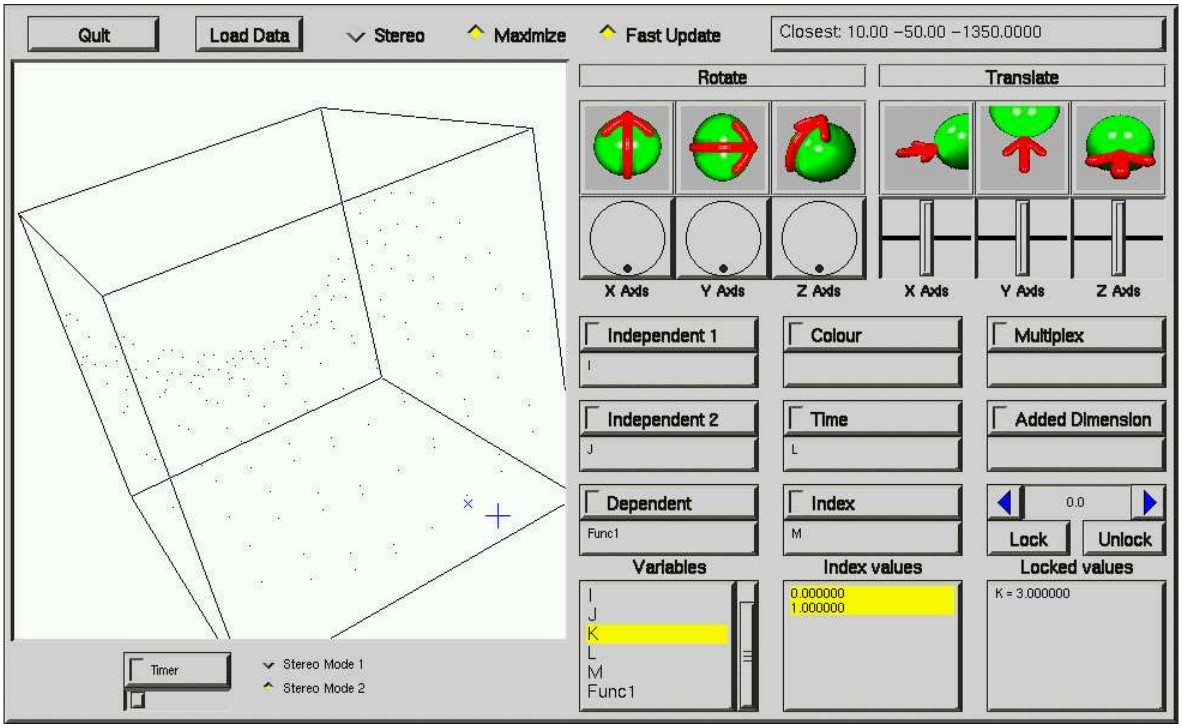

Figure 1. View of the Visiview application

4. TECHNIQUES FOR RENDERING THREE DIMENSIONS

The view of the data set provided by the window on the display is inherently two dimensional. Stereoscopic viewing and real time interaction are used to provide the effect of a third dimension.

4.1. Stereoscopic viewing

An increasingly common feature in visualization systems is the use of stereoscopic viewing, in which rendered images are presented separately to each eye. The additional data provided is enough to obtain an indication of the depth of each point.

Mechanisms for stereoscopic viewing range from Head Mounted Displays, with a display for each eye, to shutter glasses, which time-multiplex images from a single display, to low-end devices such as red/green filter glasses and split screen views.

The Visiview application supports red/green filter glasses, as well as shutter glasses. The former are cheaper and useful for group work, the latter allow the use of colour as a spatial dimension.

Stereoscopic rendering adds little overhead computationally, apart from the need to render each image twice from two slightly different viewpoints. A small change in the perspective transformation is required. Two variations of the stereoscopic projection are implemented. The variation derived by Hodges[8] involves only translation and projection, while that described in McReynolds and Blythe[9] includes a rotation to fuse images at a depth corresponding to the projection plane. While both are effective, subjects rarely express a preference for one, and neither technique is favoured significantly more than the other.

Stereoscopic viewing provides the expected additional depth perception required for viewing three dimensional data sets. It is less effective than was initially expected; the depth effect is not always immediately apparent, and some degree of optical illusion is still present. Combined with movement, however, the limitations are quickly overcome and the advantage of extra depth information outweighs the cost of greater rendering time.

4.2. Interaction

Static views of a three dimensional scene viewed on a two dimensional display device are prone to optical illusion. The most common effect is the apparent inversion of the depth axis. Even a stereoscopic image of the same scene can produce these effects. With the rendering technique used in this case (cloud of points), it is often difficult to achieve the initial visual association of points required to determine depth information.

This problem is most effectively overcome by movement of the data set, particularly by rotation about the centre point of the data. This movement quickly distinguishes points on opposite sides of the data set as they move in different directions. The effect of movement combined with perspective projection also provides additional depth cues because the amount of movement differs with depth.

In systems such as the STEREO-SCOPIC EDITOR,[10] which employs stereoscopic viewing, but in which movement is slow and stilted, the three dimensional nature of the data being edited is far less intuitive. Updating of each frame takes considerable time, and only occurs after an explicit mouse click. This limits the update rate to the speed of rapid finger movement, usually far below the limit required for smooth animation. This approach can be compared to that used in Visiview, where the image is continuously updated as long as mouse movement occurs. This latter approach also allows an intuitive feeling for the relationship between the direction of mouse movement and the corresponding change in viewpoint.

The mouse based control can provide sufficient movement for effective visualization through use of the rotations controlled by the first mouse button only. Thus it can easily be used, even by people with only a few seconds of instruction.

5. TECHNIQUES FOR ADDING A SINGLE ADDITIONAL DIMENSION

A number of additional techniques are available to allow for the visualization of further dimensions. The techniques described in this section each add a single dimension to those already visualized. They can be combined to provide additional effect, but each cannot be used more than once.

5.1. Attributes

Most scientific visualization systems allow for representation of an attribute of each point in the data set, usually represented through the addition of colour to the rendered representation.

This principle is applied to the Visiview application as well. One dimension can be chosen to be represented as an attribute attached to each data point. Since the number of colours that may be represented or perceived is limited, this view is best applied to a dimension with a limited range, and evenly distributed values. As may be expected, this option is only applicable when using shutter glasses to provide for the stereo view.

Adding a dimension through the use of an attribute at a point can be extended to other attributes of a point. Setting the point size is an alternative, although this can be difficult to implement in conjunction with any projection that introduces a decrease in size with apparent distance.

Using a physical representation attached to the point such as a shape for the point, or some vector originating from the point is another alternative. Attribute data is represented as glyphs attached to points in the visualization of multi-dimensional biological data in Chi *et al.*[11] The worlds within worlds approach[5] is based on this concept, recursively attaching complete representations of remaining dimensions at selected points.

5.2. Time

Conventional scientific visualization systems have traditionally used an animation sequence to represent data collected over a period of time. This concept can be extended to allow any dimension to be mapped onto a time axis.

The animation sequence can be set to play continuously and automatically. It proves useful on occasion to allow the user to stop the sequence and step through single frames to observe finer detail.

5.3. Multiplexed dimensions

In cases where one dimension has a very limited range it is possible to effectively combine two dimensions, A and B into a single dimension C. If dimension A has a set of k distinct values, which can be enumerated by a sequence of integers, $0, 1, 2, \ldots k$, then any pair of values (a, b), $a \in A, b \in B$ from these two dimensions can be multiplexed together to represent a value in the new dimension C using the following process:

> Let i be the integer, $0 \leq i < k$ corresponding to the value a from dimension A, and let r represent the range of values from dimension B (difference between maximum and minimum values). The value for the new dimension, c, can be found using: $c = b + i \times r$.

Visually this corresponds to a repeated tiling of dimension B for each value in A. This technique is most applicable if dimension A consists of regularly spaced values, preferably integers. In principle, the result of this multiplexing can be represented on any of the spatial axes. To simplify the rendered image, application of the technique has been restricted to the X axis in the Visiview application. This tiling process is most effective if the number of repetitions is quite small; on the order of five or less.

5.4. Index

In many cases, a set of three dimensional data points is the result of diagramming a function and consists of two independent variables and the value of the function for each pair of variable values. As such, the data set represents a surface, an implicitly two dimensional structure. Even in more general situations, the data presented for visualization is of a dimension one less than that implied by the number of coordinates for each point.

This implies that four dimensional data can be represented in three dimensions as a set of surfaces. Often with well behaved data sets, these turn out to be nested or non overlapping surfaces. An index field can be used to control the visibility of each surface. As implemented in Visiview, individual surfaces can be turned on or off, or surfaces can be selected in contiguous groups, as ordered by the values of the dimension used as an index field.

6. TECHNIQUES FOR ADDING MULTIPLE DIMENSIONS

At some point, the number of dimensions in the data becomes larger than can be catered for by the combination of the techniques presented thus far. Some mechanism to control and examine the effects of the remaining dimensions is required.

The mechanism used is similar to the indexing system described previously. The user must select a set of dimensions for control under this mechanism. These dimensions are referred to as *added* dimensions. One of these dimensions is chosen to represent the *active* added dimension. Controls allow the selection of a particular value from this dimension. Once a value is selected, the display is limited to the records containing the corresponding value in that dimension field.

The user then has a choice of actions. The active dimension can be discarded, an alternative value chosen, or the value locked. The latter case is the most interesting and is described further. Lock entries associate the dimension and the chosen value, and filter out only those records that have all the appropriate values for every locked dimension. Additional dimensions can be chosen as the active dimension and also locked. A lock field may be selected and unlocked, in which case it becomes the active dimension. The process of selecting an active dimension, choosing a value, locking it and unlocking another entry and again changing the value can be likened to the action of translation down one axis, and then turning perpendicularly and translating down another.

There is no limit in principle to the number of dimensions which can be managed in this way. The process is limited by the need to keep track of which "direction" one is currently moving. This process is considerably less intuitive than that involved in any of the previously described techniques. It is also difficult to examine the relationships between the added dimensions. In practice, many users find this mechanism convenient to remove the influence of extra dimensions while applying the other techniques to a limited subset.

7. PROJECTIONS

After all the effort invested in providing support for as many independent dimensions as possible, it would seem incongruous to consider the cases where a single dimensions is applied to more than one spatial axis. It is of interest that the techniques developed in the preceding sections can be applied to data with three or less dimensions and provide additional insight.

Use of the first three spatial axes allows the user to visualize conventional three dimensional data. Since a dimension can be associated with more than one spatial axis, the system can be used to view two (and one) dimensional data as well. This has interesting implications for two-dimensional visualization, as the use of the three dimensional environment allows viewpoints which are not commonly associated with this category of data.

Duplication of variables applied to a three dimensional data set is an easy way to produce a projection of the data down the axis corresponding to the variable that is lost. This has significant application with some cases, apart from the obvious one of converting three dimensional objects into instant roadkill. The projected form in some cases allows visualization of point density information.

Using the same variable for one of the spatial axes, as well as for colour control creates the effect of draping a height map over the data set. The data set can then be projected onto two dimensions creating an approximation of a contour map.

Time based animation, combined with projection allows tracing out of profiles of the data. This effect can also be achieved using indexing, which allows two or more profiles to be displayed simultaneously and allows finer control over the placement of the profile.

8. PERFORMANCE ISSUES

This section discusses the performance of the Visiview application with regard to time and space complexity of the techniques used.

The time performance of the system will be considered in terms of M, which represents the number of values in each of the N dimensions of the data set: thus the data set can have up to M^N points.

The basic three dimensional rendering overhead has a time cost of $O(M^N)$ for each frame, since it must iterate through each point and render it appropriately. Providing a colour value for each point involves a simple, constant overhead for each point, creating no additional complexity. Multiplexing requires an additional simple calculation (see section 5.3) at each point, which does not affect the overall form of the time complexity.

Techniques such as the use of time animation, indexing and added variables decrease the cost of rendering. These techniques filter out some of the points that need to be displayed for each frame. The rendering cycle can be reduced to traversing a smaller list of data points by performing a certain amount of preprocessing when the dimensions to be used with these techniques are chosen. For time animation and indexing, the number of points is reduced by a factor M. For active variables it is reduced by M^{K+1} where K is the number of locked variables, and assuming the presence of a single active added variable. The data structures created during the pre-processing step are comparable in size to those required to store the data set.

The overall time complexity of rendering the scene, given that all these techniques are being used, is: $O(M^{N-(K+3)})$.

9. ASSESSMENT

Visualization of a set of three dimensional data points is most effective when both stereoscopic viewing and real time interaction are used. Each technique provides less satisfactory results when used on its own. Addition of some extra visual cues also assists in visualizing the data effectively. A more comprehensive study in Ware and Franck[12] explores the relative value of each approach in greater depth. Their result found that both factors help with understanding and that the type of motion involved is not important. This result was confirmed during tests with the Visiview application. In particular, the use of static stereo images is not sufficient to provide an intuitive understanding of the nature of the data being represented.

This section describes some of the findings after an assessment of the Visiview application, and the visualization strategies provided. These tests are by no means comprehensive: the subjects are selected from a small group specializing in work in this area, and the range of experiments is limited to a few simple comparisons. The results describe issues relating to the use of the different dimensional visualization techniques.

The tests applied are as follows:

1. Identify the type of function $f(x,y)$ where the dependent variable is assigned to one of the following spatial axes: z, *colour*, *time*. The association between function and axis is varied for each experiment. The test functions are:
   (a) a two dimensional *sinc* (Mexican hat) - $f(x,y) = \frac{sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$
   (b) a hyperbolic paraboloid - $f(x,y) = x^2 - y^2$
   (c) a cone - $f(x,y) = \sqrt{x^2+y^2}$
2. Identify the type of function $f(x,y,c)$ where the dependent variable is assigned to z and the variable c is assigned one of the spatial axes: *colour*, *time*, *multiplex*. The functions used are:
   (a) a sequence of nested spheres - $f(x,y,c) = \begin{cases} \sqrt{49c^2 - (x^2+y^2)} & \text{if } 49c^2 - (x^2+y^2) > 0 \\ 0 & \text{otherwise} \end{cases}$
   (b) a collection of two dimensional *sinc* functions of different frequency - $f(x,y,c) = \frac{sin(l)}{l}$, where $l = \frac{\sqrt{x^2+y^2}}{(3+2c)}$
   (c) a set of planes with changing attitude - $f(x,y,c) = 3cx - 2cy$
3. Identify points on the boundary between regions of the function $max(f_1, f_2, f_3)$ where:
   (a) $f_1(x,y) = 20x - 5y + 1750$
   (b) $f_2(x,y) = 2y + 1590$
   (c) $f_3(x,y) = x^2 + y^2$
4. Determine the functional dependence with respect to each variable of a function $f(x,y,z,c,k)$. The subject is allowed to proceed at his own rate, and choose his own visualization strategy. The function used is $f(x,y,z,c,k) = x(x+20k) + z^2 - y^2 + 50c$. To simplify the use of some spatial axes, the variable c is limited to the values $\{0,1,2,3,4\}$ and the variable k to the values $\{0,1\}$.

Limited training in the use of the tool is given. The scenarios for each test are set up beforehand, and assistance in the use of the tool is provided during the experiment where required.

Recognition of functions from test 1 is easily and accurately accomplished when they are displayed using the z axis. In most cases the functions can also be identified using the *time* and *colour* spatial axes, although this takes, on average, about four times longer. This result may not be truly representative of use of the *time* axis, as most of the delay is in the animation tracing out the complete shape of the function.

The foray into the fourth dimension shows the use of time based animation to be the most successful technique in this case. The use of multiplexing is next best, taking about twice as long to achieve results. The colour axis is least successful, which can be attributed to the nature of the test data. The physical overlap between the functions for the various values of c, produces a confusing image which requires careful selection of viewpoint to isolate the characteristics of the function. The relative success of the each technique appears to be related to the number of points presented in physical proximity to each other.

Identification of regions, one of the design goals of Visiview, is extremely effective. It takes on average about 30 seconds to find the various regions, which is spent changing the viewpoint until the nature of the function is clear. Once identified, points on the boundary can be traced out without hesitation.

Technique	Suggested dimension	Spatial Axis
Display window X Axis	1	Screen width (x)
Display window Y Axis	2	Screen height (y)
Stereoscopic view and Interaction	3	Stereoscopic vision (z)
Attributes	4	Colour
Time	5	Time
Multiplexed dimensions	6	X axis
Index	7	Browser selection
Added dimensions	$\geq$8	Slider selection

Table 1. Summary of visualization strategies

The use of the tool for practical identification of a multi-dimensional function yields particularly interesting results. Initially the subjects are allowed to use the tool with exposure only to the spatial axes used in the tests above. During the test, the use of the remaining controls is explained. The initial demonstration leaves a number of the spatial axes active before allowing the subject to continue. Most subjects immediately reset the system back to three dimensions. Some then choose a favourite spatial dimension (usually time or colour) and examine the nature of the functional dependence for the various variables using that axis. Others, particularly those with a mathematical or scientific background, latch onto the use of the added dimensions, and used these to lock down all but two of the independent variables. Surprisingly many subjects manage to produce reasonable results, with some managing to produce reasonable or accurate dependencies for all five independent variables. This is significant considering that it occurs during the first exposure to the tool.

Visiview has been applied to real problems, of the type described in section 1. It has been successfully applied to the visualization of multi-dimensional simulation data, albeit by an experienced user (the author).

10. CONCLUSIONS

The Visiview application performs visualization of multi-dimensional data sets, and successfully fulfills its two principle requirements: it permits the nature of functional dependencies to be determined and it allows regions of common behaviour to be identified.

The tool allows the dimensionality of the data to be viewed using a range of techniques, as summarized in Table 1. The techniques can be applied to multi-dimensional data sets, or to provide additional insight into data sets of lower dimensionality. Successful application is possible, even by novice users, and many of the techniques correspond well to functionality expected and desired by subjects used to solving this problem in less automated ways.

The application code, experimental data and some screen shots may be found at the the web site: `http://cs.ru.ac.za/~cssb/visiview/`.

REFERENCES

1. S. D. Bangay, *Modelling Parallel and Distributed Virtual Reality Systems for Performance Analysis and Comparison.* PhD thesis, Computer Science Department, Rhodes University, November 1996.
2. R. Jordan and R. Lotufo, "Khoros 2: Data model, programming and visualization tools." Khoral Research Inc, available via the WWW at: http://www.khoral.com/, 1996.
3. W. Hubbard and B. Paul, "Vis5D." available via the WWW at: http://www.ssec.wisc.edu/~billh/vis5d.html, 1997.
4. D. A. Keim and H. Kriegel, "VisDB: Database exploration using multidimensional visualization," *IEEE Computer Graphics and Applications* **14**(5), pp. 40–49, September 1994.
5. C. Beshers and S. Feiner, "Autovisual: Rule-based design of interactive multivariate visualizations," *IEEE Computer Graphics and Applications* **13**(4), pp. 41–49, July 1993.

6. H. Lee, H. Ong, and K. S. Sodhi, "Visual data exploration." available via the WWW at: http://jsaic.iti.gov.sg/pubs/papers/abstract95.html, 1995.
7. F. S. Bundy, "The analysis of T48 low pressure turbine inlet temperatures using parallel coordinates," Master's thesis, Chemical Engineering Department, University of Florida, May 1995.
8. L. F. Hodges, "Tutorial: Time-multiplexed stereoscopic computer graphics," *IEEE Computer Graphics and Applications* **12**(2), pp. 20–30, March 1992.
9. T. McReynolds and D. Blythe, "Programming with OpenGL: Advanced Rendering." SIGGRAPH '97 course notes, available via the WWW at: http://www.sgi.com/Technology/OpenGL/advanced97/notes/notes.html, 1997.
10. M. Kik, "Stereo-scopic editor." available via the WWW at: http://www.tcs.uni.wroc.pl/~kik/gmk/GMK.html, July 1997.
11. E. H. Chi, J. Riedl, E. Shoop, J. V. Carlis, E. Retzel, and P. Barry, "Flexible information visualization of multivariate data from biological sequence similarity searches," in *Proceedings of IEEE Visualization '96*, pp. 133–140, IEEE CS Press, 1996.
12. C. Ware and G. Franck, "Evaluating stereo and motion cues for visualizing information nets in three dimensions," *ACM Transactions on Graphics* **15**(2), pp. 121–140, April 1996.