University of Missouri-Rolla
Electromagnetic Compatibility Laboratory

Title: EMAP Data Visualization

Report Number: TR93-4-024

Author: Andrew M. Schadt

Date: December 1993

Abstract

EMAP and other data generating programs have made data visualization techniques important in comprehending the output of these programs. Different types of plots have different advantages and disadvantages. Choosing a visualization software package depends on factors like what types of plots are important, basic software needs, and use of a standardized data format.

Abstract

EMAP and other data generating programs have made data visualization techniques important in comprehending the output of these programs. Different types of plots have different advantages and disadvantages. Choosing a visualization software package depends on factors like what types of plots are important, basic software needs, and use of a standardized data format.

Introduction

With the introduction of more advanced computers and computer programs come the possibility of modeling complex phenomena more accurately than was possible just a few years ago.[1] The use of computer models to simulate complex situations gives researchers insight into how complex phenomena occur. The extent of this insight depends on the formation of advanced methods of displaying and visualizing the data.

EMAP is a program written for the purpose of calculating values of various electromagnetic properties in a volume around a circuit board. The output of the program is in the form of three-dimensional fields of vector quantities. After the program was written, there immediately became a need for a good method to represent the data graphically. Although the data contained a wealth of useful information, its complexity made it difficult to extract this information from simple plots. To get the maximum amount of useful information out of a set of data, the method used to display it can be as important as the data itself. This rapidly growing field of study is known as data visualization.

Data Visualization

The basic aim of data visualization is to take a series of raw numbers and represent them in a way that makes it possible to gain insight into the data. Visualization can be thought of as realization, or the complete and correct comprehension of data.[2] To gain an understanding of a set of data, the key features and important trends must be established. Only then can a mental picture of the data be formed leading to insight into the data. Although this may seem like a simple task, it can become difficult when many dimensions and/or vector quantities are involved.

A series of few single data points can be easily understood using a simple line or bar graph. Single scalar data points in two dimensions are still very manageable since computer monitors naturally represent objects in two dimensions. If the 2-D data has more than one parameter to be represented at each point, such as several scalars or a vector, displaying all of the data simultaneously becomes difficult. The number of dependent variables that are represented at each point can be thought of as the *rank* of the data.[3] If multiple scalar values are represented, each one increases the rank of the data. If vectors are represented, the rank of the data is also increased. The problem deepens when three (or more) dimensions are involved, especially as the rank of the data increases. In this case representing all the data simultaneously on a two-dimensional surface is not possible.

The amount of information that can be represented on a two-dimensional surface is limited. Each point on the screen can represent a point in 2-D space. The value of this point can be represented using differences in properties like color and texture. Three dimensional data can be represented this way by drawing an isometric view of the volume on the screen. This type of plot is commonly called a volumetric plot or rendering. A sample of this type of plot is in Appendix 1. These plots are limited by the fact that the data displayed at the outside edges of the volume obscures the data that is inside.[4] To view the data inside, some method of slicing the outer data off must be used. Since slicing the data in one plane may show more details than another, the direction of slicing can greatly affect the success of the visualization. Buffering the data, or decreasing its intensity, will cause lower magnitude data to disappear revealing only the outstanding features making the decision of how to slice the data easier. Rotating the data so that different sides of the outside surface are viewed can also aid in realizing the data.

Another possibility is to construct perspective view that tilts the 2-D slices and makes them appear to have height. Differences in height can be combined with color and texture to represent different variables.[5] These types of representations are commonly called surface plots. Figure1 contains an example of a simple surface plot.

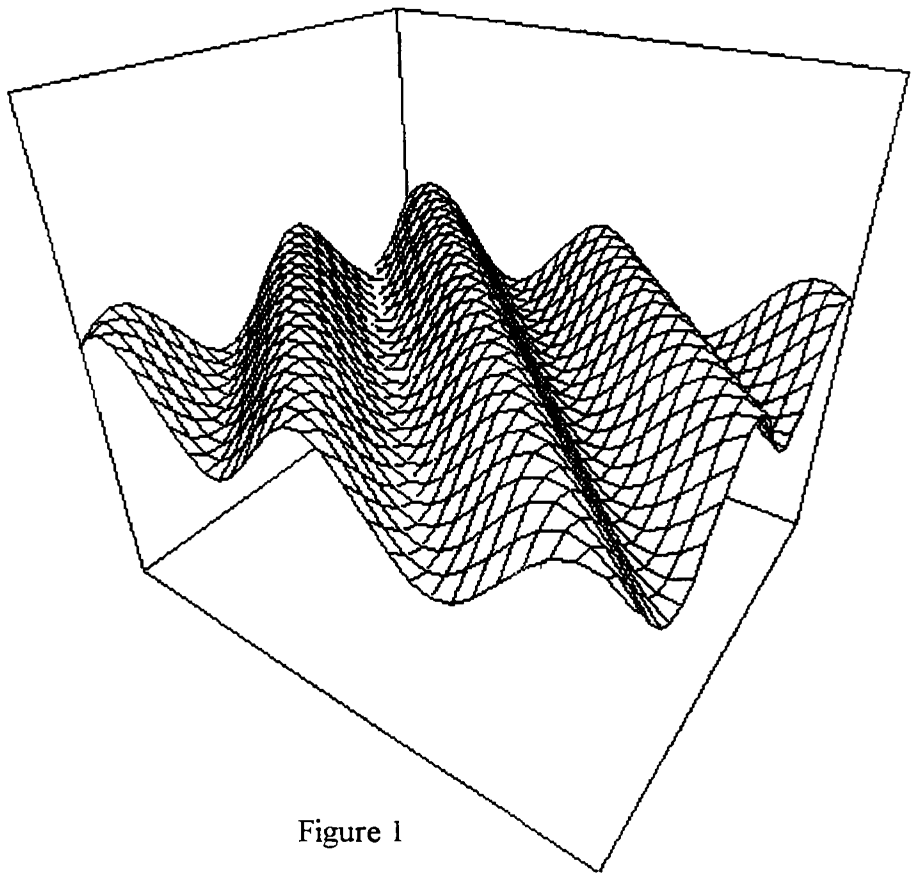

Figure 1

The third dimension can be represented by stacking multiple plots. This method is limited by the number of plots that can be displayed at one time. In most cases this limits how well changes over the third dimension are represented. If the slices are tilted so that many plots can be fit on the screen, details in the first two dimensions become obscured. Again the ability to rotate the plot so they can be viewed from different directions is very helpful. Slices of volumetric plots can also be stacked this way as in Figure 4 in Appendix 1.

In both of these cases, only data of a limited rank can be represented. If the data represents vectors, only one component of the vector or the magnitude can be display at a time. In order to display three dimensional vectors, more parameters are needed besides color and texture. This can be accomplished by using symbols at each data point. Arrows lend themselves to fairly well to representing vectors. The direction of the arrow represents the direction of the vector. Length, color or thickness can be used to represent magnitude. Figure 2 shows a sample arrow plot.[6]

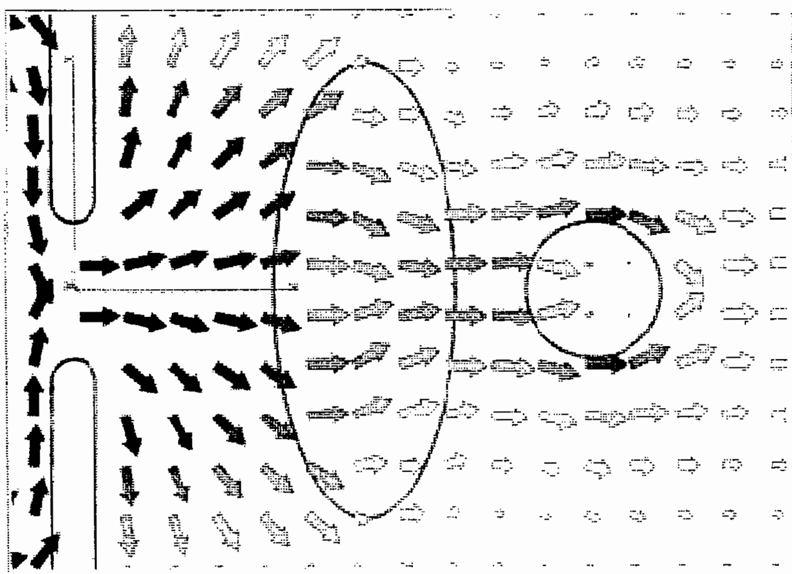

Figure 2

The problem with this method is that the arrows take up more space on the screen than simple dots of color, so not as many data points can be displayed. As with the volumetric rendering mentioned above, the arrows on the outside surfaces will obscure the data inside a 3-D volume making it necessary to slice the data in some way. This method, as with the others, does not allow all of the data to be seen at once.

One way to get around not being able to see all of the data simultaneously is by incorporating animation. Animation is normally associated only with displaying data that changes over time, but it can also be used with non time changing data. In the case of a volumetric rendering, animation can be used to peel off slices of the data in sequence allowing the viewer to scan through all of the data over a short period of time.[7] Animation can also be used to rotate the plot so that features on different sides of the volume can be viewed together.

Since there are so many different ways to represent data graphically, the logical question is: Which method is the best? The answer to this question depends highly on what type of data is being visualized, and what aspects of the data are the most important. Some researchers require a feeling for the overall trend of the data while others may benefit most by concentrating on certain areas of the data. Sometimes a combination of different plots may be needed to gain true understanding of the data. This is especially true regarding data that cannot be completely represented at one time. In any case the ability to produce a variety of different plots would be ideal. The same conclusion applies in the case of visualizing the EMAP code output. The plot that will be understood the best will change with the aspect of the data being studied.

Desired Software Features

Picking the best software package involved considering a number of desired qualities and concerns. One immediate concern was to convert the output data of the EMAP code to a format that the visualization program could read. Preferably this would be some sort of standardized format that could be used with more than one visualization tool. It would also have to be a format that could easily be generated from the existing EMAP data output.

The most important feature of the visualization program would be the ability to display the data in the way that allows the most useful information to be derived from it. As discussed in the previous section, this would ideally be a program that would allow generation of as many different types of plots as possible. Also the ability to view these plots from as many different angles and orientations as possible would be important.

After the most revealing plots and views are generated on the computer, there is usually a need to transfer this screen output to printed output. This suggests that another important feature for a visualization program is the ability to produce presentation quality output. This output should be an exact representation of what was viewed on the monitor. It could be as simple as a command that would produce a post script or some other format file that could be sent directly to a printer. A useful option would be the ability to produce a raster file in a standard format that could be imported into programs such as a report generation program.

Selection of Software

Although finding a program that generated the best type of plot for the data was a primary concern, there were other practical considerations. These included cost, portability, and flexibility. The use of a standardized data format also became an important element in the final decision.

Since the EMAP code was developed as a nonprofit research project, it would not make sense for it to require an expensive commercial program to view the output data. For that reason, cost to the visualization software was a significant factor. Several excellent packages were ruled out due to this limitation. One such program was PV-WAVE Point and Click. It has the ability to produce a wide variety of different kinds of plots very quickly and easily, but it's $2500 price tag makes it cost prohibitive.

Ease of use was also a required feature. A visualization program can only do what its user instructs it to. If the program requires knowledge of an extensive command structure, it can only be effectively used by a specially trained user. The intention was to find a program that virtually anyone with minimal knowledge could learn to use effectively in a short period of time. That way more effort could be devoted to visualizing the data rather than operating the software.

Since the EMAP code is currently running on an HP 9000 workstation, it made sense to require the visualization program to run on the same system. If the visualization software had to be run on a different system, the hassle of exporting the data to the new system for viewing would be involved. Although exporting the data to a different system may be desired in some cases, it should not be a requirement.

One of the most interesting aspects of visualization software features is the idea of expressing the data in a standardized format that multiple visualization programs would accept. One of the problems with current visualization practices is the lack of a standard format. Data formatted for one visualization program must be reformatted if it is to be accessed by a different program. A standard format, one that is supported by several visualization tools, would offer a great deal of flexibility. Not only could a larger variety of programs handle the data, but easy transfer of data between systems would also be possible. Although an overall standard has yet to be set, several groups have attempted to devise data formats that are flexible enough to be used for practically any kind of data. One such format is the Hierarchical Data Format (HDF) developed by the National Center for Supercomputing Applications. HDF in association with the NCSA visualization tools was decided to be the best choice for formatting and visualizing EMAP data.

Hierarchical Data Format

Hierarchical Data Format is a standardized data file format for scientific data and raster images. It is a multi-object file structure that is designed to facilitate the sharing of data among different machines, software packages, projects and people.[8] This allows the flexibility of using the data on different machines and in different programs. Although for convenience it was preferred that the visualization software run on the same system that the data was produced on, using HDF prevents this from being restriction. The standardized format also allows the data to be accessed from different programs making it far more flexible than a nonstandard format.

Hierarchical Data Format was developed for any user who produces scientific data of virtually any kind. It is an attempt to standardize the transfer and handling of scientific data and raster images between machines and software packages. Since scientific data can come in many different forms, it was designed to be flexible enough so that it can be adapted to the user's needs.

HDF files are self describing.[9] Predefined tags are used to identify different types of data, it's dimensions, and any other pertinent information. This allows the possibility of virtually any type of data being in HDF files. Since the data is described by the tags, it is possible to save completely different types of data in the same file. HDF files could contain scientific data as in the output of the EMAP code, or raster images produced by a visualization program, or both.

HDF application software is produced by NCSA in the form of scientific visualization tools, calling interfaces, and command line utilities. Several scientific visualization tools are offered that read, write and display HDF files directly. Calling interfaces allow HDF files to be read and written from within FORTRAN and C programs. They are designed so that the programmer only has to have a minimum knowledge about HDF files in order to incorporate them into an application. Command line utilities allow users to directly operate on HDF files. These utilities allow common operations to be performed from the command level. Normally this would require a program to be written. HDF visualization applications are available for UNIX, Macintosh, MS-DOS, and CRAY systems.

A feature that makes the HDF format especially attractive is the fact that it is available in the public domain. This makes it available to anyone at practically no cost. NCSA hopes that HDF will become the standard in the scientific data visualization community. Expressing the EMAP data in Hierarchical Data Format was the logical choice to make the more easily usable and provided a good stepping stone to several visualization programs.

NCSA Visualization Tools

Along with the HDF format, NCSA offers visualization tools that display the data. XImage and XDataSlice run under X Windows on UNIX systems. HDF applications are also available on other systems.

XImage is a 2-D interactive color imaging and data analysis tool. It displays the actual data values in a spread sheet form and can create color contour plots, shaded images, color histograms, and profile plots. It also features animation of multiple color images.[10] Since the data produced by EMAP is in three dimensions, X Image would only be useful for manipulating 2-D slices.

XDataSlice is a color imaging and data visualization tool capable of 3-D volume rendering. It will display color raster images of 2-D slices of 3-D data sets. Slicing and dicing can be accomplished at any angle or orientation. Multiple images then can be tiled in the same window. Actual data values can be displayed in spread sheet form with synchronized selection between corresponding data and image windows.[11] Color palette manipulation allows different color or gray scale schemes to be used for displaying the data.

XDataSlice also includes a V-buffer volume rendering feature. This feature allows 3-D data sets to be rendered as transparent or semi-transparent volumes making it possible to see features that may have previously been obscured. Ranges of data are divided into substances, and then the substances are assigned a opacity value. The opacity value designates how well the substance is seen, giving the user control of which parts of the data are displayed and which are transparent.[12]

One of the most powerful features of XDataSlice is the ability to animate sequences of slices continuously and in single steps. The animation routine allows the user to manually scroll through slices in either direction, or see a autoreversing continuous display. In the continuous display mode, the speed of the animation can be controlled to give the best view of the data. It also has the ability to skip frames.

Output of XDataSlice images can be handled in two ways. Images produced can be saved as color postscript files ready to print, or as raster images in HDF files. Figure 1 and 2 in Appendix 1 contain samples of XDataSlice output. The images were produced from sample wind velocity data of a severe thunderstorm simulation.

NCSA HDF calling routines and visualization tools are available via anonymous FTP from ftp.ncsa.uiuc.edu (141.142.20.50). Questions regarding NCSA software can be directed to: softdev@ncsa.uiuc.edu (Internet) or softdev@ncsagate (BITNET)

Conclusion

Data generating programs such as EMAP have made it necessary to investigate different methods of data visualization. The task of representing 3-D vector data on a 2-D surface will always leave the user viewing only part on the data at a time. Surface plots, Arrow field plots, and volumetric rendering all have advantages and disadvantages. The type of plot that best describes a data set depends what aspects of the data are most important.

Deciding on a software package to visualize EMAP data was dependent upon several factors including cost, usability, portability, flexibility. The use of a standardized data format so that multiple visualization programs on different systems could be used was also an important factor. NCSA Hiearchical Data Format was determined to be the best format to represent the EMAP data files in.

HDF is a very flexible data format that can be displayed by visualization tools on several different systems. XDataSlice produces volumetric rendering of HDF data under X Windows on UNIX systems. It allows for arbitrary slicing and can animate sliced frames to let the user see into the data. For these reasons, XDataSlice in association with HDF was choosen to be the best software to visualize EMAP data.

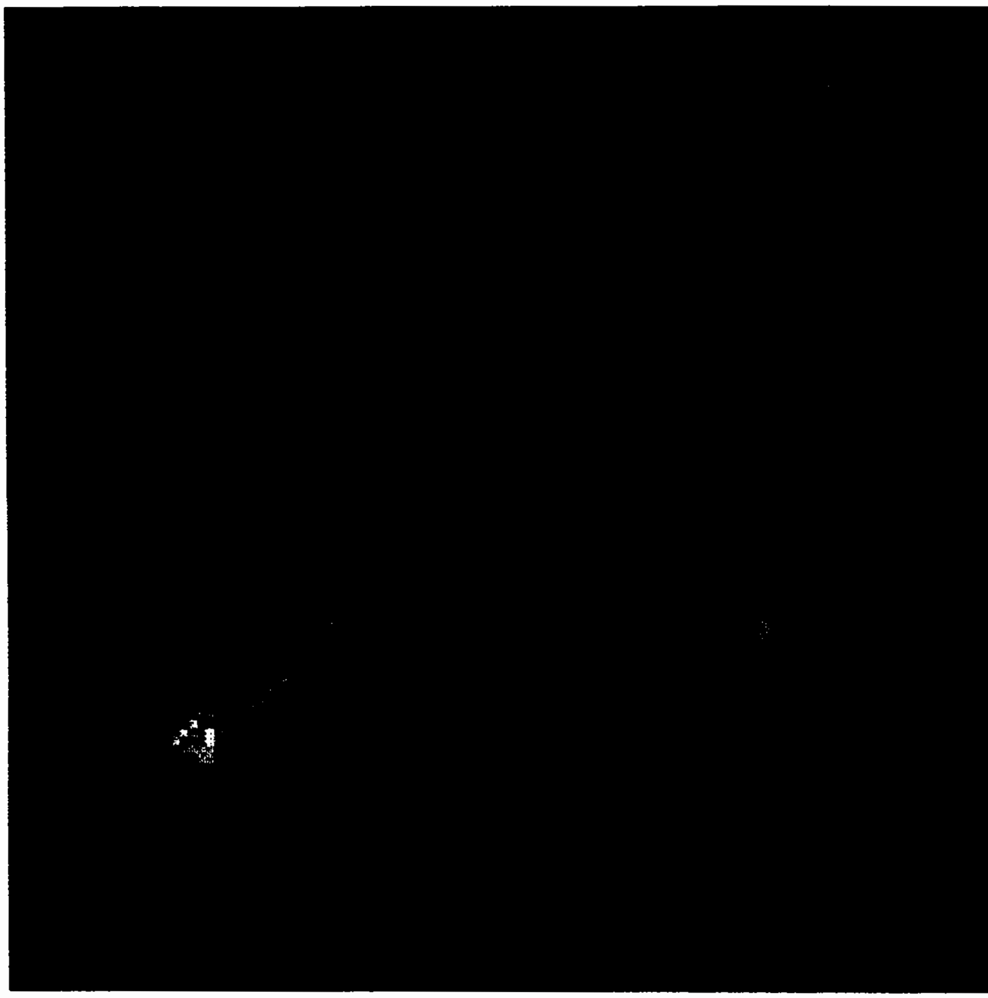

Figure 3 Sample XDataSlice Output

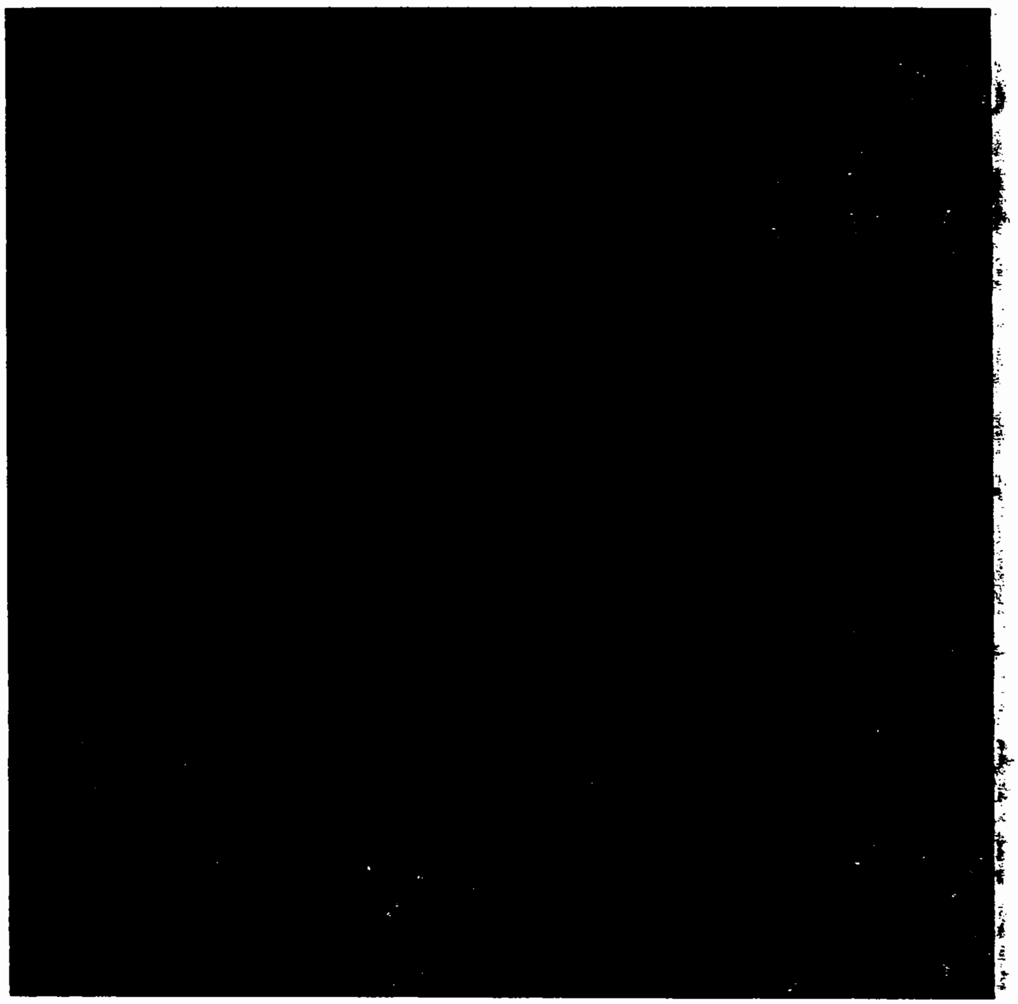

Figure 4 Sample XDataSlice Output

[1] L.J. Rosenblum and G.M. Nielson, "Visualization Comes of Age," *IEEE Computer Graphics and Applications*, Vol. 11, No.3, May 1991, pp. 15-17.

[2] L.J.Rosenblum and B.E. Brown, "Visualization," *IEEE CG&A*, July 1992, pp.18-20.

[3] L.A. Treinish, "Inside Multidimensional Data," *BYTE*, April 1993, pp.132-133.

[4] P. Wayner, "Image Building," *BYTE*, April 1993, pp.137-140.

[5] P.K. Robertson, "A Methodology for choosing Data Representations," *IEEE CG&A*, May 1991, pp.56-67.

[6] L. Bomholt, P Regli, C. Hafner, P. Leuchtmann, "MMP-3D: A Package for Computation of 3D Electromagnetic Fields on PC's and Workstations," *Applied Computational Electromagnetics* (Conference Procedings), March 19-22, 1990, p.31.

[7] T.S. Yoo, U.Neumann, H. Fuches, S.M. Pizer, T. Cullip, J. Rhoades, R. Whitaker, "Direct Visualization of Volume Data," *IEEE CG&A*, July 1992, pp.63-71.

[8] University of Illinois at Urbana-Champaign., NCSA HDF Calling Interfaces and Utilities, July 1990, p 1-2.

[9] University of Illinois at Urbana-Champaign., NCSA HDF Calling Interfaces and Utilities, July 1990, p 1-3.

[10] University of Illinois at Urbana-Champaign, NCSA X Image, March 1991, p 1-1.

[11] University of Illinois at Urbana-Champaign, NCSA X DataSlice, Sept. 1989, p 1-1.

[12] University of Illinois at Urbana-Champaign, XDS V-Buffer Volume Rendering Feature, June 1991, pp.1-3.