Subject: Looking for Large Data Sets in Exchange for 2D Visualization Tool
Posted by mcheng on Tue, 12 Jul 1994 15:27:36 GMT
View Forum Message <> Reply to Message

Hi netters,

A while back I posted a message asking people about problems with graphing (in 2D) large amounts of data. Some have indicated that this is an application that Khoros, AVS, or IDL currently do not handle well.

I have now implemented a C++ library that can be used to graph large amounts of data. I'm currently using it to look at some soil sensor data. And I'm also looking for different data sets that I can use to demonstrate its capabilities. In exchange for a large sample data set, I will provide you either with my library, or even construct the application according to your specifications so that you can use it for your own visualization. (The objected oriented approach allows me to contruct a new application in a short time.) If you have problems constructing visualizations with 2D plots, this would be an alternative that you can try out. I would also appreciate any pointer to any data sitting out there, waiting to be visualized.

If you wish, I can try to make the software available via anonymous ftp. (I would have to find out how, though)

Some information about the C++ library:

-Platform: Unix and X windows, using Xlib and Xt.

-Features:
Large amounts of data: no restriction on data size
Overlapping graphs, and multiple windows.
Linked axes between graphs (scroll one and you scroll the other automatically).
Point, discrete, and continuous line graphs.
Fine grain control: controls appearance on the granularity of each data point or each line sgement.
Can handle missing data in line graph.

-Model: Textual data (TData) --> graphical data (GData) --> visual
Textual data (TData) is supplied by the user, as a sequence of records. Graphical data is created by mapping each record of TData into a record of GData. Each GData record contains the following attributes: location, size, color, pattern, orientation, and shape. Graphical data is rendered as a visual, currently a bitmap display.

-Programmer's tasks: provide the mapping from TData to GData,

decide which graphs are linked, and act on user interaction.
However, programmer need not worry about internals such as how
to do a scroll or zoom, when to perform the mappings, or how to handle
memory management.

-Efficiency:

Does not use virtual memory. Memory buffer for TData. Memory and disk buffers for GData. Prefetching for GData and TData.

Optimized for drawing large number of shapes.

This research was conducted under the supervision of professors
at the University of Wisconsin-Madison, and supported in part by an IBM
fellowship. For further information, please contact:

graduate student: Michael Cheng (mcheng@cs.wisc.edu)
professor: Miron Livny (miron@cs.wisc.edu)
professor: Raghu Ramakrishnan (raghu@cs.wisc.edu)