Visualizing particle motion in 3-D

Henry C. Bland

ABSTRACT

This paper describes a 3-D representation of particle motion that addresses some of the shortcomings of 2-D hodograms. This technique uses "Rayshade", a visual ray tracing program, to render the path of particle motion with 3-D perspective. The path is represented by spheres indicating sample points, and cylinders representing the path between samples. Data from three component geophones are used to illustrate the technique, and show how 3-D representation helps in the interpretation of the particle motion. By using a number of networked computers to carry out the ray tracing in parallel, it is shown that this technique can be used when sophisticated real-time rendering engines are unavailable.

INTRODUCTION

With the availability of three component seismic surveys, the complete particle motion is known at each receiver location. The output of cartesian-configured geophones directly translate into the x, y and z position of the particles located at the geophone. The conventional way of viewing this particle motion is with a 2-D hodogram. Standard hodograms project the 3-D particle motion onto a 2-D plane. The resulting hodogram usually displays vertical displacement in the vertical axis, and radial (or in-line) displacement in the horizontal axis. With the aid of realistic visualization hardware and software, it is possible to view all three components of particle motion simultaneously.

Hodograms generally suffer from two problems. First, they don't show the full motion of the particle. This is because the motion must be mapped onto a single plane. In cases where particle motion does not fall within a single plane, some information is lost. The second, and more common problem with hodograms is that they are hard to read – especially if the particle motion path crosses itself.

CHOOSING THE 3-D REPRESENTATION

In order to visualize particle's motion in perspective we turn the path into a solid, 3-D object. We then choose a point of reference from which to view the object. A rendering system can then be used to view this object from the perspective of our point of reference. The final output of the rendering system is an image on a computer display or hard copy device.

In choosing the optimal object to represent the particle path, there were four considerations:

1) The object must have some bulk, or thickness. Only with thickness can perspective viewing give the impression of a third dimension.

2) The object must show the time-association between discrete sample points.

3) The object must not take up too much room, so as not to obscure parts of itself.

4) The object must have a simple geometric description which can be rendered easily using existing software or hardware.

The representation that was selected was an object composed of spheres and cylinders. This representation was copied from one commonly used to visualize molecules. For particle path visualization, the spheres are placed at each sample point's x, y, and z location (Figure 1a). Cylinders are used to connect chronologically adjacent sample points (Figure 1b). Naturally, there will be overlap of spheres and cylinders at sample point locations. This does not create any problems, since the ray tracing software always reflects light off the outer-most visible surface. The end result is that the path resembles a "tube" with rounded joints at each sample point (Figure 1c).

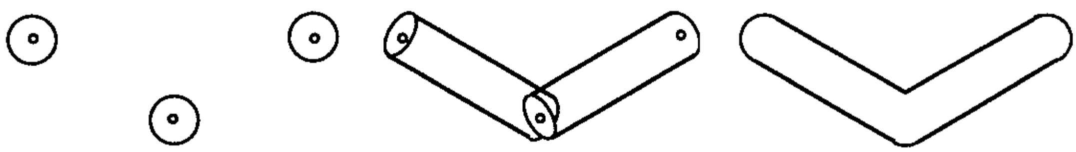

Figure 1a Figure 1b Figure 1c

In a standard hodogram, the dimension of time is indicated with a repeating cycle of markers drawn at even time intervals. The repeating cycle of markers lets one see the direction of particle motion as time passes. In this visualization experiment, we decided to change the colour of sample spheres and connecting cylinders to show the passage of time, rather than use markers. Since the rendering software can handle objects of different colours, we specify a different colour for each sample point's sphere, and interconnecting cylinder. Colours are selected from the HSV (Hue, Saturation, Value) colour model by varying the hue with time. The HSV values are then translated to RGB (Red, Green, Blue) values for the rendering program (Foley and Van Dam, 1984). This gives the effect of traversing all the colours of the spectrum, in order, as time progresses from the beginning sample to the end sample. Although the hue is changed, the brightness and saturation need to be held constant, since they contribute to depth perception in the resultant image. It should be noted that modulating the colour of interconnecting segments can be applied to a 2-D hodogram just as effectively as a 3-D hodogram. A time-calibrated colour bar can then be used to correlate each colour with time.

CONVERSION OF DATA TO AN IMAGE

A short Fortran program was written to extract the data for a single geophone from a single shot record in a three component survey. The output from this program was a text file containing vertical, radial (in-line) and transverse data points for several time samples. Another program, written in the Perl language, was used to process these tabular data into a series of object definitions suitable for input to a rendering program. This program handled the selection of rendering parameters (such as the point of reference), the choice of colours for different portions of the path object, and the scaling of the data. It important to scale the data so that all the path-objects lie within the field of view of the point of reference.

Due to the research nature of this project, no attempt was made to optimize the performance of the visualization. The aim instead, was to use freely available software on the hardware that was available. If we had access to a real-time 3-D rendering system, we would have surely used it, since ray tracing is an extremely compute intensive method of 3-D visualization, and the high-realism capabilities of ray tracing were largely unused. An approximation would have sufficed. Instead of having a hardware rendering system, the systems that were available were nine Sun Sparcstation 2 workstations and one dual CPU Sun 670MP server system. In addition to the hardware was a free visual ray tracing program called "Rayshade".

Rayshade is a program that evolved from work done at Princeton University in 1987-1988. It was based on a public domain ray tracer written by Roman Kuchjuda. The current incarnation was written by Craig Kolb and Rod Bogart and many other members of the visualization research community. It is a freely distributable ray tracer that accepts a description of a scene and produces a colour image corresponding to the description. Rayshade is renowned for its high quality output, and its ability to handle complex scenes containing solid, transparent, semitransparent and translucent objects, atmosphere effects, surface textures, and objects formed from boolean combinations of geometric primitives.

To increase the speed of rendering, a variant of Rayshade was used that utilizes several networked computers at the same time to generate an image in parallel. This extension to Rayshade, called "Inetray", was written by Andreas Thurnherr at the Interdisziplinaeres Projektzentrum fuer Supercomputing at the ETH in Zurich Switzerland. Inetray is ideally suited to the computing environment at the CREWES project, since many of the workstations have spare CPU cycles during off-peak hours. Using this program all 9 workstation CPU's and the two server CPU's were combined to produce output images. A comparison of imaging times showed that processing speed increased approximately linearly with the number of systems working on the problem at the same time. It is obvious that there must be limitations to the number of systems that can parallel process efficiently. However, this limit was not reached in our tests.

PATH ROTATION

After rendering several views of a single data set it became apparent that it would be useful to view the same data set from many different angles to get a better appreciation of the full 3-D particle motion. This was easily accomplished by changing the viewpoint incrementally over a series of viewing frames. In an initial test we rendered 50 frames – each with an increasing rotation of 7.2 degrees. By displaying these frames one after another, the particle path was made to spin through all 360 degrees. This greatly enhanced the appreciation of the particle motion, since it was possible to view it from all sides.

CONCLUSIONS

Visualizing particle motion in 3-D is superior to 2-D visualization in traditional hodograms since it gives a better understanding of the full 3-D particle motion. It also avoids the need to choose a optimal viewing plane, as is needed in 2-D hodograms. The problem of displaying the passage of time has been addressed. By correlating the hue of sample points with time we can easily determine the direction of particle motion and pinpoint the time for a particular sample by comparing the sample's colour to a time-calibrated colour bar.

Although high speed rendering engines are ideal for this kind of visualization, ray tracing can also be used if there is no hurry for results. The computationally intensive task of ray tracing can be sped-up significantly by using distributed parallel processing.

ACKNOWLEDGEMENTS

The author wishes to thank the Department of Geology and Geophysics for providing some of the computer facilities used in the preparation this paper. Thanks also go to the authors of Rayshade, the Utah Raster Toolkit, Inetray, and Perl for their contributions of free software. Dr. Don Lawton is also thanked for supplying the three component data used in this paper.

REFERENCES

Foley, J.D., Van Dam,A., 1984, Fundamentals of Interactive Computer Graphics:Addison-Wesley Publishing Company, pp615

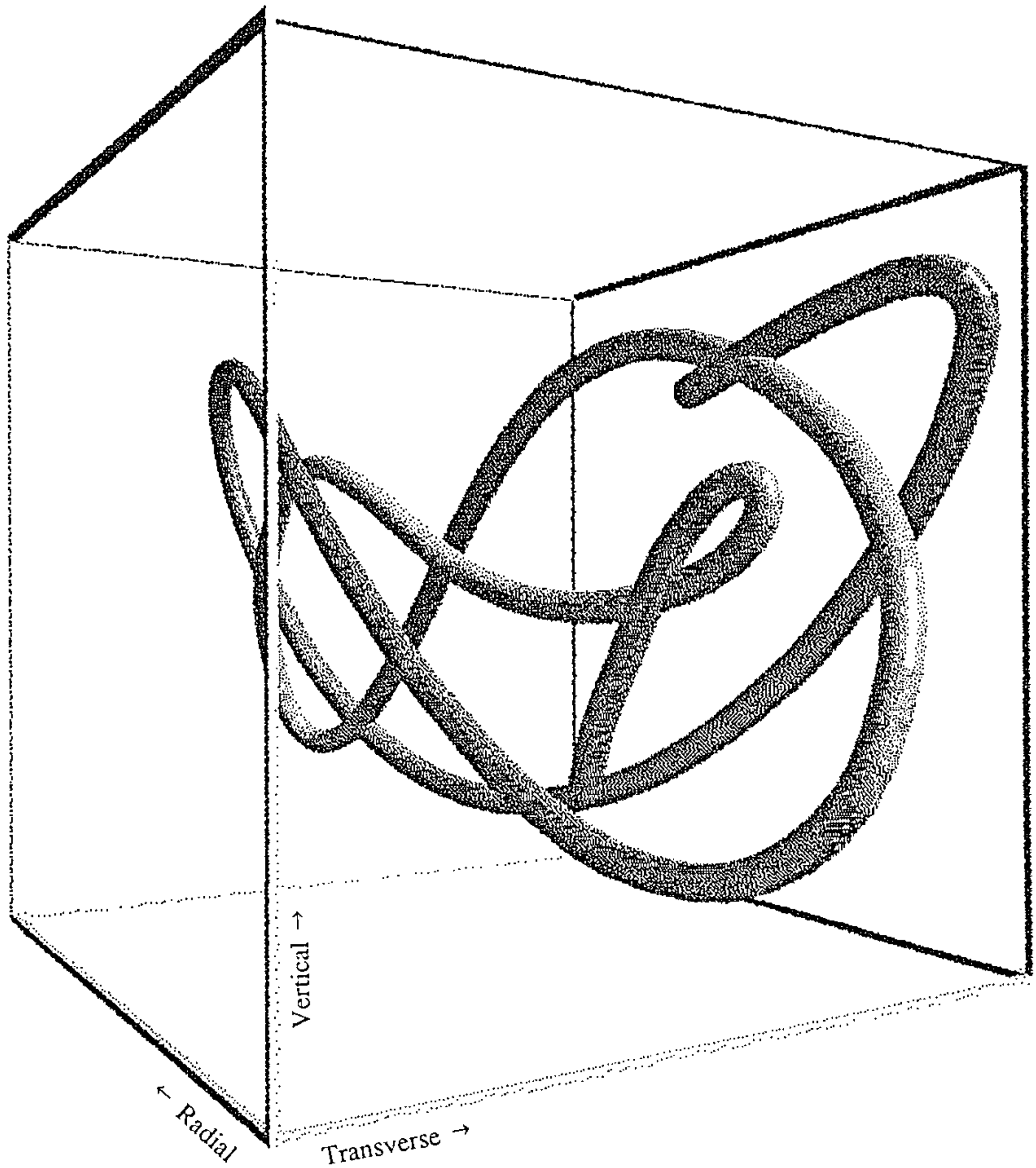

FIG. 2. Particle path visualized in 3-D. Colours have been removed for printing.