NASA
Technical
Memorandum

SURVEY OF VISUALIZATION AND ANALYSIS TOOLS

By P. J. Meyer

Space Sciences Laboratory
Science and Engineering Directorate

January 1994

NASA

National Aeronautics and
Space Administration

George C. Marshall Space Flight Center

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE January 1994	3. REPORT TYPE AND DATES COVERED Technical Memorandum

4. TITLE AND SUBTITLE

Survey of Visualization and Analysis Tools

5. FUNDING NUMBERS

6. AUTHOR(S)

P. J. Meyer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, D.C. 20546

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM-108436

11. SUPPLEMENTARY NOTES

Prepared by Space Sciences Laboratory, Science & Engineering Directorate.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified--Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A large number of commercially available visualization and analysis tools are available to the researcher. This document discusses some of the strengths and limitations of some of these tools, from the viewpoint of the earth sciences discipline. Visualization and analysis tools fall into one of two categories: those that are designed to a specific purpose and are non-extensivle, and those that are generic visual programming tools that are extensible. Most of the extensible packages examined incorporate a data flow paradigm.

14. SUBJECT TERMS Visualization, Analysis, Survey, Data Flow, Extensible	15. NUMBER OF PAGES 40
	16. PRICE CODE NTIS

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited

NSN 7540-01-280-5500

PREFACE

The motivation for this paper is to assist earth scientists at the NASA Marshall Space Flight Center in the understanding of some available visualization and analysis tools. The tools discussed have been used by the author for the analysis of atmospheric data. Of these tools, the author is intimately familiar with the McIDAS system and is the technical contract monitor for the effort to develop the VIS-5D analysis tool.

While this paper does not survey all visualization and analysis packages, it does discuss some packages with which the author has had experience. It is hoped that this is a first in a series of survey papers, and that as new visualization and analysis tools become available and as the examined tools mature, the author may update this report.

The author would like to thank Ms. Karen A. Butler, and Dr. Michael E. Botts for reviewing this document, and for their helpful suggestions, and Ms. Tauna Moorehead for her editorial review, and assistance in preparation of this document.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

TECHNICAL MEMORANDUM
SURVEY OF VISUALIZATION AND ANALYSIS TOOLS

INTRODUCTION

Data sets for conducting scientific research continue to grow in size and complexity. Simple tabular printouts and stripcharts used to be sufficient for scientific inquiry. Then came the advent of pen plotters and their ability to contour two- or three-dimensional representations of data sets. Recently, however, scientific data sets have become quite massive due to finer resolution numerical models and greater numbers of included parameters. Multispectral remotely sensed data from aircraft and space-borne instruments require large data storage systems as well as sophisticated software for processing. The scientist also is motivated toward an interdisciplinary approach to answer his questions. These large data sets now require new and sophisticated tools to allow the scientist to make sufficient queries about the interdisciplinary data in order to answer these questions. The simple tabular printouts just do not suffice. These new tools also require sophisticated workstations in order to provide a timely and visual response to the researcher.

Many visualization and analysis tools are available for use on UNIX based workstations. No one tool is able to do everything the scientist desires. Some packages are strong in image analysis capabilities while others may be strong in their ability to graphically depict the inter-relationships among various scientific parameters. Most likely, the scientist will require some suite of tools to conduct his/her research.

Two different software designs typically describe visualization and analysis tools. These include general-purpose extensible programs and those which are designed for a specific purpose.

Data Flow Paradigm

The general-purpose environments tend to allow the user to add new modules to an existing core system. At present, they also tend to be based on a data flow paradigm. A data flow paradigm moves data between modules with each, in turn, transforming the data into some intermediate or final form desired by the user, be it some visual display or some new variable. These tools are also designed around a visual programming environment. Tools are also provided to assist the user in the design of a new applications module. A nice feature of all the data flow paradigm environments is that when new data become available to a module which is upstream from the currently executing module, the network will "sense" this and begin to refire (execute) the upstream modules. Only modules downstream of the module with new data (such as a changed slider value) will fire. This conserves execution time by only activating those modules which need activation.

Probably one of the strongest aspects of all the visual programming data flow paradigm packages is their ability to allow the scientist to rapidly prototype an analysis scheme.

These packages are not limited to visualization, they may be used to develop numerical models and even directly combine model simulations with a visual analysis of the output.

Application-Specific Paradigm

Tools designed for a specific purpose tend to be more interactive and responsive in many cases, and perhaps more user friendly. This responsiveness is due in part to all the data being memory resident, with little or no interprocess communication overhead between applications modules, as the application tends to be bundled as one process. Since the tool is designed for some specific purpose, the analysis features are already provided to the end user, relieving him of the responsibility of building new analysis modules. They do, however, tend to be not as extensible as their counterparts. This increases the time required to add new capabilities to the application, as it needs to be done by the developer of the application, not the end user.

This paper attempts to define the capabilities and deficiencies of a few select tools with which the author has had some experience. It is hoped the reader will be able to use this guide to determine which tools to use to further analyze his/her data. This list in no way represents all the analytical and visual tools available. It lists most of the tools currently available to our group. It is not intended that this guide illustrate every feature available from each of the visualization and analysis tools.

Tools discussed in this paper include AVS, Data Explorer, IRIS Explorer, Khoros, Link-Winds, McIDAS, SciAn, VIS-5D and some miscellaneous packages.

AVS

AVS from Advanced Visual Systems is based on a data flow paradigm which utilizes shared memory and interprocess communication. It is a general-purpose visualization package which runs on almost every UNIX platform that supports X Windows 11. It even runs on PC's. The software has a licensing fee of $6,500 for a fixed license, $8,000 for a floating license. A ten user node-locked license is available for $10,000. Licensing for Cray computers, however, is somewhat higher. For $25,000 one may also obtain a developer's license which provides source code. The following discussion is based on both the AVS 4 and AVS 5 Release.

Synopsis

- Visual programming environment.
- Works on almost every UNIX vendors platform using X Windows 11, and color X terminals. Even PC's with SCO Open Desktop. Plan to also port to Windows NT.
- SunVision Image processing, volume rendering support built in.
- 230 supported modules (Release 5). Over 600 third party modules available at the International AVS center.

- Internationaľ AVS center is a member of the Khoros consortium.

- Applications may execute among heterogeneous machines on the network.

- Support for co-routines and parallel module execution.

- Module development in FORTRAN, C, also supports C++ modules.

- Data import application (AVS Data Interchange Application (ADIA)).

- Strongly typed data environment.

- Macro modules may be defined.

- Supports grid templates of uniform, irregular and curvilinear.

- Command language interpreter, allows user written programs to utilize AVS functionality.

- Data probes are supported to allow the user to extract values.

- The user may selectively turn off modules not needed for a particular run. For example, a histogram display may not always be needed.

- Optional support for keyframe animation ($2,000).

- On-line help.

Discussion

AVS is a general-purpose visual programming environment that allows the user to utilize existing modules to generate a program. Alternately, the user may design their own modules for use in conjunction with the predefined modules. The concept is based on a strongly typed data flow paradigm which allows the user to connect, via "pipelines," compatible modules to generate and then execute a visual program. Visual cues are provided to inform the user as to the data types accepted by modules. A plus with this system is that it is based on the commercially accepted X Windows 11 standard. Depending on your screen display capabilities, the software will reconfigure itself to either work on a true-color display or pseudocolor terminal. Figure 1 illustrates the AVS network editor, with a simple AVS program network which reads an image and displays it. Visible in the figure, behind the image, are programming modules under major category headings. One simply drags and drops these glyphs into the workspace, and connects them to form a program.

One of the great strengths of this approach is the general-purpose nature of the software. The scientist may include almost any type of data imaginable, in nearly any coordinate system. In fact, not only does the software allow one to use uniform grids, it also allows one to incorporate rectilinear grid systems, as well as irregular grids and scattered data. Just provide the coordinates to the software and it will handle the rest. Figure 2 shows the grid systems accepted by AVS, and typical names associated with them. These grid data templates are referred to throughout this paper.

Image processing support is provided by the SunVision image processing library, as well as the image processing modules available from the Khoros system. The International
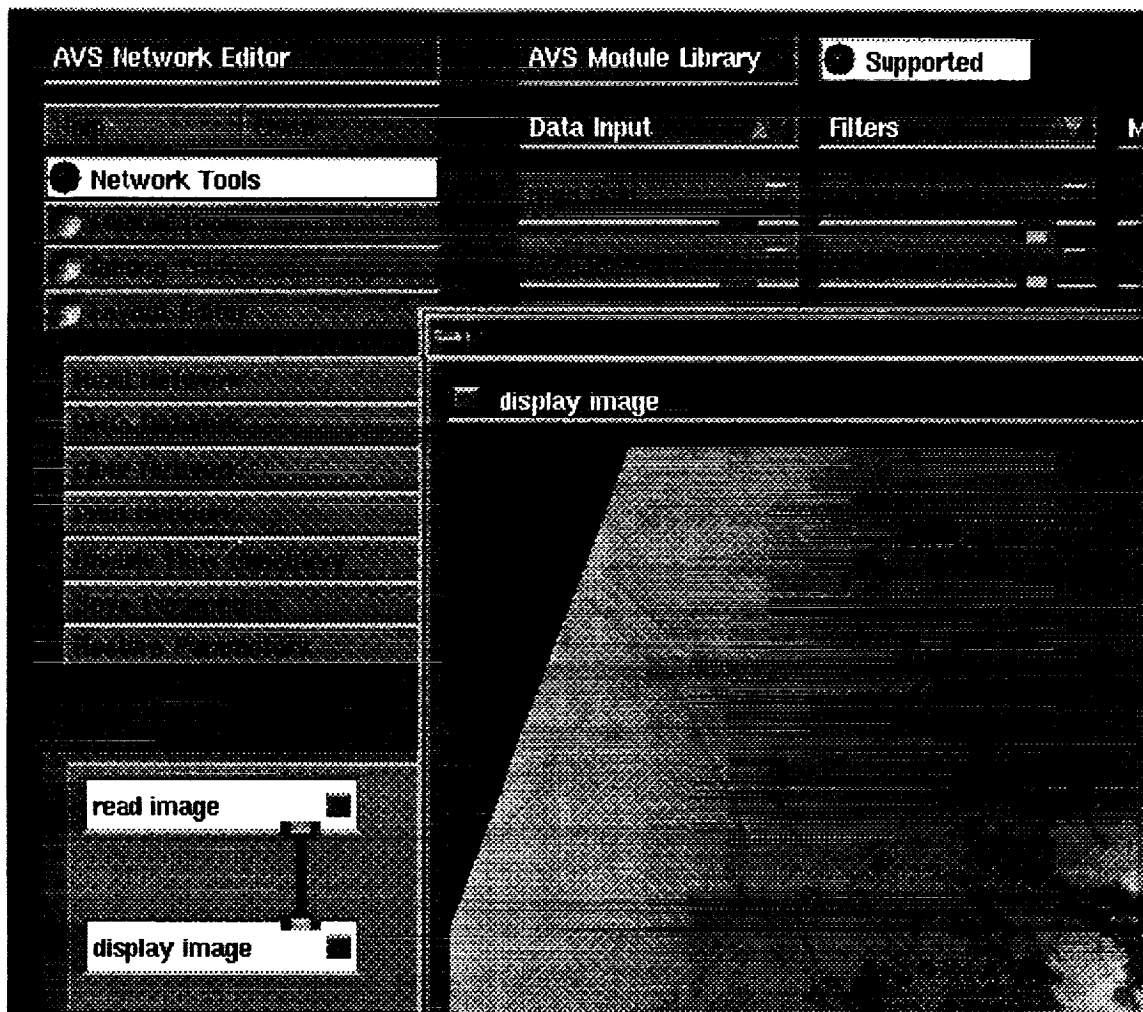
Figure 1. A portion of the AVS network editor; note some of the available modules under the module library categories. A simple network which displays an image is depicted. Note the pipeline connecting the read image and display image modules. The control panel area is not shown.

| Uniform
Regular | Irregular
Perimeter
Separable | Curvilinear | Unstructured
Scattered |
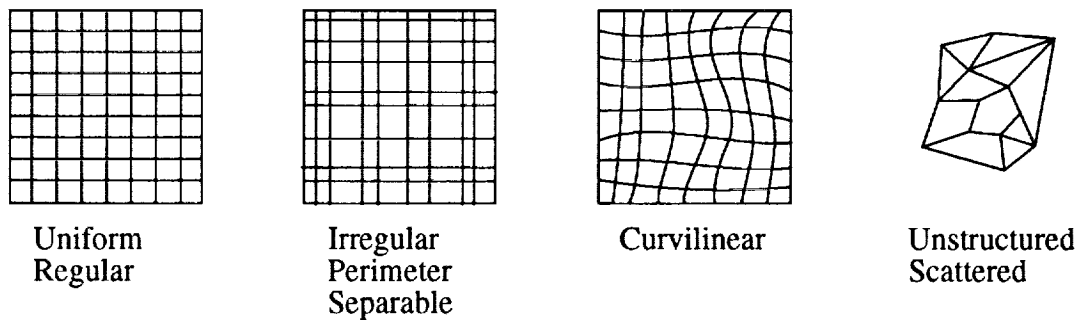
Figure 2. Various grid data coordinate templates with typical names given to each gridding scheme.

AVS center is a member of the Khoros consortium and has placed wrappers around nearly all the Khoros 1.0 functionality. So not only do you get AVS, you also get Khoros.

Over 600 user-developed modules are available free from the international AVS center. They are available via anonymous ftp. Some modules may use specific features of certain machines or hardware. Some of the available modules include arbitrary slicers, modules to read and display images of almost any format, colormap generators, loop control constructs, and output to video devices such as the Abekas video frame storage unit.

Capability to do looping constructs is not very strong with AVS. Some user-written modules are available at the International AVS center, however, they seem limited. Perhaps this will be better supported under AVS 6. However, the system does support upstream capabilities, mainly for use with the data probes.

AVS supports statistical analysis via the IDL and Mathematica packages. The user must license these packages from their respective companies and request the AVS module wrappers when purchasing the respective products. The Mathematica modules are also available free from the anonymous ftp server at Wolfram Research.

AVS has a central control mechanism that determines when to fire modules in the algorithm. This seems to assist the resource utilization issues with regard to their package. By having this central controller, they are able to place multiple modules within one source file, yet still treat them as distinct modules. This reduces the number of open files on the system, as well as the amount of interprocess communication via socket calls and shared memory. It also makes it easier to stop the program if you realize you have made a mistake. Much of their communication is done by passing pointers to physical memory between the routines. The Explorer package, discussed later, relies more heavily on the interprocess communication and tends to tax system resources. AVS only utilizes the shared memory and interprocess communications aspects when necessary. For example, when running over a heterogeneous network, one must use interprocess communication, or, perhaps, when displaying an image in both the geometry window and the image display window.

The AVS Data Interchange Application (ADIA) allows the user to read their data into AVS without having to write code to read the data. The menu system allows one to define the format of the desired data set. Once you have read the data and verified the correctness of the reading routine, you may use a dictionary module to store your data format. Any subsequent applications that you build which require that format may use the definition stored in AVS's data dictionary.

The user is allowed to develop their own modules in either FORTRAN, C, or C++. AVS allows the definition of co-routines, as well as parallel modules. The co-routines are allowed to do their execution whenever necessary. Normally, subroutines perform execution only when told to do so.

The keyframe animation allows the user to define key points in animation sequences and the time at which they will occur. The system will then fill in the gaps with appropriately

5

rendered and positioned data in order to generate a smooth movie sequence for video presentation.

Unsupported and user-contributed modules may be obtained from the International AVS Center via anonymous ftp at:
avs.ncsc.org (128.109.178.23)

Limitations

- Use of shared memory and Interprocess Communication may be a bottleneck for data transfer and amount of memory necessary. AVS 6.0 apparently addresses this issue by allowing for Gigabyte-sized files.

- Limited ability to program conditional branch and loop constructs.

- ADIA under Release 4 not as visual and does not support sufficient number of data types, as in Explorer's DataScribe.

- Data probe appears to pick only individual points; no user-defined geometric regions.

Data Explorer

Data Explorer is a package from International Business Machines (IBM).

Cost is $5,900 list for a single user, non-floating license, single user floating is $7,400, Tentative Government licensing fees are, $4130, single user node locked, $5180 single user floating. Multi-user concurrent floating licenses are available at additional cost.

Runs on IBM RS/6000 systems, Hewlett-Packard, Silicon Graphics, and Sun Microsystems, Inc., and DG Aviion computers.

Synopsis

- Provides full control over the animation rate.
- Records images to files. In either RGB, TIFF, PostScript.
- Support for data probes.
- Several rendering modes, dots, wireframe, volume.
- May automatically generate axes with tick marks.
- Extensive, well written documentation.
- May zoom, rotate, pan.
- Supports perspective and orthogonal views.
- May texture map data from one parameter onto another.
- Supports a colormap editor.
- Support for macros (module groupings).

- Allows user to build their own control panel.
- May generate parallel execution modules.
- Support for regular, curvilinear, irregular grids and unstructured data.
- Data types are real, complex, scalar, vector, tensor, byte, integer.
- General importer for arrays, either ASCII or binary data.
- Handles dx (data explorer) and netCDF format files.
- Easy to use.
- Support for scripting language.
- Extensible.
- Extensive On-line hierarchical and cross indexed help.
- May add your own comments to your visual program.

Discussion

Data Explorer is an offering from IBM. It utilizes a visual program editor to implement a data flow paradigm. Their system utilizes the client/server model in order to implement execution; as such, the server may be on a different platform than the client (end user). Figure 3 shows a Data Explorer session depicting portions of the Visual Programming Environment (VPE) and a display image depicting cloud water content.

The system is easy to use. Without having read documentation, the author was able to manipulate and create new programs that interacted with provided data. Facilitating this was the on-line help panels.

The on-line help is quite extensive, internally cross referenced and context sensitive. The hard copy documentation is well written.

The system supports two file formats; data explorer (dx) and netCDF. The data explorer format appears quite robust and has been designed to handle many forms of data including regular grids, irregular, curvilinear, and unstructured data. The data format also supports the use of connections between data points. These connections are useful for irregular data to assist data explorer in determining which points are near-neighbors to a given point.

Also provided with the system is a general array importer which may read either ASCII or binary data. A user-provided template file informs data explorer about the format of the input file. This feature also allows for the import of coordinate information.

The system supports data probes. Multiple probes may be active at any given time.

Data explorer allows one to generate module groupings known as macros. These macros are merely the combination of one or several modules, and provide input and output ports as chosen by the user. This allows the user to combine several frequently used modules

7
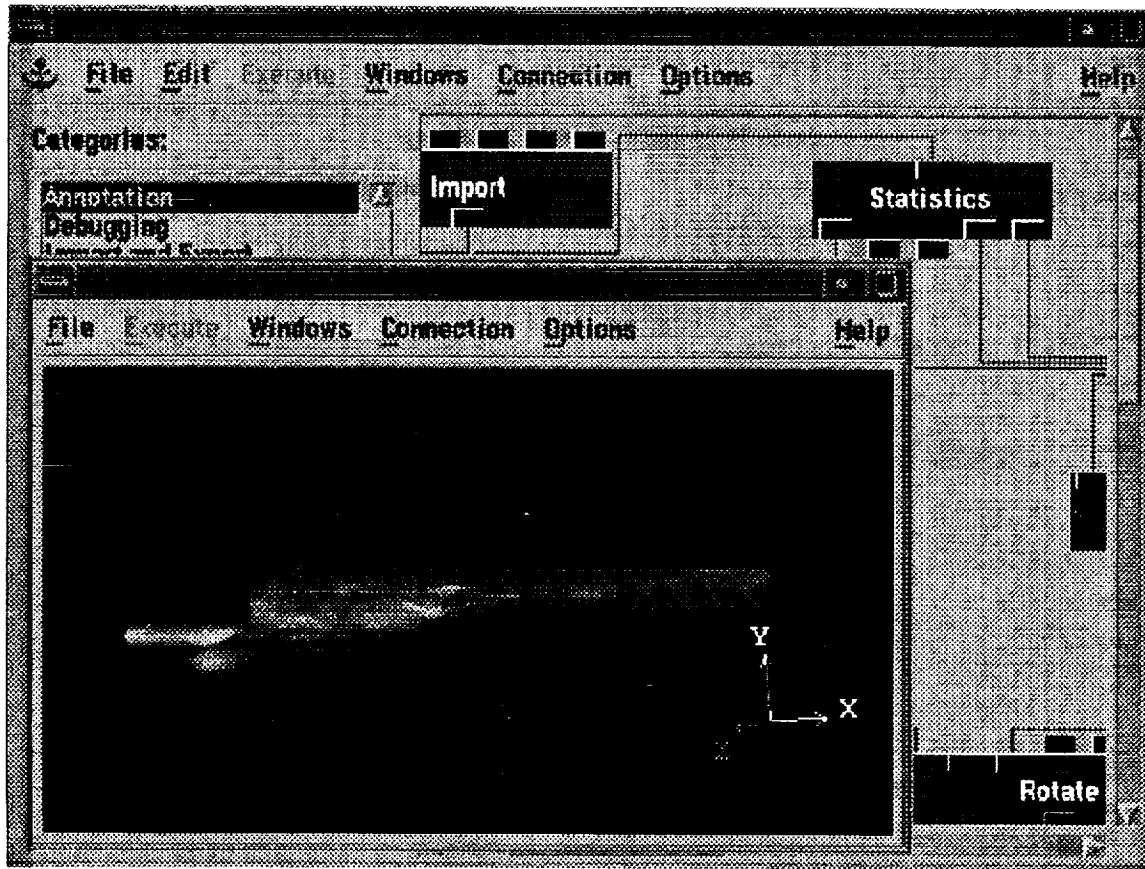
Figure 3. Data Explorer Visual Programming Environment, with an output image window depicting cloud water content.

into one macro, thereby saving time when creating new programs. Macros also allow one to set and/or hide certain parameters. This is useful for setting default values.

The system may records images to files, in either RGB, TIFF, PostScript. This is useful for recording a visualization session. During animation, multiple images are sequentially stored in the specified file. This file may then by played back at a later time.

Several rendering modes are supported. Included in these are wireframe, dots, isosurface and volume rendering. It also appears that the system attempts to determine which mode to use based on the data set. Depending on the platform on which the software executes, the user is allowed to choose either software or hardware rendering modes.

Data explorer provides a calculate module that may be used in conjunction with a sequencer to automatically control animation sequences, chosen isosurface values, etc. The system also allows one to set parameter values in a control panel for interactive control by the user.

The system also provides support for pan, zoom, and rotation of graphics objects. However, one must first open a control panel to choose an option, then use the mouse to set the

rotation. One may not simply with the choice of mouse button directly interact with an object.

Limitations

- Appears that one must open a control panel in order to perform pan/zoom, rotation of objects under mouse control.

IRIS Explorer

IRIS Explorer is a software product from Silicon Graphics, Inc. Explorer comes free with SGI systems. Software support for other vendors is handled by a third party reseller (Numerical Analysis Group Ltd., U.K.) The platforms on which Explorer runs include Cray, Hewlett Packard, IBM, and Sun. Licensing for these other platforms is a few thousand dollars and around $10,000 for a Cray computer. The discussion that follows is based on Explorer Release 2.0.

Synopsis

- Requires Silicon Graphics, Graphics Library (GL), Open GL in the future.
- Visual programming environment.
- Data scribe application to import your data.
- Module builder allows module development in FORTRAN, C, C++.
- Approximately 160 supported modules and approximately 50 unsupported modules are available via anonymous ftp.
- Modules may be selectively disabled by the user.
- Reads Hierarchical Data Format (HDF) raster image and Scientific Data Sets (SDS)-data.
- Reads AVS data sets.
- Strongly typed data types.
- Support for uniform, perimeter (unevenly spaced cartesian grid), curvilinear, and unstructured coordinate systems.
- Extensive conditional branch and looping constructs.
- Camera and lighting control built in, and interacts with looping.
- Scripting language support.
- On-line help.

Discussion

Explorer is a general-purpose visual programming environment based on the data flow paradigm. It provides support for one-, two-, and three-dimensional data sets and appropriate tools to analyze these data. The system is comprised of three main environments: Applications, Prototyping, and Module programming.

The applications environment consists of a map editor and a group editor. The map editor allows one to generate the visual program. This is what the typical user works with in Explorer. The group editor groups collections of modules into one functional unit. It is essentially a macro facility. Figure 4 depicts the map editor of Explorer, with a simple programming network that reads and displays an image. The "Blurimage" module simply performs a blurring function on the image data.
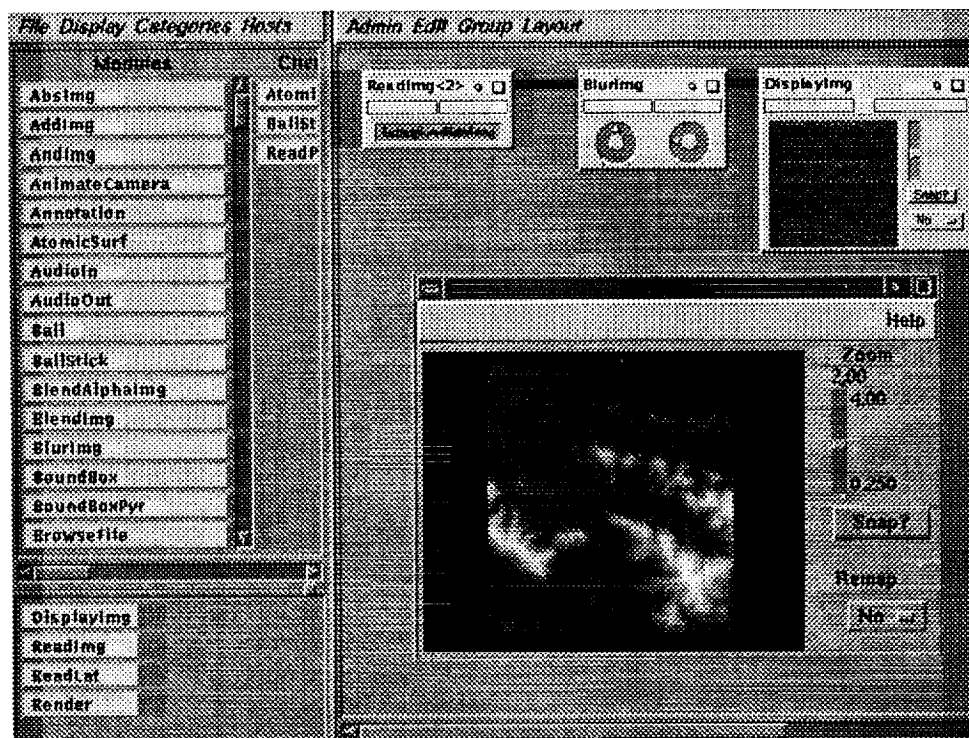
Figure 4. An Explorer map editor session with an image reader, manipulator, and viewer. Note the similarity to the AVS network editor.

The prototyping environment consists of the DataScribe. The DataScribe allows one to generate a script which reads input data or may even export data from Explorer.

The module programming environment is comprised of the module builder and the control panel editor. The module builder allows the scientist to write his own applications module which may be included with the pre-defined Explorer modules to define a program. The control panel editor allows one to change the look and feel of a control panel, or to build one's own. You must use the Explorer-provided widgets such as the sliders and dials. This

interface may even be designed so that the end user does not realize Explorer is executing in the background.

Four data types are supported by Explorer: lattice, pyramid, geometry, and picking. Lattices may be uniform grids, perimeter grids, or curvilinear grids (Figure 2). Coordinate information must also be specified with the data where necessary. The pyramid data type is associated with finite element data and may be used to describe molecular data sets. The geometry type is used for storage of the geometrical primitives for the rendering process. The picking data type is used in conjunction with the data probing aspects of Explorer. The user is also allowed to define their own data types in a C-like syntax. However, when you design your own data type, you are responsible for all the interfaces to other modules and existing data types.

Explorer uses a distributed firing algorithm, which at times may be a nuisance, as there is no central control to disable module firing. Rather one must disable the modules themselves. Each module resides in a separate source file. As such, modules may not share information via common storage. They must share information via direct module to module communication either via shared memory or through interprocess communication. Explorer will work in a heterogeneous environment with the interprocess communications mechanism. Since each module is an individual process, the Explorer system tends to be a systems resource hog. Also, since it relies more heavily on interprocess communication than AVS, it has more open files. It seems, however, that through module grouping, one may effectively combine several modules into one process. These heavy uses of the system resources also mean that less memory is then available for the scientific data.

One of the greatest strengths of Explorer is the built-in control flow logic. Support exists for "case," "for," and "while" constructs. The looping constructs may also be nested. For example, perhaps one would like to perform a time animation in an inner loop, while in an outer loop modify the values of an isosurface.

Explorer also has modules that allow one to animate the camera view and to create movie sequences. A great deal of support is also available for multi-media options including audio. This also includes support for video output, video LAN output, and video frame recorders (Abekas and others.)

Unsupported and user contributed modules may be obtained at the following sites via anonymous ftp:
ftp.epcc.ed.ac.uk (primary site; NAG Ltd., United Kingdom)
swedishchef.lerc.nasa.gov (mirror site; NASA Lewis)

Limitations

- Currently requires Graphics Library or GL emulation software or firmware in your computer.

- A system resource hog. Every module is a process. All data transfer is done through shared memory or interprocess communication (open files).

- Limited number of available modules at this time.

- Data probe (picking) works only with the pyramid data type.
- Does not handle large data sets well.

Khoros

Available free from University of New Mexico. Khoros is a visual programming environment based on a data flow paradigm, although Khoros routines may also be employed using more conventional programming methods. Khoros is comprised of a suite of tools. The software is open; that is, one may obtain it from Khoros, but may not freely distribute the code, nor remarket without the express permission of the Khoros consortium. Discussion is based on Release 1.0 and notes related to Release 2.0.

Synopsis

- Runs on UNIX platforms with X Windows 11.
- Source code is provided with the system.
- Release 2.0 will support HDF and Common Data Format (CDF) data formats.
- May compile Khoros applications to run independent of the Khoros environment.
- Accepts large data files (100 - 500 MB).
- User may write their own modules in FORTRAN or C.
- May perform distributed computing, but temporary files on each machine must have the same pathname.
- Large number of image processing modules.
- Collaboration possible (concert).
- Conditional branch and loop constructs are supported.
- On-line help.

Discussion

The Khoros package is comprised of several parts, some of which are cantata, concert, and the module builder. It is another of the data flow paradigm packages. While the package is free to the user, customer support is only available via bulletin boards, news groups, or e-mail. Telephone support is not provided. In order to receive telephone support for the software, you must join the Khoros consortium. The membership fees begin at $5,000 and continue to $50,000. Each of these provide telephone support and pre-releases of the software. Training credits are also received with the larger membership fee.

A major difference between the Khoros system and the other data flow paradigm-oriented environments is the ability to generate applications that run independently of the Khoros kernel. After testing and verifying an application under the cantata environment of Khoros, the user may save this grouping as an executable. The advantage of this is that

you no longer have the cantata window environment executing with your application, saving some additional overhead.

A Khoros 1.0 cantata session which allows one to view an image is given in Figure 5. As seen from the figure, glyphs are used to represent programming modules; and are connected via a pipeline. Available data are indicated by dots next to the input or output ports (arrows). Program modules are available under the routines button, or are available via the category buttons at the upper portion of the window.
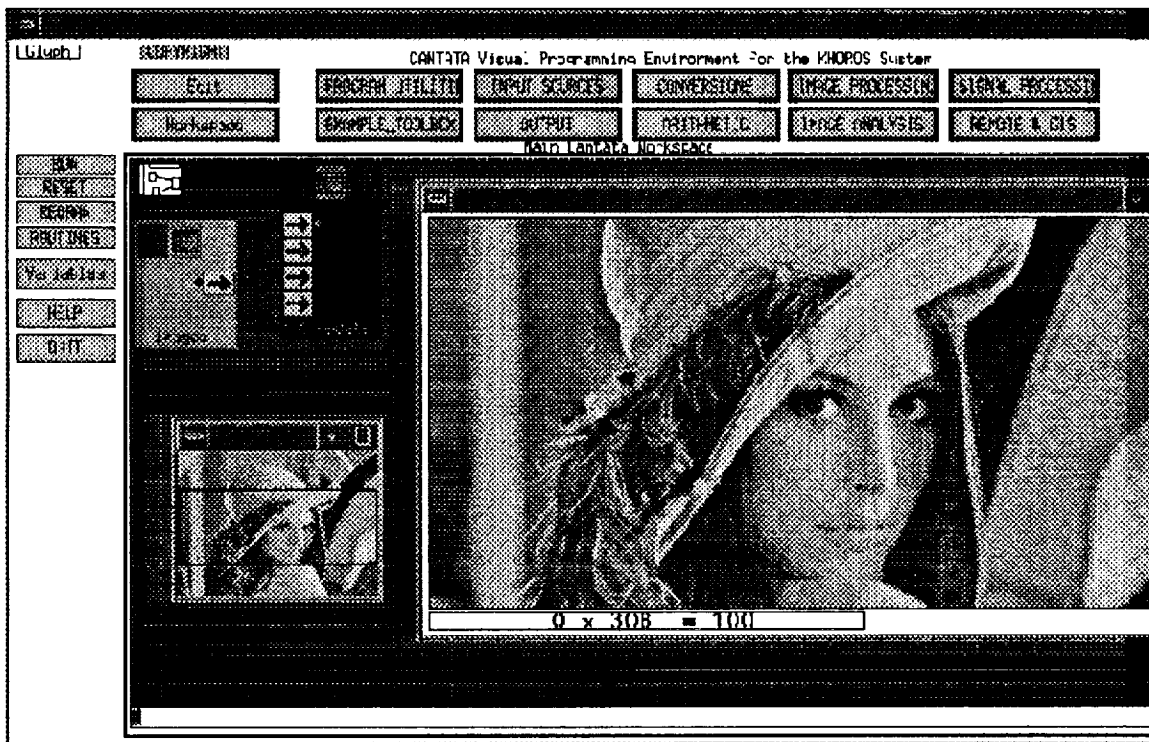
Figure 5.   The Khoros cantata workspace under version 1.0. This program reads an image and displays it in an output window. Module input and output ports are denoted by arrows and are connected via a pipe.

Another strength of Khoros is the broad range of support given to image processing within its environment. A large number of modules are provided. The International AVS center has even wrapped these modules with AVS.

The cantata interface has support for the use of variables. This allows for branch and looping constructs.

Khoros supports distributed computing; however, the user must set up temporary files on each machine, and these must have the same pathname on each machine. It also supports collaborative science via the concert mechanism. With Release 1.0, the collaboration tends to bog down the network. All data and command manipulations are sent over the network.

Under Khoros 1.0, there appears to be no support for timing based applications, animation rates, etc.

The Khoros system supports shared memory, interprocess communication, and disk file connections for inter-module communication. The disk connection allows users to utilize large data files.

NASA's Tropical Rainfall Measurement Mission and the Earth Observation System Data and Information System plan to use Khoros as a generic applications environment.

Khoros 2.0 will not rely strictly on the internal Khoros VIFF format; it shall work with HDF and CDF formats, which will be embedded in the data services layer.

With Release 2.0 of Khoros:

- Entire system has been redesigned.
- Time sequencing has been better handled.
- Motif window manager has been implemented.
- Data services support is purported to be able to handle any data format, provided it already understands your data format (data dictionary?)

Limitations

- Limited support for three-dimensional rendering in Release 2.0.
- No support for animation.
- Khoros does not want you to move binaries between machines; you must obtain source code for each machine and have it resident. Then you need to recompile the application for each machine on which you wish to run the application.
- No data ingest utility like ADIA, DataScribe.
- Large package; requires over 100 MB disk storage.

LinkWinds

The Linked Windows (LinkWinds) system is currently under development at NASA's Jet Propulsion Laboratory. It is funded solely by the Federal Government, and as such is available free to the public. LinkWinds is still in Beta testing so it is not yet widely released. They will provide binaries of the code, but only to individuals that seek them out and request the software. The tool, even in a beta state, is highly functional and mostly bug free.

Synopsis

- Extremely easy to use.
- May redesign user interface.

- Animation control.

- Movies from screen dumped images.

- Journal and playback support.

- Data probe support.

- Cursor synchronization.

- Screen dump support.

- Mathematical editing.

- May connect nearly any control to any analysis widget.

- Collaborative science with the Multi-User Science Environment (MUSE) capability.

- Reads HDF data sets.

- Object based, so should be extensible.

- Support for missing and invalid data values.

Discussion

The LinkWinds system currently runs on Silicon Graphics platforms and systems that support GL graphics. Porting of LinkWinds to the X-Windows 11 system is nearly complete. The system was designed by a scientist to perform scientific analysis. The concept is to have a set of analysis and control tools that may be linked to form an analysis environment. One simply opens up a tool and links tools (windows) and data sets together to create an analysis environment. Figure 6 shows a typical LinkWinds session. Depicted in Figure 6 is ocean temperature at 100 meters. The slider tool controls which image is being viewed. The data set in this case is three-dimensional with two dimensions representing images and the third dimension depicting ocean depth.

The main strengths of LinkWinds lie in its ease of use and its high degree of interactivity. It is also able to easily combine, compare and analyze the relationship between different parameters or different data sets. LinkWinds is best suited to the analysis of three-dimensional gridded data, which may consist of two spatial dimensions and a third dimension related to time, spectral band, or collection of variables.

Due to the object oriented nature of the package, the scientist may connect any control widget to any number of tools. This allows the control functions to simultaneously control one or more analysis sessions. For example, one may have a slider control, which controls time animation linked to data sets of temperature and atmospheric pressure. As the slider is moved, sequential time steps for each data set may be simultaneously observed.

Native data formats for LinkWinds include raw byte data, CDF, HDF, and SGI's rgb image format. The raw byte data are for support of image data, and strictly data streams with no header information. A user-supplied database file informs LinkWinds about the image size and number of images. The CDF format is a recent addition and is not yet fully supported. HDF support is for raster 8 image and for the SDS formats.
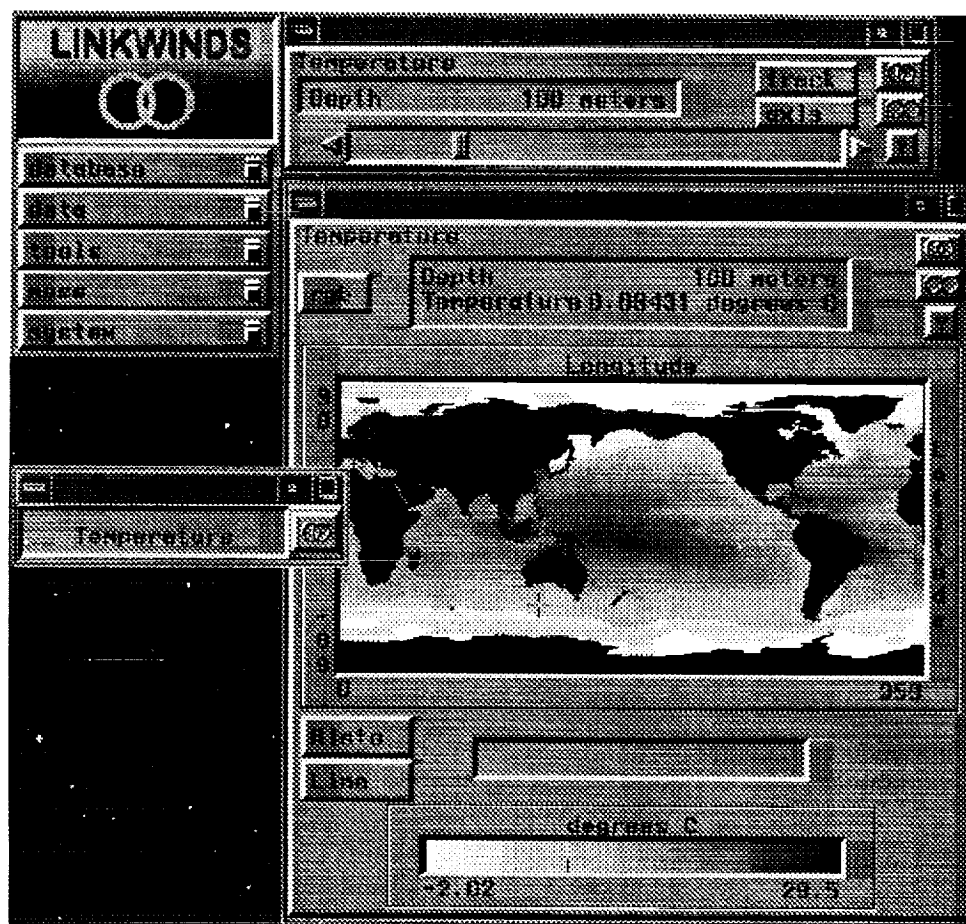
Figure 6. A simple LinkWinds session including an ocean temperature data set, slider, image viewer, and the menu panel. The slider controls the viewed image.

LinkWinds sessions are maintained in a journal. This allows a user to replay an analysis session. This is useful to continue work where left off, as well as for demonstration, debugging and training purposes.

Tools provided in LinkWinds 1.4 are as follows:

Control	Display and Control	Display
Slider	Combine with Calculator	Plane
3-Axis Slider	Compare	Globe
3-Axis Rotator	Data Subset	Orthoview
2-Axis Rotator	Line Plot	Profile
Pan-Zoom Slider	Histogram	2D Scatter Plot
Combine Slider	Image	3D Scatter Plot
Animator		Track Pixel
Frame Animator		
Color Tool		

The MUSE capability allows one to network LinkWinds sessions. This is useful for collaborative science, as well as for training purposes. A session may be receive only, send only, or both receive and send. MUSE only sends control information on the network. Each workstation in the MUSE session must have the same data loaded in order to participate, however, the benefit of this scheme is less network traffic and increased interactivity.

Cursor functionality includes the ability to synchronize cursors between display windows. This allows the user to see which features correspond among related data. Also provided with the cursor is a probe capability. The data value under the cursor always tends to be displayed in the output window. One may also define regions of interest and perform calculations based only on that region.

Limitations

- Currently only works with machines running GL graphics. Development nearly completed for an X-Windows 11 version.

- Presently must set up templates to read one's data set. This is being improved to take advantage of encapsulated information within HDF data sets.

- Must have the same data set on each machine in a MUSE (collaborative) session.

McIDAS

The Man-computer Interactive Data Access System (McIDAS) was developed at the University of Wisconsin, Space Science and Engineering Center. The tool has been transformed and developed for more than 20 years. The original intent of this package was to derive wind fields by tracking cloud tops from satellite imagery. The system is currently undergoing another transition which utilizes X Windows 11 and UNIX, rather than proprietary display systems. There are a variety of licensing fees for this system, depending on hardware platform, number of users, etc. This license is collected in the form of a users group fee.

Synopsis

- Primarily designed for meteorological satellite data analysis.
- System runs on IBM mainframe with proprietary workstations (with the MVS operating system and is being phased out), IBM RS 6000, Sun, and Silicon Graphics platforms under UNIX and X Windows 11.
- Handles satellite imagery (typically visible and IR channels) from GOES, NOAA polar orbiters, GMS, Meteosat.
- Database support for multi-spectral imagery.
- Ingests standard weather wire products, NMC model outputs.
- Allows for overlay of data either as plotted values or contours onto satellite imagery.
- Support for georeferenced map overlays.

- Good animation support.
- Supports scattered data, gridded data.
- User may write their own applications.
- Data probe support.

Discussion

McIDAS is primarily a data management facility. Its analysis and visualization capabilities are tuned for the atmospheric scientist, so it is not necessarily a general-purpose tool. Three basic types of data structures are supported by the McIDAS system. These are image, grid, and scatter data. The image data structure is designed to handle two-dimensional images of multi-spectral data sets. Figure 7 shows an 8 km visible image of North America, with a map overlay, from the McIDAS-X system. The internal data structure supports both 1 and 2 byte data. It also supports calibration information for each scan line of data, as well as navigational support. The gridded data structures support a few simple map projections: rectilinear, polar stereographic, and pseudo mercator. The data files for the scatter data are treated as two-dimensional arrays (time vs. location). Software is provided in the system to either plot the scatter data in some graphical map projection or overlay it onto satellite imagery. One may also transform this scatter data into a grid in order to obtain contour plots.

Satellite products ingest for the system is dependent on which satellite receiving antennas are available at a site, or networking of your local McIDAS with other McIDAS nodes that may receive a product. The same is true of the weather wire data ingest. Various data streams are available: one contains standard weather observations, another contains numerical model grids from the National Meteorological Center, and yet another circuit will provide weather radar information.

The greatest strength of the McIDAS system is its support for animation of satellite imagery, and/or graphical animations. The system supports animation rates (on proprietary hardware) of 1/15 th of a second per frame increments. The X Windows environment timing is not quite as accurate. The user may individually set the dwell rate of each frame for the animation. This allows the user, for example, to set the time spent on the last frame of an animation sequence to be longer than all other frames.

A user may write their own applications that access the McIDAS data base. These applications are most likely written in FORTRAN 77, as most of the McIDAS routines are written in FORTRAN. Some of the lower level McIDAS routines now are written in C, and there may be support for a C language interface in the future. The system does not provide software to assist in the development of modules. New code is developed by modifying existing McIDAS applications, or by developing from scratch. One may also create macros which call various McIDAS command line-driven functions.

The McIDAS-X allows one to dynamically define new image frames and to generate them at different sizes. This allows one to display images larger than 640 by 480 pixels. One may also pan images larger than the display window.
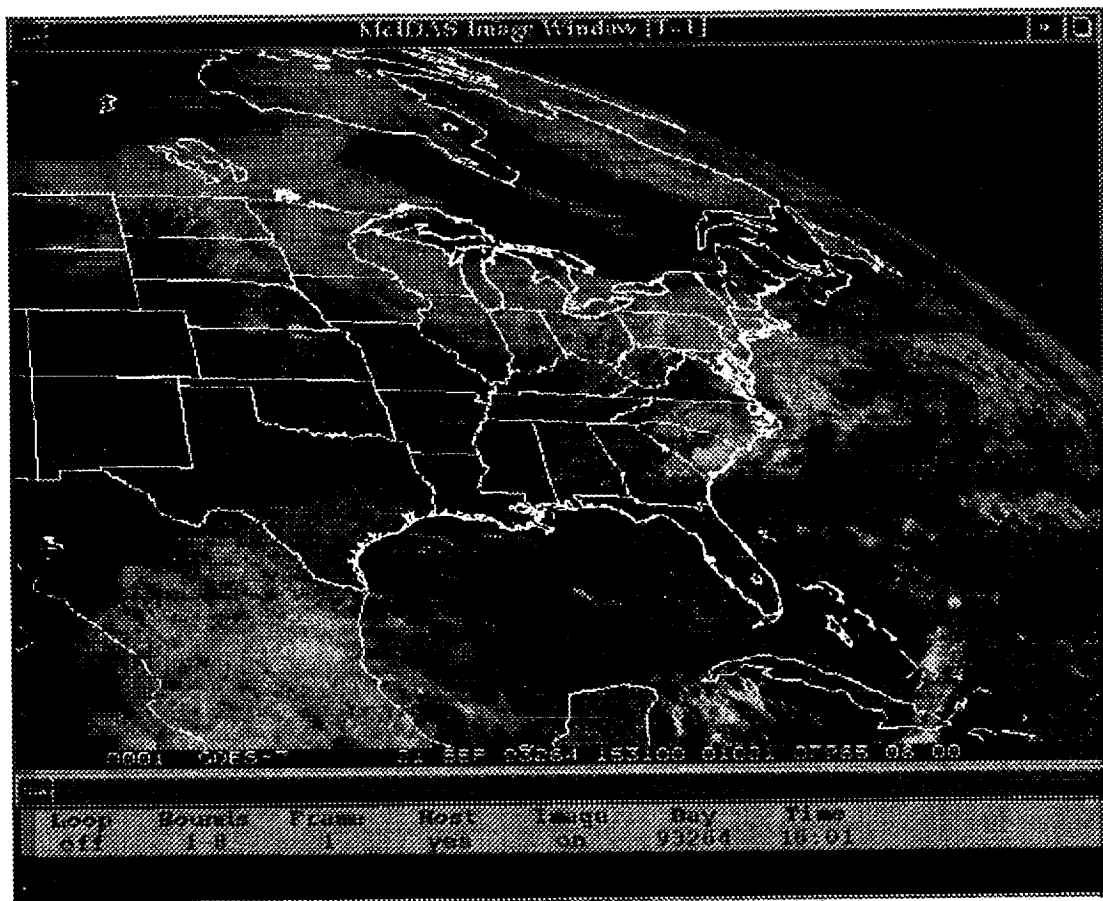
Figure 7. An 8 km visible image of North America from the GOES satellite as displayed with the McIDAS-X system. The command window is also shown. The textual output window is not shown.

Also provided with McIDAS is an ability to probe satellite data under a cursor. This may be done as a single point value or as an array of values under the cursor (based on the size of the cursor.) Also, if calibration data are available, the user may return the values either as pixel brightness, raw values, or a calibrated value such as temperature.

Limitations

- Currently only displays 8-bit grayscale imagery. The grayscale imagery may have a pseudocolor map applied. No display support for true color imagery.

- Only one output display with the proprietary McIDAS display and the X Windows environment. As the X Windows system matures, there will hopefully be support for multiple viewports.

- Limits on file sizes. This is mainly based on historical problems with computer architectures.

- Currently does not support the HDF standard.

- No programming development environment. Library support only.

SciAn

Developed by the Supercomputer Computations Research Institute (SCRI) at Florida State University under funding from the Department of Energy and the State of Florida. This package is an object-based, three-dimensional visualization application. The purpose of the package is for Scientific Animation; hence, the name SciAn. It is available free to the public. The following discussion is based on SciAn Release 0.831.

Synopsis

- Runs on Silicon Graphics or IBM RS 6000 with GL graphics.
- Excellent documentation.
- Accepts a wide range of file formats including: HDF, NetCDF, PLOT3D, Neutral File Format (geometry objects), Simple Text Format, Protein Data Bank, and others.
- Allows the user to set preferences.
- Handles both two- and three-dimensional grids which may be regular, separable, curvilinear, or unstructured.
- Data of disparate sizes and types may be combined.
- Reads and interpolates time-dependent data.
- Support for missing data flag.
- Selectively chooses the type of visualization based on the data type, or the user may choose via a menu.
- Supports multiple lights and rendering modes.
- Supports multiple rendering spaces.
- Rotation of objects may be done in either a wireframe or polygonal mode.
- Cross sectional slicing.
- Contour plotting.
- Supports a 2D background panel, 3D rendering panel, and a 2D foreground.
- Support for constant animation rates.
- Support for single frame recording to video devices.
- Scripting language support.
- On-line help.

Discussion

This package is useful for a wide variety of scientific disciplines including the earth sciences. The authors of this package have written excellent documentation. Perhaps the greatest strength of the package is that it is designed around the object oriented principle. This gives the user great flexibility in determining what type of data he/she desires to visu-

alize. For example, one may texture map radar data onto a terrain data set by dragging the radar object into a terrain color panel. A sample SciAn session depicting radar reflectivity over a terrain map is given in Figure 8.
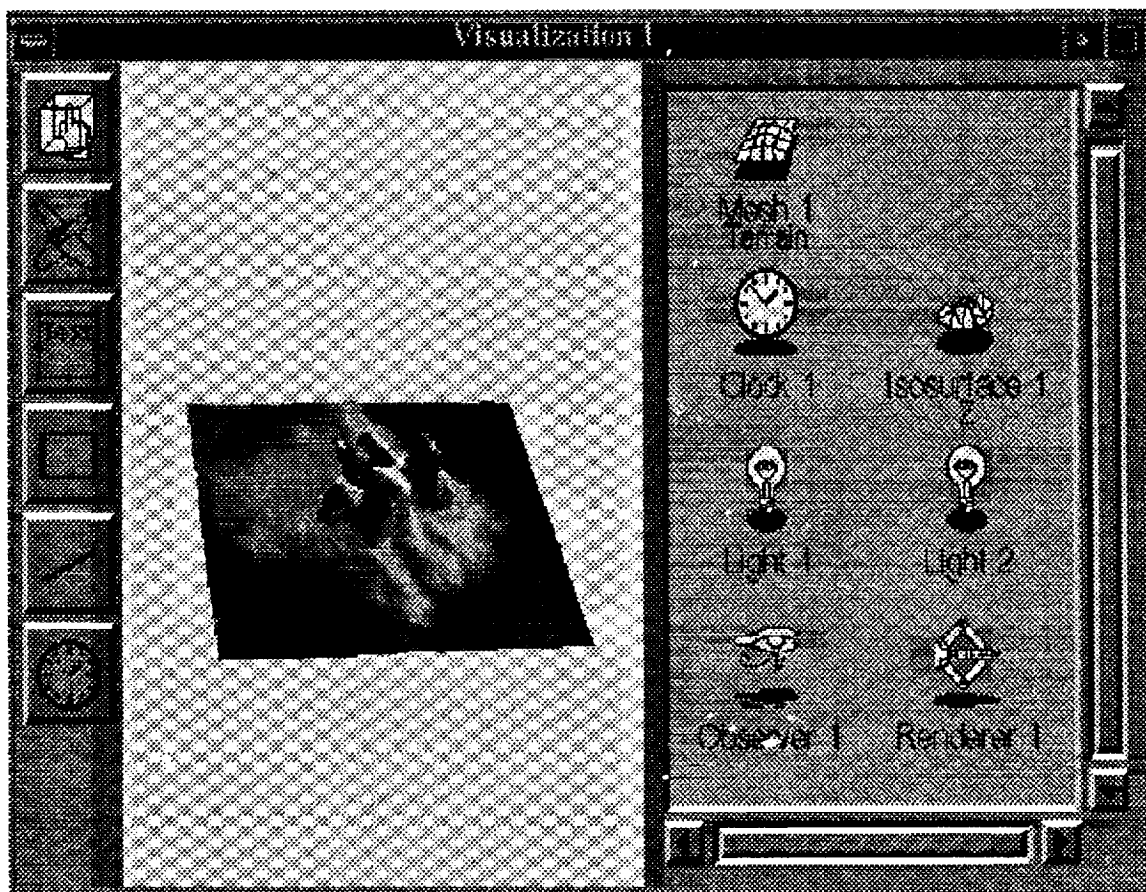
Figure 8. SciAn visualization window with control panels. A radar reflectivity isosurface is depicted with surface topography values.

Another great strength of the package are the numerous file formats accepted and written by the package. The NCSA HDF SDS may be utilized, as well as the NetCDF format. The PLOT3D format is also accepted. In order to support these data format libraries, the user must obtain the packages from their respective sources. These libraries are optionally linked against the SciAn package, dependent on whether they reside on the computer. Many other data sources are available including: Protein Data Bank, Neutral File Format (geometry objects), Gaussian 90, and Simple Text Format.

The package may combine data of disparate types and sizes. For example, the data from Figure 8 consist of a two-dimensional terrain data base with a size of 45 by 45, and a three-dimensional data set of radar reflectivity with dimensions 30 by 30 by 20 which covers approximately one-fourth the area of the terrain data base. Should one have time-dependent data sets, SciAn will interpolate data for the missing time steps. This ensures that animation sequences are synchronized.

In order to control the placement of disparate data sets, one must use the same coordinate system for each data set visualized. Currently there is no built-in support for georeferencing in the strictest sense; it is more the user's responsibility. For example, in order to do a map overlay, this would be an independent input data set, which would necessarily need to be referenced to the same coordinate space as the other viewed data sets. As SciAn is a package meant for a wide range of disciplinary fields, this specific feature for the earth sciences has not yet been incorporated. They are working on a method to handle this and are even working toward handling different map projections.

Coloration of surfaces may be either a user selected color or optionally may be based on values from the data set or, for that matter, from another data set. The terrain data base from the figure is pseudo-colored with the elevation data from the terrain data set. Optionally, the user could have chosen the radar data set to determine the coloring of the underlying terrain, or any other data set for that matter.

Due to the object oriented nature of the package, one may easily generate new isosurfaces, meshes, or slices. Simply duplicate the object to obtain a new visualization. Optionally, one may also create another visualization window to examine the data.

Object manipulation may be performed in a variety of manners. When rotating an object one may either rotate in a high resolution mode where the polygons are always rendered while manipulating the view, or it may be done in a bounding box mode. A third option "automatic," is also available which determines the complexity of the scene and determines which of the above modes to use. The user may set one of these modes as a preference.

Cross sectional slicing is available in SciAn. One may generate multiple slices in any orientation, but for each object may only choose one orthogonal axis in which to cut the cross section. Currently there is no capability to define arbitrary cross sections; however, it is on the list of things to do.

Animation of rendered frames is under precise control. Each visualization object may have its own clock or may use another visualization object's clock such that two objects are under the same clock control. Support for the more sophisticated keyframe animation is not yet provided.

Device drivers for several single frame recording devices are included in the SciAn package. Simply direct your output to one of these video objects to perform a recording session of your visualization. Of course, you do need to have one of these devices available and properly connected to your workstation before it will work.

SciAn also supports the JPEG video format. This allows one to store movie sequences on digital disk and play them back later with SGI provided software. Various screen capture capabilities are also provided.

The package currently lacks a data probe capability. They plan on incorporating a probe, but are in the process of determining the design ramifications of single point vs. multiple probes and whether or not the probe may be active during animation sequences.

The SciAn developers are looking at the feasibility of adding collaborative analysis capabilities to their package.

SciAn currently does not allow the user to extend the analysis capabilities of the package. They have not yet designed as clean of a interface to accommodate this as they have done with the file readers. In support of user extensibility, however, they are adding a calculator module which will allow calculation of simple parameters. They are looking at adding to this a generic C and FORTRAN interface that would allow users to link in their own analysis modules.

SciAn is available free via anonymous ftp from ftp.scri.fsu.edu (144.174.128.34) in the pub/SciAn directory.

Limitations

- It appears that the user is responsible for georeferencing of input data, as all rendered data are centered in the output display.
- No clear means to handle overlay of georeferenced map information.
- No capability for non-orthogonal cross sections.
- No current support for a data probe.
- Must have a workstation that has a Z buffer.
- Currently does not allow a user to extend the package. They have not yet designed an interface.

VIS-5D

Developed at the University of Wisconsin, Space Science and Engineering Center for the Marshall Space Flight Center, this package allows for interactive visualization of five dimensional (spatial, temporal, parametric) data sets. It is based on the data being memory resident. This package is free to the public. The following discussion is based on the VIS-5D Release 3.2.

Synopsis

- Runs on Stardent GS2000, Silicon Graphics, IBM RS 6000 (select models.)
- Limited extensibility via user-defined analytical functions.
- Handles rectilinear grids only.
- Support for animation of time series data.
- Able to generate multiple isosurfaces of any parameter or several parameters.
- Horizontal and vertical cross sectional slice support. In shaded image form as well as drawn contour lines.
- Wind vector depiction.

- Wind trajectories.

- Topography and map overlays.

- Color and transparency under user control.

- May read HDF SDS format data sets.

- Volume rendering support on Silicon Graphics machines with VGX or better graphics.

- Single point data probe.

Discussion

VIS-5D was designed specifically for the earth science community. It has seen uses in the overview of cave passages and medical imaging. The major limiting factor to the package is the rectilinear coordinate system. There are some limits as to the size of the input grids, but those are somewhat artificial, based on current limitations regarding physical memory on machines. A strength of the package is the ability to animate time series data sets and provide highly interactive control over three-dimensional rendering. Another is its ability to combine any visually sensible sequence of analyses in order to support the understanding of a data set. For example, the display may include isosurface rendering, a horizontal contour plot, underlying topography, and wind trajectory analyses. An example VIS-5D session output is depicted in Figure 9. Cloud liquid water (isosurface) and a three-dimensional wind vector field from a numerical model of a thunderstorm are shown.
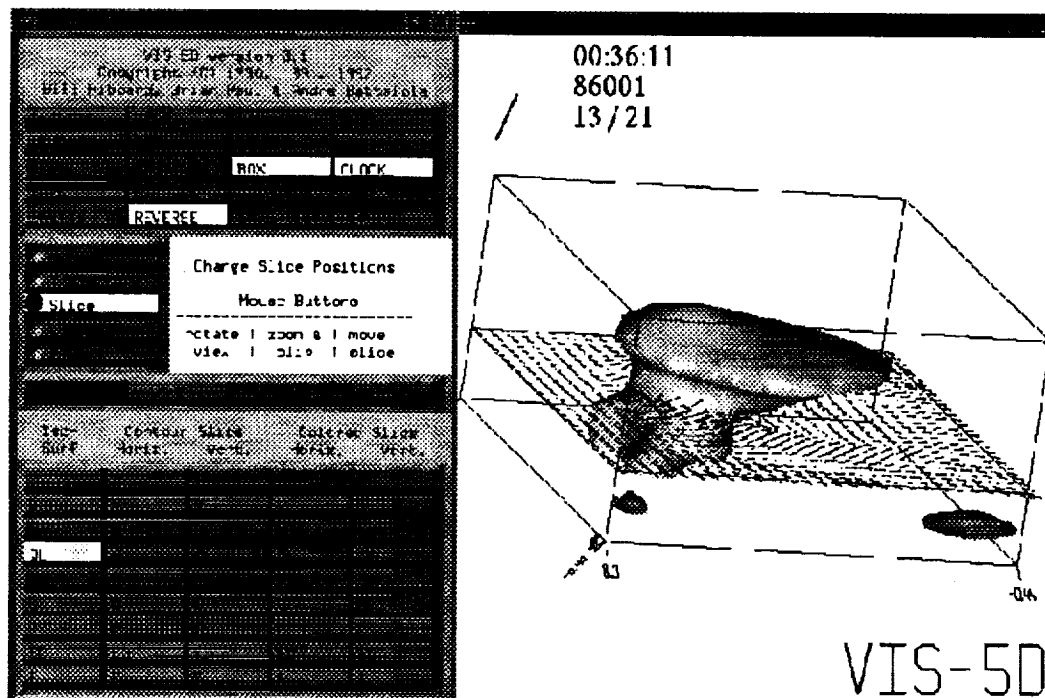

Figure 9. A VIS-5D session depicting an isosurface of cloud liquid water and three-dimensional wind vectors at a constant altitude. Slider controls are not shown.

The greatest strength of the package is that it is highly interactive and efficient. The rendering capabilities do not automatically "fire" as in some of the data flow paradigm packages. Rather, the user has control over when generation of isosurfaces is performed. This is efficient because you can set slider values, and fire the rendering process only when you are happy with the chosen value. One may easily change viewpoint by simple rotation of the object with the mouse. You may also zoom into a region for closer inspection. One is allowed to easily combine many different parameters to inspect their relationships. Support for multiple isosurfaces of the same parameter allow one to easily examine storm cores, for example. One may also examine isosurfaces of differing parameters simultaneously. These capabilities are naturally augmented by user control over color and transparency selection.

Volumetric displays may sometimes be difficult to comprehend. VIS-5D allows the scientist to generate cross sectional displays of the volumetric data set. These slices may have either horizontal or vertical aspects. The vertical planes may also be turned at oblique angles, orthogonal to the horizontal plane. Grips are provided to move the slices through three-dimensional space. Support for random slices, however, is not available. These slices may either be depicted as contour plots, or as pseudo-colored images, which may also be semi-transparent. The tool also allows one to "clone" the slices to have multiple slices present. It is the user's responsibility to offset the slices and/or set them at differing angles to view some interesting feature.

A variety of wind analysis features are included in the package. This presupposes that U, V, and W parameters have been defined which correspond to the standard meteorological convention. One analysis feature is the inclusion of support for three-dimensional vector plots of the wind field. Two horizontal planes and two vertical cross sectional slices are provided. The user has complete control over placement, color, plot density, and vector length. The other wind analysis feature is the ability to generate up to eight independent sets of wind trajectories. The user places a three-dimensional cursor within the data set, and with the click of a button generates trajectories which pass through that point (Animation must be turned on to generate the entire set). Multiple points may be chosen for each set. These trajectory sets may be colorized and be depicted either as lines or ribbons in space.

The user has a choice of map overlays, or may provide their own. Topography is also included, with an option to allow user-defined topography.

Support has been incorporated to allow the conversion of HDF SDS data sets. The convertor software generates a data set which is readable by the VIS-5D application.

Another great strength and extensible feature of the software is the ability to dynamically include user-defined analytical functions. These functions take the current data parameters as input, which then generates a new parameter that may be further analyzed by the VIS-5D application.

Most packages allow for some type of output capabilities. VIS-5D allows the user to save current settings such as isosurfaces, placement of slices, etc. The user may also add labels

to the output, and turn features on/off. An option is provided to allow the user to save the graphical display to an output file. The screen-dumped image format depends on the platform on which the software is running. A "pretty" button is also included to do a better rendering of the graphic display.

The software is available via anonymous ftp from vis5d.ssec.wisc.edu (144.92.108.63). It is available as a compressed tar file (6 MB). Compressed tar files of the binaries are also available.

Limitations

- Limited control over the animation rate. May specify a frame rate, but it is more dependent on the number of polygons to be rendered.

- Only one output display per session. May start application more than once.

- No allowance to incorporate two-dimensional grids. They must be stored as three-dimensional grids.

- No type-ins are allowed on sliders in order to set exact values.

- All data must be available at the start of the application. No allowance to incorporate additional time steps once the application has begun.

MISCELLANEOUS PACKAGES

Collage

The Collage software package, available free of charge from the NCSA, allows the scientist to visualize and do some coloration of their data sets. This package is not so much a visualization package, but rather it is based more on a communications protocol which allows scientists to collaborate using a computer network. It also is based on a spreadsheet concept where the scientist may apply mathematical operations on data arrays. The basic idea is that a group of scientists begin a collaborative visualization session, with each having a basic set of tools such as an image viewer, whiteboard, and message board. If scientist A displays a satellite image, scientists B and C will also have that image displayed on their workstation. Any one of them may then use some graphics utilities to outline features, type or draw on the whiteboard, or type in window. Plans exist to packetize voice information on the network such that conference calls are not necessary in conjunction with this session. The package deals mostly with HDF formatted data sets, although it may read in tabular sets of ASCII data.

Envision

Envision is under development at Texas A&M University under NASA sponsorship. The package is designed to manage and display large scientific data sets. Data sets may be combined into projects. The system requires a visualization package to be linked against it in order to properly function, such as NCSA Collage, NCSA XImage, and IDL. The sys-

tem works with HDF and netCDF data sets. Available via anonymous ftp from ftp.vista.atmos.uiuc.edu.

IDL

IDL is a high level programming language designed for array manipulation. It is produced by Research Systems, Inc., Boulder, CO. Mathematical functions included are matrix inversion, singular value decomposition, complex arithmetic, FFT, convolution, differentiation, and integration. Also provided are transcendental and Bessel functions. Statistical routines are provided to support multiple regression, cluster analysis, and non-parametric statistics. Delaunay triangulation is provided to support surface rendering of irregular gridded data. Visualization and plotting support is provided using simple calls. It allows for extensive design of the user interface for developing application-specific programs. This package runs on most UNIX platforms which support X Windows 11. The package is being adopted as a portion of the standard toolkit for use with the NASA Tropical Rainfall Measurement Mission.

PV-WAVE

This system is actually a family of products and comes with the merger of IMSL, Inc. and Precision Visuals, Inc. The company is now known as Visual Numerics, Inc. IMSL was known for its numerical analysis libraries, and Precision Visuals was well known for its visual analysis software (PV-WAVE). The product is supported on most major UNIX workstations, VMS, and Windows NT. Licenses may be either node locked or floating.

PV-WAVE is a family of products supporting data visualization and analysis through the Advantage package. This package ties the IMSL statistical and mathematical tools to visualization. The product has support for numerical analysis, two-dimensional plotting, three-dimensional display through contouring, isosurface rendering, and texture mapping of images to surfaces. The tool also provides image processing support. Data import and export is facilitated through ties with relational data base management systems, as well as spreadsheets and word processors.

Optional products include Maple, an interactive algebra system that supports both numeric and symbolic computations. Also supported is Structured Query Language links to ORACLE and SYBASE relational data base management systems. PV-WAVE GTGRID will generate grids from irregular data sets. If you don't like command line-driven applications, PV-WAVE also has an optional point and click interface.

SpyGlass Transform

Transform is designed for surface plots, contour plots, vector plots, raster imaging, overlays, and analytic functions. It is essentially based on a spreadsheet concept, which allows one to visualize data in a variety of ways, such as generation of a pseudocolored image, contour plots, or line plots. One may also synchronize the cursor to point to the same element within related data sets.

A nice feature of Transform is the Notebook facility. This allows one to document information about a data set as well as to store algebraic equations. These algebraic equations may then be used to calculate new information based on input data sets. To facilitate these transformations, a library of math functions is provided to act on your data set variables. Included are trigonometric, complex number, kernel functions, and matrix manipulation.

Transform is primarily designed to work with two-dimensional data sets. It does, however, read three-dimensional HDF files one slice at a time. Formats accepted by the package include HDF SDS, and HDF raster image, ASCII, ASCII spreadsheet, TIFF, and FITS image files. Output is provided for HDF formats and as images. Support is also provided for PostScript printers.

SUMMARY

No one visualization or analysis tool will meet all the needs of the researcher. Each tool adopts a different approach, and because of this does not perform well under certain situations. Nor does each tool support all data formats. Some even limit the amount of data the user may ingest. Some of the more successful and useful tools support user extensibility of the package. It also seems that a great deal of the packages now support the HDF format which is being adopted by the EOSDIS project. This is quite useful to the scientist, because now he may concentrate on analyzing his data and not spend time reformatting data for another package. Now the scientist may effectively use distinctive analysis features provided by two or three different packages to conduct research.

This paper in no way discusses all the visualization and analysis packages available to the researcher. Nor does it discuss all the capabilities within each package. Rather, it summarizes capabilities of packages with which the author is familiar. It is hoped that this will allow the reader to become further aware of these packages and their capabilities.

LIST OF ACRONYMS

AVS

Application Visualization System

CDF

Common Data Format, developed by the NASA Goddard Space Flight Center

EOSDIS

Earth Observation System Data and Information System

GMS

Geosynchronous Meteorological Satellite

GOES

Geosynchronous Operational Environmental Satellite

HDF

Hierarchical Data Format

MUSE

Multi-User Science Environment

NCSA

National Center for Supercomputing Applications

NMC

National Meteorological Center

NOAA

National Oceanic and Atmospheric Administration

SDS

Scientific Data Set - a subset of the HDF format

GLOSSARY

Co-Routines

A construct to simulate parallelism on a uniprocessor by explicitly interleaving the execution of a set of concurrent units. They do not return as a subroutine does, but may explicitly activate (resume) execution of one another, while maintaining their local data. They are typically under their own control, whereas a subroutine must be explicitly invoked. When resumed, these modules continue execution where they left off.

Data Flow Paradigm

A model in which data move from one process to another, where each process transforms the data in some manner.

Fire

A module that executes or is acting on data is said to fire.

Glyph

A pictorial representation of an application, or an element of an application.

Interprocess Communication

A mechanism where different processes may communicate with one another by listening on special ports or "sockets." This mechanism allows heterogeneous machines on a network to share data. Data are formed into packets and actually moved between the processes.

Shared Memory

Another interprocess communications mechanism which allows processes to share information. This mechanism typically works on the same machine, with the system kernel refereeing access to the memory space. No data transfer takes place; both processes maintain pointers to the beginning of the shared memory segment and negotiate access with the kernel.

Strongly Typed

This refers to programming data types. For example, one may not pass a floating point representation of a number to a module expecting an integer representation. In this context it also refers to user-defined data types such as C programming language structures.

Upstream

In a data flow paradigm, the data tend to flow in one direction among modules. Upstreaming allows certain items of data to be sent upstream (against the flow) to a module which executed before the current module. Items such as data probes and loop control utilize this mechanism.

Visual Program

A program in which the programming language is represented by graphical icons (glyphs) which the user may then connect in some fashion to form a program.

Wrapper

Typically this may be considered interface code. For example, the window manager on your computer is a wrapper around the software you are executing. Via widget buttons the user controls execution of the software. The internal interface to the software is hidden from the user.

X Windows 11

A protocol specification developed by the Massachusetts Institute of Technology. This is an event-driven system that primarily serves as a graphics display tool on most UNIX window-based computer systems. Programs also exist to act as terminal sessions on the computer.

BIBLIOGRAPHY

An Overview of Volume Visualization Software, 1992, Scott Dyer Lamb & Company, Course Notes 8, Introduction to Volume Visualization, 19th International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH, 228-249.

AVS™ Developer's Guide, 1992, Advanced Visual Systems, Inc., Part Number 320-0013-02, Rev B.

AVS™ User's Guide, 1992, Advanced Visual Systems, Inc., Part Number 320-0011-02, Rev B.

IBM Visualization Data Explorer User's Guide, 3rd Edition, 1992, Part Number SC38-0496-02.

Image Processing Short Course with Khoros™, 1992, University of New Mexico.

IRIS Explorer™ Module Writer's Guide, 1993, Silicon Graphics, Inc., Document Number 007-1369-020.

IRIS Explorer™ User's Guide, 1993, Silicon Graphics, Inc., Document Number 007-1371-020.

Khoros™ Training Course, 1993, Khoral Research, Inc.

McIDAS Applications Programming Manual, 1993, Space Science and Engineering Center, University of Wisconsin - Madison.

McIDAS-MVS User's Guide, 1993, Space Science and Engineering Center, University of Wisconsin - Madison.

McIDAS-X User's Guide, 1993, Space Science and Engineering Center, University of Wisconsin - Madison.

Overview of the Linked Windows Interactive Data System (LinkWinds Version 1.4), 1993, Allan S. Jacobson, NASA Jet Propulsion Laboratory MS 183-501.

SciAn User's Manual, 1993, Eric Pepke and Jim Lyons, Supercomputer Computations Research Institute, Florida State University.

SciAn Reference Manual, 1993, Eric Pepke and Jim Lyons, Supercomputer Computations Research Institute, Florida State University.

Visualization and NASA in the Coming Decades, 1992, R. Weidner, M. Lee, K. Dozier, NASA Jet Propulsion Laboratory.

APPROVAL

SURVEY OF VISUALIZATION AND ANALYSIS TOOLS

By

P. J. Meyer

This report has been reviewed for technical accuracy and contains no information concerning national security or nuclear energy activities or programs. The report, in its entirety, is unclassified.

Gregory S. Wilson
Director, Space Sciences Laboratory