Module Catalog

International AVS Center

Date: November 9, 1995

1. International AVS Center Module Catalog

This catalog summarizes the public-domain AVS modules available from the International AVS Center as of May 1, 1993. The catalog separates the AVS modules into four different sections – data input, filter, mapper, and data output – and alphabetically organizes modules within each section.

A postscript version of this module catalog is available on the International AVS Center ftp site. It is updated continuously and is suitable for printing on any postscript printer. For help on acquiring this information, please read the following Obtaining Modules section.

2. International AVS Center

"The North Carolina Supercomputing Center was chosen [as the International AVS Center] primarily because of its unparalleled capability to gather and distribute modules and act as a user/vendor communication channel via its CONCERT network, which is ideal to meet the information and module distribution needs of the AVS users and vendors," says Paul Esdale of Advanced Visual Systems Inc.

The International AVS Center serves as a catalyst for expanding the AVS user base and for increasing the functionality of AVS by fostering discipline-specific module development and development of new AVS applications. As a world clearinghouse for AVS information, the International AVS Center collects, ports, and distributes user-contributed, public-domain AVS modules and acts as liaison between AVS users and vendors.

In addition to fostering collaborations among AVS users with related interests, the International AVS Center coordinates an international AVS users group, publishes the quarterly AVS Network News magazine, operates an AVS showroom, and serves as a porting facility for distributing AVS applications and modules across heterogeneous platforms.

3. AVS Consortium

The AVS Consortium provides growth, direction, and promotion strategies for AVS to become the de facto standard visualization environment. It also manages the International AVS Center's activities and the allocation of its resources.

In operation since October 1, 1991, the AVS Consortium currently comprises representatives from various AVS vendors, including Advanced Visual Systems Inc., CONVEX Computer Corporation, Digital Equipment Corporation, Hewlett-Packard, IBM, SUN Microsystems, and Wavetracer. These vendors' cooperative effort to provide a common interface for their users' benefit exemplifies true vision and marks a tremendous milestone in visualization. Additional AVS vendors may join the consortium in the near future.

4. Obtaining Modules

AVS modules may be obtained by standard ftp protocol, email, or standard mail. The International AVS Center ftp site is located on the Internet at avs.ncsc.org (128.109.178.23). When connected to the avs server, log in as anonymous and enter your email address as the password. Instructions on navigating through the ftp site's multiple directories are in your local man pages on ftp.

Several introductory files are at the ftp site's top level, including an FAQ, this module catalog, and an AVS_README file that includes detailed information on the International AVS Center ftp site. The top level also includes four directories that correspond to this catalog's data, filters, mappers, and renderers sections.

Each of these four directories has subdirectories for individual modules that include module source code, module documentation, sample networks, and Makefiles for each platform to which the International AVS Center staff port the module (e.g., make.CONVEX, make.DEC, make.IBM, make.HP, make.Sun, and make.Wave).

For users who do not have ftp protocol on their system, an email system is available for obtaining modules. The three Internet email address are avs@ncsc.org for general questions and module submissions; avsemail@ncsc.org for current catalog and README retrieval; and avsorder@ncsc.org for automated module retrieval that includes all files associated with the retrieved module.

Until completion of current construction of porting facilities for modules on tape, tape requests continue to be serviced by email and standard mail. For instructions on obtaining modules on tape, send email to avs@ncsc.org or standard mail to the International AVS Center, Post Office Box 12889, 3021 Cornwallis Road, Research Triangle Park, North Carolina 27709-2889.

5. Submitting Modules

Donating a module to the International AVS Center benefits the entire AVS user community by facilitating AVS use to visualize complex scientific phenomena. A module that is not donated may be duplicated by someone else, wasting that person's time and possibly hindering further development of other module capabilities for all users' benefit. All modules listed in this catalog are free submissions by module authors.

Modules can be submitted by ftp, email, or tape methods that parallel the preceding module retrieval methods. Modules can be placed on the ftp site within the SUBMIT directory, sent by email to avs@ncsc.org, or put on tape and mailed to the International AVS Center.

All submissions must include the entire source code for the module, a Makefile for the appropriate platforms upon which the module was developed, and documentation on the module's usage. A sample AVS network demonstrating module usage also is requested.

Submitting Modules

secno = ount0

DATA INPUT MODULES

International AVS Center

3D_axis

Author :	**Anonymous** Kubota Pacific Computer Inc
Number :	1152
Language :	C
Submitted :	03/12/92
Version :	1.000
Last Updated :	03/12/92
Ported To :	Kubota DEC IBM Sun

This module displays the coordinate axis of the 3D graphic display.

ACE_to_UCD

Author :	**Ian Curington** Advanced Visual Systems, Inc.
Number :	1637
Language :	C
Submitted :	09/24/93
Version :	1.000
Last Updated :	09/24/93
Ported To :	HP Kubota Sun

This is intended as a source code example to show how to allocate and manage a simple UCD output port structure, from within a free-running AVS coroutine. This simulates time varying data on a 2D quadrilateral mesh. It has no parameters or input ports, for simplicity all values are hand-coded. The UCD output structure contains a 2D mesh of quadrilateral cells, with connected shared nodes. The geometric and connectivity info is only built the first time, then only the values change from each time step.

AFS_extract

Author :	**Jeff Wang** MCNC, Environmetal Programs Group
Number :	1732
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Sun

This module allows users to extract data from AIRS AFS (AIRS Facility System) files. The AIRS AFS format are refered to User's Guide For the Urban Airshed Model, EPA-450/4-90-007D. Users may read through the very begining of the c codes provided to get an idea how the data is constructed. This is one of a suite of modules we used to plot the emission data for the state of North Carolina. This group of modules allows displaying points, lines and polygons over a map of the United States. The module suite is stored as a group in a compressed tar file at the International AVS Center, with some sample networks and datasets included. You may find it easier to access the modules this way. The compressed tar file is approximately 1.2MB in size, and can be found via anonymous ftp to avs.ncsc.org, in /sample_data/avs_data/Wang_GIS_mods s.tar.Z.

AMS_extract

Author :	**Jeff Wang** MCNC, Environmetal Programs Group
Number :	1733
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Sun

This module allows users to extract data from AIRS AMS (AIRS Area and Mobile Subsystem) files. The AIRS AMS format are refered to User's Guide For the Urban Airshed Model, EPA-450/4-90-007D. Users may read through the very begining of the C codes provided to get an idea how the data is constructed. This is one of a suite of modules we used to plot the emission data for the state of North Carolina. This group of modules allows displaying points, lines and polygons over a map of the United States. The module suite is stored as a group in a compressed tar file at the International AVS Center, with some sample networks and datasets included. You may find it easier to access the modules this way. The compressed tar file is approximately 1.2MB in size, and can be found via anonymous ftp to avs.ncsc.org, in /sample_data/avs_data/Wang_GIS_mods s.tar.Z.

ANIMATE_DUAL

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1793
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP Kubota Sun

Unlike the AVS module, "animated integer", this animator co-routine keeps up with the range of the values available. It also allows the user to set a delta value for going forward (positive) or backward (negative). Additionally, the user can advance by single steps through data or continuously. When the end of the range is reached, the animator will wrap around the end of the range to the opposite end by delta. A second set of variables allows the synchronous control of two viewing windows. Output from the animator will cease for "Pause Seconds" after each previous output. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

ARC_MAP

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1792
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

The ARCMAP module creates an AVS geometry by reading input from an ARCinfo gen file. The gen file structure is that of ARC segment number, a list of x-y coordinates of the segment polygon, and an 'END' statement. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

Application_Menu

Author :	**Joseph Fall** Essential Planning Systems
Number :	1829
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP Kubota Sun

The ApplMenu module reads a menu description file and creates an AVS menu that conforms to this description. It also controls this menu and issues the appropriate CLI commands when various menu choices are made by a user.

BGF_Protein_Disp

Author :	**Advanced Visual Systems, Inc.**
Number :	1679
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

This module is both a reader and a mapper. It reads ASCII Biograph protein files, and generates a geometric object for the geometry viewer. It has advanced modes, such as creating a spline ribbon representation of the backbone chain.

BP_Coroutine

Author :	**Ian Curington** Advanced Visual Systems
Number :	1325
Language :	C
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	DEC Sun HP

This module is a source example of how to write a free-running asynchronous coroutine that provides data to AVS on the output port when data is ready. In its present form, it free-runs, that is continually sends data to AVS as fast as it can be processed. The project specific version of this had external calls to obtain data from a data acquisition instrument.

ConstImage

Author :	**Stefano Brozzi, Andrea Menozzi, Giulio E. Destri** Universita' degli Studi di Parma, 43100 Parma, Italy
Number :	1873
Language :	C
Submitted :	09/06/95
Version :	1.000
Last Updated :	09/06/95
Ported To :	DEC_ultrix DEC_alpha

This module was built due to the need to be able to generate a constant image to input as bias or as initial state in the module aCNN. The module accept as input a field, tipically the image, and assumig the dimensions of this field gives as output a field 2-D scalar byte of a constant value.

DMOL_to_field

Author :	**Mark Reed** North Carolina Supercomputing Center
Number :	1857
Language :	C
Submitted :	10/17/94
Version :	1.000
Last Updated :	10/17/94
Ported To :	DEC_alpha HP IBM Sun

This module reads the graphical output files generated by DMol from Biosym (see REFERENCES included in documentation), and generates an AVS field. This field can then be manipulated to display an isosurface of the electronic charge or use slicers to view the data.

Dials_Contro

Author :	**Anonymous** Stardent Computer
Number :	1027
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Sun Kubota IBM DEC HP

This simple module provides rotation, scale, and translation to any object in the geometry viewer scene, with separate controls in the form of dial parameters. The module connects in PARALLEL with other geometric objects to the geometry viewer. It operates on the named object in the scene.

Dials_Control

Author :	**James Peters (Convex), Ian Curington (AVS Inc.), and Stephen Pass (???)**
Number :	1745
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This simple module provides rotation, scale, and translation to any object in the geometry viewer scene, with separate controls in the form of dial parameters. The module connects in PARALLEL with other geometric inputs to the geometry viewer. It operates on the named object in the scene.

Dipole

Author :	**John Sheehan** Advanced Visual Systems
Number :	1701
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP IBM Kubota Sun

This is an example module to show how one can go from cylindrical or spherical to cartesian coordinates. This has been written for specific datasets, but it should serve the purpose. Anyways, the trick is to calculate the points array properly in order to feed field2mesh, in most cases. See also - Read.

Edit_File

Author :	**Jeff Wang**
	MCNC, Ennvironmetal Programs Group
Number :	1720
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP Kubota Sun

This module allows users to run a vi editor on a remote machine inside AVS. The user selects the file to edit and inputs the IP number for the local machine.

Edit_String

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1328
Language :	C
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	

The Edit String module puts up a LONG dialog box, allows typing of any string, then with either "OK" or "CANCEL" controls, will output this string to the output port of the module. The next time the "Edit" button is pushed, the dialog box reappears, with the old string, which can be edited. This module is for long expressions, lists of ints, etc, where the normal string type_in widget is too small. The dialog box comes up with nicer colours than the default grey.

Extrusion

Author :	**Feizal Mirza**
	University of Texas CHPC
Number :	1313
Language :	C
Submitted :	08/30/92
Version :	1.000
Last Updated :	08/30/92
Ported To :	DEC Sun HP

This module reads an ASCII file consisting of lists of x and y points and constructs a 3-D geometry object using the points in the file as vertices. It uses the value specified by the user in the depth dial for the depth of the object.

FIND_DATA_AND_MA

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1796
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Sun

FIND DATA AND MAP – locates data and map based on user choice from a DATA LIST. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

FIND_MAP

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1797
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

FIND MAP – locates a map based on user choice from a MAP LIST. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

FITS_read

Author :	**D. LE CORRE** CONVEX S.A
Number :	1306
Language :	CandFORTRAN
Submitted :	08/30/92
Version :	2.000
Last Updated :	11/04/92
Ported To :	IBM

This DATA INPUT module reads a FITS (Flexible Image Transport System) file into a field, and optionally to a text window. For primary data or image extensions, the number of dimensions is derived from the NAXIS keyword. For Ascii and binary tables, the number of dimensions may be 1 or 2, depending on a parameter. Random Groups do not seem to be supported (I did'nt see any example of them). Version 2 has code to port to the IBM platform.

GET_MODEL_MAP

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1799
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Sun

GET MODEL MAP – generates map and crop information based on selected map. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

Generate_label

Author :	**Penny Rheingans** US EPA Scientific Visualization Center
Number :	1300
Language :	C
Submitted :	08/26/92
Version :	1.000
Last Updated :	08/26/92
Ported To :	DEC Sun HP

Generate label creates a geom object describing a label object. This label is a title, that is it has a fixed position on the screen, regardless of viewpoint transformations. This module provides a portion of the functionality of labels in the geometry viewer, but adds the ability to generate labels automatically. This is useful for putting up frame or time-step numbers.

Get_Terrain_Data

Author :	**Jeff Wang** MCNC, Environmetal Programs Group
Number :	1780
Language :	C
Submitted :	02/09/94
Version :	1.000
Last Updated :	02/09/94
Ported To :	HP Sun

This module allows users to read in sorted ASCII terrain data and generate an 2-D irregular field. Together with the "Draw Map" module on the ftp site, users can create a 2-D terrain plot over the political boundary map. Sample data for this module can be found via anonymous ftp to avs.ncsc.org, in /sample_data/avs_data/Get_Terrain_D Data/dma.dat.

Graph_slider

Author :	**Hideo Miyachi** Kubota Corporation
Number :	1614
Language :	C
Submitted :	05/01/93
Version :	1.000
Last Updated :	05/01/93
Ported To :	Kubota HP

Graph slider can shift the range in x axis in graph viewer. By default, you can specify the max and min value respectively. You can set the range and min value in this module so that if you change only min value the graph slides. This module has no input and output ports. That means this module accesses CLI. This module is very simple and short code. You can use as a refernce for CLI access.

Image_Sequen

Author :	**Brian Kaplan** Center for Innovative Computer Applications
Number :	1134
Language :	C
Submitted :	02/23/92
Version :	1.000
Last Updated :	02/23/92
Ported To :	Kubota HP

The Image Sequencer module is used to read a series of images that have been previously saved to disk and output them as AVS image data. This can be very useful for displaying a series of images, recording them to video, or compositing or filtering a series of images. The images can be in any of the formats supported by the Image Sequencer. Image formats currently supported are AVS image format, MIFF, RLE, and RLA/B.

LBF_extract

Author :	**Jeff Wang** MCNC, Environmetal Programs Group
Number :	1734
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Sun

This module allows users to extract data from link base mobile emission files. The link base emission format are refered to User's Guide For the Urban Airshed Model, EPA-450/4-90-007D. Users may read through the very begining of the C codes provided to get an idea how the data is constructed. This is one of a suite of modules we used to plot the emission data for the state of North Carolina. This group of modules allows displaying points, lines and polygons over a map of the United States. The module suite is stored as a group in a compressed tar file at the International AVS Center, with some sample networks and datasets included. You may find it easier to access the modules this way. The compressed tar file is approximately 1.2MB in size, and can be found via anonymous ftp to avs.ncsc.org, in /sample_data/avs_data/Wang_GIS_mods s.tar.Z.

Label_Axis

Author :	**Roy Wirthlin** Advanced Visual Systems Inc
Number :	1289
Language :	C
Submitted :	07/03/92
Version :	1.000
Last Updated :	07/03/92
Ported To :	Sun HP

The Label Axis module generates axes with tic marks and labels in 3D space. The axis can be generated with tic marks and tic labels along each axis. This module very simular to the supported module generate axis but porvides more flexibility and funcionality to the user.

Life

Author :	**Steve Thorpe**
	International AVS Center
Number :	1277
Language :	C
Submitted :	05/27/92
Version :	1.000
Last Updated :	05/27/92
Ported To :	DEC IBM Sun Kubota HP

Life is an implementation of the "Life" problem. This is an algorithm where "neighbors", or cells, propagate or die, based on the proximity of other living neighbors. In this case, a living neighbor is represented by a simulation of a processor in the massively parrallel Wavetracer being "turned on". The thoughts behind the algorithm are - If there are too many neighbors, a cell will die due to overcrowding. If there are too few neighbors, a cell will die due to lack of resources. It is meant to be used with the Life_WT AVS module, which is a parrallel implementation of Life. As the number of processors increases, the performance of Life_WT greatly supercedes that of Life, due to the benefits of the Wavetracer's multiple processors operating in parrallel. The output from this coroutine module is an AVS image, which can be sent to the display_image module for viewing as the computation progresses. The user sees a visualization of the processors that are "alive" at that moment in the computation.

Life_WT

Author :	**Steve Thorpe**
	International AVS Center
Number :	1280
Language :	multiC
Submitted :	06/02/92
Version :	1.000
Last Updated :	06/02/92
Ported To :	IBM Wavetracer

Life_WT is an implementation of the "Life" problem. This is an algorithm where "neighbors", or cells, propagate or die, based on the proximity of other living neighbors. In this case, a living neighbor is represented by a processor in the massively parrallel Wavetracer being "turned on". The thoughts behind the algorithm are If there are too many neighbors, a cell will die due to overcrowding. If there are too few neighbors, a cell will die due to lack of resources. The user is able to control the number of processors in the X direction, the number of processors in the Y direction, the number of generations to compute, and the number of generations to compute between outputs of an image. A oneshot parameter, "Go for it.." is supplied to begin the execution of the module. Life_WT is written in multiC, a fully compatible extension of the ANSI C programming language. It was developed using a Data Transport Computer from Wavetracer, Inc.

Load_Simulation

Author :	**Prashant Gupta and Steven Smith**
	North Carolina State University
Number :	1729
Language :	C
Submitted :	12/14/93
Version :	1.000
Last Updated :	12/14/93
Ported To :	Sun

The Loadsim module will create a ball and stick geometry representation of molecular dynamics or Monte Carlo data. The ability to interactively select positions in the data file as well as individual chains is advantageous for visualizing chain motions. Although written for visualizing simple chain-like molecules it can be extended to other systems.

MOSS_LINES

Author :	**Scott Goodyear**
Number :	1843
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP IBM Sun

MOSS_Polylines - Reader for MOSS format polyline vector lists. The Arc/Info MOSS export files are an ASCII format file readable by the US Department of the Interior's MOSS public domain GIS. The file contains points, lines, or closed polygon loops, and a 30-character attribute field, called the "subject" value. The Arc/Info coverages are converted into the MOSS files using the "ARCMOSS" utility in the data conversion utility subsystem. This module has a file browser, for selecting the ASCII input MOSS file, which is scanned for closed polygon areas. The polylines are converted into AVS geometry polylines for use by the AVS geometry viewer. The polyline does not have a color defined in this module, such that it can be interactively altered in the viewer, and saved with a network file.

MOSS_Markers

Author :	**Scott Goodyear**
Number :	1841
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP Kubota IBM Sun

MOSS_Markers - Reader for MOSS format markers. The Arc/Info MOSS export files are an ASCII format file readable by the US Department of the Interior's MOSS public domain GIS. The file contains points, lines, or closed polygon loops, and a 30-character attribute field, called the "subject" value. The Arc/Info coverages are converted into the MOSS files using the "ARCMOSS" utility in the data conversion utility subsystem. This module has a file browser, for selecting the ASCII input MOSS file, which is scanned for markers. The markers are converted into AVS geometry for use by the AVS geometry viewer. Each marker is shown as a sphere, with radius controlled by the dial. The marker does not have a color defined in this module, such that it can be interactively altered in the viewer, and saved with a network file.

MOSS_POLYGONS

Author :	**Scott Goodyear**
Number :	1842
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP IBM Sun

MOSS_Polygons - Reader for MOSS format polygon coverages. The Arc/Info MOSS export files are an ASCII format file readable by the US Department of the Interior's MOSS public domain GIS. The file contains points, lines, or closed polygon loops, and a 30-character attribute field, called the "subject" value. The Arc/Info coverages are converted into the MOSS files using the "ARCMOSS" utility in the data conversion utility subsystem. This module has a file browser, for selecting the ASCII input MOSS file, which is scanned for closed polygon areas. The polygons are converted into AVS geometry for use by the AVS geometry viewer. The polygon does not have a color defined in this module, such that it can be interactively altered in the viewer, and saved with a network file.

Menger_sponge

Author :	**Larry Gelberg** Advanced Visual Systems
Number :	1710
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP IBM Kubota Sun

Menger sponge produces a Menger Sponge (Sierpinski Sponge) as a byte volume. A Menger Sponge (sometimes known as a Sierpinski Sponge) is a recursively defined volume. It is the 3D analog to a Sierpinski Carpet which is can be poorly defined as a recursive 3x3 grid where the center square of each level of the recursion is missing. A more formal definition can be found in Benoit Mandelbrot's book "The Fractal Geometry of Nature" on page 144. You can see a Sierpinski Carpet if you orthoslice the sponge and look at slice 0 (any direction). CURDLE (at the IAC) is another example of a module which produces recursively defined "fractal" data sets.

Mouse_Trap

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1670
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP

The "Mouse Trap" module accesses the X-Window of the Geometry Viewer display, and changes the mouse button definitions. It is intended as an AVS application developers example, to create custom interface controls. It intercepts all mouse events, and then selectively sends commands on to AVS for action. At present, it ignores all except the right mouse button, which behaves as a "pick", like the normal AVS left mouse button, while ignoring rotation and translation requests. The implementation is to make an invisible input only window, exactly covering the geometry viewer window. Events from this invisible window are detected, masking the geometry viewer window native events. Commands are then sent via the CLI mechanism. As a coroutine, this module can also take events from user interface parameters, although none are present in this version.

Mouse_Trap_IV

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1828
Language :	C
Submitted :	07/15/94
Version :	1.000
Last Updated :	07/15/94
Ported To :	DEC HP IBM Kubota Sun

The "Mouse Trap IV" module accesses the X-Window of the Image Viewer display, and changes the mouse button definitions. It intercepts all mouse events, and then selectively sends commands on to AVS for action. This version ignores all normal image viewer mouse controls, and causes special behavior instead. The left button will "zoom in" the image, while the right button will "zoom out" again, each by a preset amount. It is intended as an AVS application developers example, to create custom interface controls. This module also shows how to change the mouse cursor shape while over viewer window. This is intended as a source example for further development work.

QUAL_field

Author :	**Marc Curry** North Carolina Supercomputing Center
Number :	1321
Language :	C
Submitted :	08/30/92
Version :	1.000
Last Updated :	08/30/92
Ported To :	DEC Sun HP

The QUAL field module is used to selectively grab data elements from a large output plot file and map them to the geometry of an example river. The geometry can be changed (with some effort) to match that of any river. The codes that generate the output plot files used by this module are QUAL2E and QUAL2E-UNCAS which were developed by Linfield Brown (Dept. of Civil Eng, Tufts Univ.) and Thomas Barnwell (EPA, Athens GA.).

RADM_Credits

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1111
Language :	C
Submitted :	12/06/91
Version :	1.000
Last Updated :	12/06/91
Ported To :	Kubota DEC IBM Sun HP

This module outputs a series of “title” labels, creating a credit page for an AVS application. This module provides an example for creating a single geometry object with multiple labels within it.

READ_ANY_IMAGE

Author :	**Terry Myerson, Steve Thorpe, and SDSC** International AVS Center (NCSC) and San Diego Supercomputer Center
Number :	1211
Language :	C
Submitted :	05/21/92
Version :	3.000
Last Updated :	07/05/93
Ported To :	HP IBM DEC Sun

This module reads an image into an AVS Network in a variety of formats. All formats which the San Diego Supercomputing Center's Image Tools support, this module supports. With this module you can read in a postscript file, and edit it with the standard tools of AVS . Any of the following image file formats can be read in by this module eps Encapsulated PostScript file, gif Compuserve Graphics image file, hdf Hierarchical Data File, icon Sun Icon and Cursor file, iff Sun TAAC Image File Format, mpnt Apple Macintosh MacPaint file, pbm Portable Bit Map file, pcx ZSoft IBM PC Paintbrush file, pgm Portable Gray Map file, pic PIXAR picture file, pict Apple Macintosh QuickDraw/PICT file, pix Alias image file, pnm Portable aNy Map file, ppm Portable Pixel Map file, ps PostScript file, ras Sun Rasterfile, rgb SGI RGB image file, rla Wavefront raster image file, rle Utah Run length encoded image file, rpbm Raw Portable Bit Map file, rpgm Raw Portable Gray Map file, rpnm Raw Portable aNy Map file, rppm Raw Portable Pixel Map file, synu Synu image file, tiff Tagged image file, x Stardent AVS X image file, xbm X11 bitmap file, xwd X Window System window dump image file, and any others supported by SDSC's Image Tools. Version 3.0 is much easier to compile than before, as it doesn't link with the Image Tools - it calls the imconv utility directly using a system call.

READ_TIMESTAMP

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1813
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Sun

READ TIMESTAMP – read the input field and timestamp file and output a subset field and a subset timestamp string (suitable as recursive inputs to this module once the timestamp string is written into a file). The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

READ_TIN

Author :	**Scott Goodyear**
Number :	1845
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP Sun

NAME READ_TIN - Reader for Arc/Info Triangulated Region Mesh The READ_TIN module accesses external ASCII files defining a triangulated irregular network (TIN) structure from Arc/Info. This is generally used for a surface representation derived from irregulary space sample points and breakline features. The TIN data set includes topological relationships between points and their proximal triangles. Each sample point has an x,y coordinate and a surface, or z-value. These points are connected by edges to form a set of nonoverlapping triangles userd to represent the surface. TIN files are also called "irregular triangular mesh" or "irregular triangular surface model" in some contexts. "TIN" is the name of the ESRI Arc/Info application module that models, creates, displays, and exports TIN files. Within the TIN application the UNGENERATETIN command with the NET option should generate files compatable with this reader module. An optional image input port will apply a full colour texture map onto the TIN geometry, such that the geometry viewer can show the texture overlay on the terrain. The second file browser (WorldImageFile) allows a small coordinate range correction file to specify the mapping between the optional input raster texture image and the TIN coordinate system. This module requires AVS 5.01, with texture per object support.

READ_WIND_TIMEST

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1814
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Sun

READ WINDTIMESTAMP – read the input wind field and timestamp file and output a subset field and a subset timestamp string (suitable as recursive inputs to this module once the timestamp string is written into a file). The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

RdUcdAnim

Author :	**D. LE CORRE** CONVEX S.A
Number :	1316
Language :	C
Submitted :	08/30/92
Version :	1.200
Last Updated :	12/30/92
Ported To :	Sun HP

This DATA input module animates a sequence of data applied to a UCD structure. The base format is exactly the same as the standard UCD, except that you can specify a 6-th value on the first line (if you use ASCII format) , giving the maximum number of time steps to consider. By default, allocation is provided for at most 256 time steps.

ReadLat

Author :	**Dominique Le Corre** Tethys S.A.
Number :	1596
Language :	C
Submitted :	03/04/93
Version :	1.000
Last Updated :	03/04/93
Ported To :	DEC IBM Kubota HP

This module reads an Explorer Lattice file and generates a field. It supports both ASCII and binary files. The file name parameter only shows file names with extensions .lat and .cmap, but you can enter any file name. Components are labeled Channel 0, Channel 1, etc. and units also.

ReadRadiance

Author :	**Stephen H. Price for Advanced Visual Systems, Inc.**
Number :	1727
Language :	C
Submitted :	12/14/93
Version :	1.000
Last Updated :	12/14/93
Ported To :	Sun

This data input module reads a *.pic file created with Radiance 2.1 (or greater) radiosity package. It outputs a Field 2D 4-vector b yte field for viewing with the image viewer module. An sample image file is included.

ReadSeqAnim

Author :	**Ion Barosan** Eindhoven University of Technology, The Netherlands
Number :	1584
Language :	C
Submitted :	02/11/93
Version :	2.000
Last Updated :	03/04/93
Ported To :	IBM Kubota Sun

The read_seq input modules sends a sequence of files' names, to a string parameter ports. Its purpose is to generate a sequence of files names, a sequence which will be used as input for different data input modules. It has four parameters, called “File name“, ”Animation On/Off“, ”Current step“, and ”Nr.frames“. In a typical network with read_seq, before you can connect the character string to the receiving module,you must make that receiving module's parameter port visible.

Read_2

Author :	**John Sheehan**
	Advanced Visual Systems
Number :	1702
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP IBM Kubota Sun

This is an example module to show how one can go from cylindrical or spherical to cartesian coordinates. This has been written for specific datasets, but it should serve the purpose. Anyways, the trick is to calculate the points array properly in order to feed field2mesh, in most cases. See also - Dipole.

Read_CFD

Author :	**KGT (Kubota Graphics Technologies) Inc.**
Number :	1882
Language :	CandFORTRAN
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95

You can use this module for example in the field of CFD. The "Read CFD" module reads a sample data. The format of a sample data(CFD) is discussed below in the "Read_CFD.txt" file. It is the most populer format of CFD.

Read_DXF

Author :	**John Sheehan**
	Advanced Visual Systems
Number :	1281
Language :	C
Submitted :	06/02/92
Version :	1.000
Last Updated :	06/02/92
Ported To :	DEC Kubota HP

Read DXF is a utility module for converting 2D and 3D DXF files into an AVS Geometry Data Type. If level information is available in the DXF file, Read DXF will use this to create separate objects. DXF is the file type for autoCAD output. Supported Entities LINE POLYLINE 3DFACE BLOCK LEVEL Read DXF can be easily modified to handle more primitives, ie. points, arcs, and circles. All of the DXF header information is overlooked. If Read DXF encounters a primitive it doesn't understand it dumps out a geometry of what it had generated to that point, plus a message with the code number , the command to interpret, and the current layer name.

Read_EJ65

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1680
Language :	FORTRAN
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP

The EJ65 file format interface is for 3D body fitted coordinate flowfield and mesh data. This module is a Reader for the EJ65 geometry and field data, in FORTRAN. It accesses the files via interactive browsers, and reads the binary files directly into the AVS session. The two browsers are for the mesh and flow field files. They are combined on one output port. The module is automatic, and follow the input array dimensions, but must be compiled with an interal array size to match the largest expected field. Two types of input are supported, native floating point format of the host machine (double precision), and if the host is bigendian, then native CONVEX non-IEEE binary input is supported, using the optional mode. By default, the flow solution data is moved 1/2 a cell to the node positions. If requested, it is left at the cell centers, and nodal values are calculated by nearest cell interpolation. The interpolation mode is untested. The Freeze button us used to set new flow or mesh files, or other parameters on the module, before attempting the read. Both files are accessed, regardless of the parameter change. The EJ65 file format is in use at a number of sites, including British Aerospace, PLC, UK

Read_Hologram

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1322
Language :	FORTRAN
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	HP

A well-commented Reader for 2D field data, in FORTRAN, that combines ASCII file inport in fixed FORTRAN FORMAT, and does numeric computation on the data before sending it to the output port. It allows multiplication of a coefficient on one particular row, defined by a channel dial. This is intended as a working example of a FORTRAN read module, to be modified by others.

Read_SDRC

Author :	**Steve Larkin, Ian Curington, and Geoff Cook** University of Manchester and AVS Inc
Number :	1590
Language :	C
Submitted :	02/21/93
Version :	1.000
Last Updated :	02/21/93
Ported To :	DEC IBM Kubota HP

Read SDRC reads a file which is in the SDRC I-DEAS universal format converts it into an AVS UCD format. At present only two types of node data record are implemented, static and transient. A static record has one set of values at each node. A transient record has a set of values at each node for different times. Please see the International AVS Center's sample dataset subdirectory for two datafiles you might want to try with this module - crankshaft.unv and trans-therm.unv. PARAMETERS

Read_SEG_Y

Author :	**John Tee** AVS Inc. Consultant
Number :	1753
Language :	CandFORTRAN
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota

READ SEG Y is an data input module which reads seismic data stored in the SEG Y format. SEG "Y" is a format devised by the Society of Exploration Geophysicists (SEG), for transfer of demultiplexed seismic data on 9-track tape. SEG Y was published in 1974, and superceded the SEG "Ex" format. SEG Y defines, for each tape reel, a 3200 byte EBCDIC text header, followed by a 400 byte binary header. The data follows as a variable number of traces. Each trace has a 240 byte header, followed by a variable number of samples, each represented as a 4-byte real value.

Read_Unversal_FI

Author :	**Hiro Kawanishi** Digital Equipment Corporation JAPAN
Number :	1562
Language :	C
Submitted :	12/30/92
Version :	1.000
Last Updated :	12/30/92
Ported To :	IBM Kubota

Read Unversal File is a utility module for conversion of I-DEAS Unversal Files into AVS UCD data. This module only extracts Stress data and node-element information.

Sin_Wav

Author :	**Steve Thorpe** International AVS Center
Number :	1600
Language :	C
Submitted :	11/30/92
Version :	1.000
Last Updated :	11/30/92
Ported To :	DEC Kubota HP IBM

This little module was written while playing with the graph viewer. A simple module to spit out a 1-D field was desired, and this did the trick. It computes a sin wave over the range 1..npoints degrees, and spits out a "field 1-D 1-space 1-vector uniform float" suitable for display with the graph viewer module. This may be a useful starter module if you want to learn how to get your 1-D data into AVS.

Sonify

Author :	**Brian Kaplan** Center for Innovative Computer Applications
Number :	1635
Language :	C
Submitted :	07/05/93
Version :	1.000
Last Updated :	07/05/93

The Sonify module will synthesize sounds and can be controlled by other modules via command string input. It is most useful for data sonification where a data-producing module will send commands to Sonify based on the data represented.

Splash

Author :	**Marc Curry** North Carolina Supercomputing Center
Number :	1585
Language :	C
Submitted :	02/11/93
Version :	1.000
Last Updated :	02/11/93
Ported To :	DEC IBM Kubota HP

This module is a simple C subroutine that will output a variety of surfaces/colors depending upon what 3-D mathematical function you choose it to have. The basic framework for the existing module example simply computes the distance from the center of a 100x100 grid (R = sqrt(r1*r1 + r2*r2)) and multiplies it by a sinusoidal factor (cos(scale * R)) where scale is a scaling parameter, used to give it the waviness (z-dimension) we want. You can make it very interesting by attaching the "animated float" module to the widget input port instead of the float dial. The example network provided does this. This is a great example to learn how to manipulate AVS fields.

Tick_Marks

Author :	**Jeffrey A Thingvold** National Center for Supercomputing Applications
Number :	1338
Language :	C
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	DEC Sun HP

The tick mark module creates sheets with major and minor line grids along the three primary axes. It makes polytriangle objects, not lines, so they can have variable thickness. It also has label annotation on the 3D object.

Vector_Label

Author :	**Charles J Williams III** Naval Research Lab
Number :	1563
Language :	C
Submitted :	12/30/92
Version :	1.000
Last Updated :	12/30/92
Ported To :	IBM Kubota HP

This module produces labels that are geometries. They can be scaled, rotated, and translated. Other geometric operators work as well, such as color, perspective, and material properties. The source for the vector information for the fonts is derived from the Hershey Fonts, stored in comp.sources.unix, volume 4, on your nearest archive site. The Hershey Fonts were originally created by Dr. A. V. Hershey while working at the U.S. National Bureau of Standards. The format of the font data in the Internet distribution was originally created by James Hurt at Cognition, Inc.

Vernier_Dial

Author :	**Ian Curington** Advanced Visual Systems
Number :	1331
Language :	C
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	

This module is a C coroutine that presents a special X-window with a large dial. The dial is similar to the standard AVS dial widget, but has higher resolution and a secondary needle that moves much slower than the primary needle. The dial can work in accumulator mode (around and around) or with fixed low and high end stops. In immediate mode it sends an output float to the output port continuously while the needle is moving, otherwise only when the mouse button lifts up.

WriteSeqAnim

Author :	**Ion Barosan** Eindhoven University of Technology, The Netherlands
Number :	1598
Language :	C
Submitted :	03/04/93
Version :	1.000
Last Updated :	03/04/93
Ported To :	DEC IBM Kubota

The WriteSeqAnim module sends a sequence of file names to a string parameter port. Its purpose is to generate a sequence of file names; the sequence will be used as input for different DATA OUTPUT modules. Before you can connect the character string to the receiving module, you must make that receiving module's parameter port visible. To make a parameter port visible, call up the module's Editor Window panel by pressing the middle or right mouse button on the module icon dimple. Next, look under the "Parameters" list to find the parameter you want to plug into. Position the mouse cursor over that parameter's button and press any mouse button. When the Parameter's Editor Window appears, click any mouse button over its "Port Visible" switch. A blue-green (teal) parameter port should appear on the module icon. Connect this parameter port to the character string module icon in the usual way one connects modules.

Zoomer

Author :	**Jeff Wang** MCNC, Environmetal Programs Group
Number :	1743
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This module allows users to set the coordinate of the "zoom box" and UTM zone of the view area. This is one of a suite of modules we used to plot the emission data for the state of North Carolina. This group of modules allows displaying points, lines and polygons over a map of the United States. The module suite is stored as a group in a compressed tar file at the International AVS Center, with some sample networks and datasets included. You may find it easier to access the modules this way. The compressed tar file is approximately 1.2MB in size, and can be found via anonymous ftp to avs.ncsc.org, in /sample_data/avs_data/Wang_GIS_mods s.tar.Z.

animate_file

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1135
Language :	C
Submitted :	02/24/92
Version :	2.000
Last Updated :	11/04/92
Ported To :	Kubota DEC IBM Sun HP

anim_fname is used to output a series of filenames for input into a reader module. The module inputs an integer and a filename base, and output a filename in the form "$base.%3d". This module is very useful for a series of files containing a time series of data. Bug fixes and extensions for version 2.0 added by Wes Bethel, LBL.

animate_floa

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1009
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

The animate float module automatically modifies teh floating point parameters. It is used to create simple animations or to drive user simulation code. This module is similar to standard animate float module except for the following modifications (1) The maximum number of steps has been modified to be minimum distance detector 10000, rather than the standard 100 (2) The "mode" choice has been removed. This module operates in a manner analogous to "animate float"'s "one time" mode. That is, the module executes the number of times indicated by steps. (3) A restart switch has been added. This is useful for restarting the module from step 0 (without having to kill then reinistate the module)..

animate_inte

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1010
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

The animate integer module automatically modifies the integer parameters. It is used to create simple animations or to drive user simulation code. This module is similar to the standard animate integer module except for the following modifications (1) The maximum number of steps has been modified to be 10000, rather than the standard 100. (2) The "mode" choice has been removed. This module operates in a manner analgous to "animated float"'s "one time" mode. That is, the module executes the number of times indicated by "steps". (3) A restart switch has been added. This is useful for restarting the module from step 0 (without having to kill then reinstantiate the module).

animated_boolean

Author :	**William Hunter** GEC Ferranti DSL, Edinborough, Scotland
Number :	1183
Language :	C
Submitted :	05/04/92
Version :	1.000
Last Updated :	05/04/92
Ported To :	DEC IBM Sun Kubota HP

The animated boolean module automatically modifies toggle parameters. This can be used to create simple animations or to drive user simulation code. You plug animated boolean into another module's toggle parameter port, type in a number of steps (default 10). When you turn off sleep, it starts the animation. animated boolean can be connected to multiple modules.

animated_str

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1011
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

The animated string module automatically sends string parameters. Unlike the "animated integer" and "animated float" modules, which compute their respective output value, the animated string module requires the user to supply a file containing the strings which are sent to downstream modules.

animated_track

Author :	**Ian Curington** Advanced Visual Systems
Number :	1329
Language :	C
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	DEC Sun HP

Animated Track uses the "track" widget to control transformations by producing a 4x4 transformation matrix, or using direct geometry object transformation control. The track control is in "immediate" mode, so that transformations are send dynamically in a stream to the rest of the network. The module has both Field and Geom output ports, so that either one or both may be used. The module is a coroutine, so that animation sequences could be added by inserting matrix modification code to the inner loop.

application_mana

Author :	**John Stephen** Tessella Support Services
Number :	1638
Language :	C
Submitted :	09/24/93
Version :	1.000
Last Updated :	09/24/93
Ported To :	HP Sun

The user selects a file which is then played as an AVS CLI script. The effect is identical to the AVS CLI command "script -play myfile.scr".

awais

Author :	**Steve Thorpe** International AVS Center
Number :	1208
Language :	C
Submitted :	05/15/92
Version :	1.000
Last Updated :	05/15/92
Ported To :	DEC Sun IBM HP

There is now a WAIS (Wide Area Information Servers) server running at the International AVS Center. WAIS allows a user to ask a question to a server, which provides a ranked list of documents that may help answer that question. The user can then peruse through the documents that seem useful. It provides a convienient interface to large amounts of data. AWAIS is a WAIS client, of which there are several freely available. For example, I might be interested in JPEG files, so I might ask the server the question "Can I read or write JPEG files using AVS?" A typical response from a WAIS client that you could type your question into, would be a document list ranked from "most hits" on down. Each list item shows that document's rank, its score (1000 is for the "most hits"), its name (this may show only partially), and the number of lines in that document. I could then select any of these documents for viewing.

cabinet_maker

Author :	**Larry Gelberg** Advanced Visual Systems
Number :	1631
Language :	C
Submitted :	07/05/93
Version :	1.000
Last Updated :	07/05/93
Ported To :	HP IBM Kubota DEC Sun

The cabinet_maker module is one of a suite of four AVS modules which can be used to design simple woodworking projects. For more information on the purpose of the project, please read the file called "Poster_Session" and the man pages for each module. The four modules are "cabinet maker" - generates structures made of boxes, "taper" - taperer - generates structures made of tapered boxes, "extruder" - creates capped, extruded tubes from lists of X/Y pairs, and "grid_gen" - creates measured grids so that accurate models can be constructed. Due to the current structure of the IAC's directories, this suite has been split into 4 separate directories on the ftp site. However, a compressed tar archive of all 4 mods is included with each of the directories as wood_mod_suite.tar.Z. Also note that there are several images contributed with this module suite. These have been stored on the ftp site in the directory at avs.ncsc.org called sample_data/avs_data/woodworking_ima ages.

ccyl_pipe

Author :	**Ken Flurchick** North Carolina Supercomputing Center
Number :	1581
Language :	C
Submitted :	02/11/93
Version :	1.000
Last Updated :	02/11/93
Ported To :	DEC IBM Kubota HP

This module defines the cylindrical geometry coordinates which can used by "volume bounds" (a supported module) to draw a hull. This module also passes the geometry field to "newfilla" (available via anonymous ftp) which in turn fills the cylindrical geometry with data.

choice_2

Author :	**Jonathan Cox** Department of Computer Science, University of Manchester
Number :	1711
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP

The choice module allows the selection and output of choice parameter values. The output can be linked to choice parameters of other modules. This allows a single choice widget to control more than one choice parameters. It can also be used to restrict the range of choices available. The available choices are set by the choice_str parameter. This should be set to the list of the choices delimited by colons.

choice_alias

Author :	**John Stephen** Tessella Support Services
Number :	1642
Language :	C
Submitted :	09/24/93
Version :	1.000
Last Updated :	09/24/93
Ported To :	HP Kubota Sun

This module navigates a menu structure which can be designed and configured by the user. Each menu level is described by a single ASCII text file. A menu level contains several buttons with user-configurable names. Each button results in one of three basic kinds of action in the module - (1) load a new menu level (2) pass a command to downstream modules (3) go to a file to find a series of commands to be passed to down stream modules. Option 3 allows multiple commands to be played in background as the result of a single button press. The file containing the commands to be played in background is the second type of file associated with this system. Thus there are two types of file - (1) the MENU FILE to define a menu level (2) the AUTOMATIC FILE which contains a series of commands to be executed

choice_selec

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1744
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This module takes a string input port, reconfigures a choice list parameter based on the strings and separators in the input port, then outputs the sub-string on a string output port, along with a zero-based integer from the index down the choice list.

color_cube

Author :	**Larry Gelberg**
	Stardent Computer
Number :	1018
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

COLOR CUBE generates a color cube in either RGB (red-green-blue) or HSV (hue-saturation-value) space. The RESOLUTION knob controls the number of voxels in each direction (setting to 16 creates a 16x16x16 or 4096 voxel data set). The ALPHA knob sets a constant alpha for all the voxels - values of 16 are pretty transparent - anything above about 32 becomes pretty opaque.

color_palette

Author :	**John Tee**
	AVS Inc. Consultant
Number :	1643
Language :	C
Submitted :	09/24/93
Version :	1.000
Last Updated :	09/24/93
Ported To :	HP IBM Sun

COLOR PALETTE is used to create a colormap suitable for the display of "label" images, ie scalar fields containing pixel values in the range 1 to n where n is the number of distinguishable categories of pixels. Such images arise from multispectral classification operations. COLOR PALETTE creates a colormap in which the hue values are equally spaced, and an amount of "jitter" may be introduced to the saturation and value in order to improve the differentiation of similar hues. The saturation alternates between 1. and 1. - satn_jitter. The value is successively 1., 1. - val_jitter/2, 1. - val_jitter.

component_menu

Author :	**John Stephen** Tessella Support Services
Number :	1644
Language :	C
Submitted :	09/24/93
Version :	1.000
Last Updated :	09/24/93
Ported To :	HP IBM Sun

This module controls selection of part id's based on integer part lists, in and out. The selection is by browser, and can be added, inverted, or cleared. This can be used with a UCD application to select various material groupings for display or processing.

compress_file

Author :	**Lee J. Silverman** Brown University
Number :	1588
Language :	C
Submitted :	02/21/93
Version :	1.000
Last Updated :	02/21/93
Ported To :	DEC IBM HP

The compress/uncompress module pair is designed to allow datasets or images or other files to be stored and kept in compressed format, but still be read into an AVS network.

coord2geom

Author :	**Mark Reed** North Carolina Supercomputing Center
Number :	1856
Language :	C
Submitted :	10/17/94
Version :	1.000
Last Updated :	10/17/94
Ported To :	DEC_alpha HP Sun

This module reads molecular geometry information from a variety of file formats, namely, xyz, DMol, MOPAC, MOL2, Chem36, and pdb. The user can specify the units of the coordinates, and the display type via widgits.

crystal

Author :	**Keith Refson** Earth Sciences Department, Oxford University

Number :	1833
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	Sun

This is not *quite* a new module, rather it is a successor to "read_shak" which now becomes obsolete. CRYSTAL is a crystal structure viewer for AVS. It accepts files in one of several formats - currently SCHAKAL, SHALXL and PDB, although I hope that some enterprising person may add more readers. The SCHAKAL and SHELXL formats fully support generation of complex structures from an assymetric unit with space-group symmetry operations which can be typed in almost verbatim from the "International Tables". It has facilities for replication of unit cells, slicing the displayed structure using a miller plane, calculation of interatomic distances and variations on the ball-and-stick/space-filling display. The distribution includes a small number of example datasets, though I can supply more on request. This actually comes in 2 pieces. crystal - an AVS module for displaying crystal structure or molecular models, and cryst_to_geom - an offline filter for conversion of various crystal structure files to geometry.

curdle

Author :	**Larry Gelberg**
	Stardent Computer
Number :	1025
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

CURDLE creates a fractal image of a 'curdle' or Cantor dust. This kind of fractal is based on a square 'kernel' whose size is controlled by the KERNEL_SIZE parameter. Every element in the kernel gets recursively turned into a new copy of the kernel. The number of levels of recursion are controlled by the RECURSION parameter. Every element in the kernel which is ACTIVE gets evaluated further. Inactive kernel elements do not get evaluated. You activate kernel elements by dialing the CELL_NUMBER to a particular cell within the kernel and hitting the ACTIVE button.

cylinders

Author :	**Eric Sills**
	North Carolina
	Supercomputing Center
Number :	1298
Language :	C
Submitted :	08/26/92
Version :	1.000
Last Updated :	08/26/92
Ported To :	DEC Sun HP

Cylinders reads a file containing cylinder center point coordinates (x,y,z), cylinder radius, and cylinder height, one per line of the input file. Cylinders builds a polyhedron object for each cylinder (surface normals are computed for shading).

draw_arrow

Author :	**anonymous**
Number :	1888
Language :	C
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_ultrix DEC_alpha

This modules creates a 3 dimensional arrow for use in the geometry viewer. The arrow can then be moved and rotated so that it points at a particular object or interesting data value. The arrow can be straight, or it can have a bend in the shaft. The arrow is created each time that the 'Make Arrow' button is pressed, removing any previous arrow in the process. Therefore, to create two arrows, you must use two modules.

draw_polygon

Author :	**anonymous**
Number :	1889
Language :	C
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_ultrix

This module reads an ascii input file and creates a set of polygons in a single geometry object. The module expects the file to have an extention of ".ply" or ".txt". Polygons can have different numbers of vertices. The information that must be supplied in the ascii file is the number of polygons, the number of vertices in each polygon, and the coordinate information for the vertices. Additionally, normal information may be supplied for each polygon. In this module, it is assumed that the polygons are flat, and therefore only require 1 normal per polygon, instead of one per vertex. If the normal information is not present, the vertices are assumed to be arranged counterclockwise from the "outside" viewpoint.

endif

Author :	**John Tee**
	AVS Inc. Consultant
Number :	1336
Language :	FORTRAN
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	DEC HP

Two modules, "if" and "endif" can be used to make a logical branch in a flow network. The "if" module copies the input to one of two output ports, based on a boolean flag, while "endif" copies one of two input ports data to the output port, closing the condition. Different processing modules can then be placed along the two logical branches of the network. Both modules are in Fortran, and dynamically allocate new space for the output fields.

extruder

Author :	**Larry Gelberg**
	Advanced Visual Systems
Number :	1632
Language :	C
Submitted :	07/05/93
Version :	1.000
Last Updated :	07/05/93
Ported To :	HP IBM Kubota DEC Sun

The extruder module is one of a suite of four AVS modules which can be used to design simple woodworking projects. For more information on the purpose of the project, please read the file called "Poster_Session" and the man pages for each module. The four modules are "cabinet maker" - generates structures made of boxes, "taperer" - generates structures made of tapered boxes, "extruder" - creates capped, extruded tubes from lists of X/Y pairs, and "grid_gen" - creates measured grids so that accurate models can be constructed. Due to the current structure of the IAC's directories, this suite has been split into 4 separate directories on the ftp site. However, a compressed tar archive of all 4 mods is included with each of the directories as wood_mod_suite.tar.Z. Also note that there are several images contributed with this module suite. These have been stored on the ftp site in the directory at avs.ncsc.org called sample_data/avs_data/woodworking_ima ages.

fast_animate

Author :	**Keith Refson**
	Oxford University
Number :	1181
Language :	C
Submitted :	04/30/92
Version :	1.000
Last Updated :	04/30/92
Ported To :	DEC IBM Sun Kubota HP

The fast animate module implements an animation of a sequence of AVS geometry files by reading them into memory and sending the geometries in rapid succession to its output port. By storing the geometries in memory it can achieve substantially better animation rates than the use of a cycle object in a script would. Furthermore the animation rate is independent of the length of the sequence until physical memory is exhausted. The module was developed for visuali- zation of molecular dynamics simulations where the geometries are "ball and stick" representations of molecular assemblies, but is not restricted to such.

field2_from_Math

Author :	**Oleg Perelet, Wolfram Research**
Number :	1542
Language :	C
Submitted :	10/14/92
Version :	1.000
Last Updated :	10/14/92
Ported To :	Sun DEC Kubota

The "field2 from Math" module allows a two-dimensional scalar field to be supplied to AVS. When the "field2 from Math" module is invoked an xterm is started running Mathematica with the input and output of the xterm directed to and from the Mathematica session main loop. Entering Mathematica commands into the window will result in the appropriate Mathematica results. The session is initialised with a package of Mathematica commands which have started a MathLink communication channel to AVS. Other initialisations are made so that the Mathematica command AVSWriteField[data] will write a three-dimensional scalar field to AVS. Please note that at the present time the IAC does NOT have Mathematica installed on a , Sun, or Kubota platforms. As a result this module was placed on the ftp site using the Makefiles provided by the author, without testing.

field3_from_Math

Author :	**Oleg Perelet**
	Wolfram Research
Number :	1540
Language :	C
Submitted :	10/14/92
Version :	1.000
Last Updated :	10/14/92
Ported To :	Sun DEC Kubota

The “field3 from Math” module allows a three-dimensional scalar field to be supplied to AVS. When the “field3 from Math“ module is invoked an xterm is started running Mathematica with the input and output of the xterm directed to and from the Mathematica session main loop. Entering Mathematica commands into the window will result in the appropriate Mathematica results. The session is initialised with a package of Mathematica commands which have started a MathLink communication channel to AVS. Other initialisations are made so that the Mathematica command AVSWriteField[data] will write a three-dimensional scalar field to AVS. Please note that at the present time the IAC does not have Mathematica installed on a , Sun, or Kubota platform. As a result this module was placed on the ftp site using the Makefiles provided by the author, without testing.

file_path_and_na

Author :	**John Stephen**
	Tessella Support Services
Number :	1655
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

This module takes an interactive file browser widget, and outputs the actual filename and the full absolute directory path as separate string output ports.

file_to_field

Author :	**Phil McDonald** NOAA/ERL/Forecast Systems Laboratory
Number :	1267
Language :	C
Submitted :	05/22/92
Version :	1.100
Last Updated :	04/21/93
Ported To :	HP Sun Kubota

The file to field module creates a uniform field from an ASCII or binary file. The field parameters that are normally contained in the header of a field file are specified interactively through the file to field parameters.

flight_path

Author :	**Ian Curington** Advanced Visual Systems
Number :	1656
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

The Flight Path Module takes a scatter field position list and animates the camera path along the trajectory. Derived from camera.c in /usr/avs/examples. The scatter path field is in the form of "field 1D 3-space irregular float" where the coordinate information in the field specifies the path. The data values are ignored. It moves %top, not the camera, so multiple camera views can be set up to watch the path. The file_descriptor or read_field modules can be used to get the path from an external file. Animated Integer can be used to control time, for a moving sequence.

fractal_3d_f

Author :	**Anonymous** Stardent Computer
Number :	1036
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Produce mid-point displacement fractal in two or three dimensions. Controls are max width, power, and reseed value.

fractal_field

Author :	**David E. Harris** Georgia Tech Research Institute
Number :	1205
Language :	C
Submitted :	05/11/92
Version :	1.000
Last Updated :	05/11/92
Ported To :	Kubota

This module generates a 2D scalar byte field whose elements are statistically self-similar (fractal) in distribution without regard to scale. (I.e. size. Not to be confused with the intensity scale value parameter.) A gaussian distributed random field is first created and then transformed to the frequency domain by FFT. The result is then filtered to agree with the fractal power law and inverse transformed. The second ouput port provides access to the gaussian random field used to generate the fractal field.

gaussian_field

Author :	**David E. Harris** Georgia Tech Research Institute
Number :	1204
Language :	C
Submitted :	05/11/92
Version :	1.000
Last Updated :	05/11/92
Ported To :	Kubota

This module generates a 1-, 2-, or 3-D fields of arbitrary size whose elements are gaussian-distributed random floating point values.

generate_contine

Author :	**Ed Bender** Convex Computer Corporation
Number :	1594
Language :	C
Submitted :	03/04/93
Version :	3.000
Last Updated :	03/09/93
Ported To :	Kubota IBM DEC

This module is useful for decorating globes that represent the Earth. Given an input radius (Earth radius) a geometry is generated consisting of multiple polylines that show the outlines of the continents and islands of the earth. Optionally, a single sphere is generated also at a slightly smaller radius than the continental outlines. Version 2.0 has fixes by NCSC's Will Ivey for DEC version. Version 3.0 has fixes by ORNL's Ray Flanery for Kubota version.

generate_gri

Author : **Larry Gelberg**
Stardent Computer
Number : 1037
Language : C
Submitted : 10/10/91
Version : 1.000
Last Updated : 10/10/91
Ported To : Kubota DEC Sun HP

Generate_grid creates a colored image of a grid of one color superimposed over a background color. You can control the colors of the grid with the FOREGROUND HUE, SAT(uration), and VALUE dials, the color of the background with BACKGROUND HUE, SAT(uration), and VALUE dials, the thickness of the grid lines with the GRID WIDTH dial, the size of the squares between the grid liines with the GRID SPACING dial, and the resolution of the output image with the RESOLUTION typeins.

genloadmd

Author : **Ken Flurchick and Mark Reed**
North Carolina Supercomputing Center
Number : 1603
Language : C
Submitted : 04/09/93
Version : 1.000
Last Updated : 04/09/93
Ported To : DEC IBM Kubota

This module is designed to read in position and velocity information from the file specified as “Input File”, for a group of point particles in three dimensions. The program represents each point with a sphere at the position specifed and color coded by the magnitude of the velocity. The colors 0-255 are specified in the colors.h file and range from blue (slowest) to red (fastest). Initially the module will compute the minimum and maximum speeds from the data set read in and echo these back to the parameter widgets. The user can change the size of the spheres and the min and max values used to determine the colors of the sphere. Note that if the minimum speed is greater than or equal to the maximum speed then a message is echoed back to standard output (the screen) and no action is taken. Sphere color is determined by a linear interpolation between the minimum and maximum speeds of the particle velocity magnitude.If the velocity magnitude of a particular sphere exceeds the maximum speed then it is displayed as white while those less than the minimum are displayed as black (note - change the background color if this also is black).

geom_from_Math

Author :	**Oleg Perelet**
	Wolfram Research
Number :	1541
Language :	C
Submitted :	10/14/92
Version :	1.000
Last Updated :	10/14/92
Ported To :	Sun DEC Kubota

The "geom from Math" module allows three-dimensional graphics commands in Mathematica to supply AVS with an input geometry. When the "geom from Math" module is invoked an xterm is started running Mathematica with the input and output of the xterm directed to and from the Mathematica session main loop. Entering Mathematica commands into the window will result in the appropriate Mathematica results. The session is initialised with a package of Mathematica code which starts a MathLink communication channel to AVS. Other initialisations are made so the typical Mathematica three dimensional graphics commands such as Plot3D or ParametricPlot3D send their results into AVS rather than rendering them in the more typical Mathematica ways. Please note that at the present time the IAC does not have Mathematica installed on a , Sun, or Kubota platform. As a result this module was placed on the ftp site using the Makefiles provided by the author, without testing.

geometry_sphere

Author :	**Marc Curry**
	North Carolina
	Supercomputing Center
Number :	1320
Language :	C
Submitted :	08/30/92
Version :	2.000
Last Updated :	10/02/92
Ported To :	Sun

This geometry sphere module demonstrates how to convert a set of vertices into a geomtry format for viewing within AVS. Also, some simple widgets are provided to take care of (what were) program parameters. The sphere is actually a triangle mesh approximating a sphere by recursive subdivision into more and more triangles. The first approximation is an octahedron, with each level of refinement increasing the number of triangles by a factor of 4.

graduate32

Author :	**David Haith**
	Power-Media
Number :	1041
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Generates shaded 2D images for creating synthetic backgrounds. Parameters control degree of ramp, starting and ending values. Gradation is top to bottom, with horizontal variation constant. RGB color specs are given for both top and bottom. A nice noise source is available for mixing with the ramp. Output can be seen with "display image".

grid_generator

Author :	**Larry Gelberg**
	Advanced Visual Systems
Number :	1633
Language :	C
Submitted :	07/05/93
Version :	1.000
Last Updated :	07/05/93
Ported To :	HP IBM Kubota DEC Sun

The grid_generator module is one of a suite of four AVS modules which can be used to design simple woodworking projects. For more information on the purpose of the project, please read the file called "Poster_Session" and the man pages for each module. The four modules are "cabinet maker" - generates structures made of boxes, "taperer" - generates structures made of tapered boxes, "extruder" - creates capped, extruded tubes from lists of X/Y pairs, and "grid_gen" - creates measured grids so that accurate models can be constructed. Due to the current structure of the IAC's directories, this suite has been split into 4 separate directories on the ftp site. However, a compressed tar archive of all 4 mods is included with each of the directories as wood_mod_suite.tar.Z. Also note that there are several images contributed with this module suite. These have been stored on the ftp site in the directory at avs.ncsc.org called sample_data/avs_data/woodworking_ima ages.

harwell_colo

Author :	**Anonymous**
	Stardent Computer
Number :	1042
Language :	C
Submitted :	10/10/91
Version :	2.000
Last Updated :	09/30/93
Ported To :	Kubota DEC IBM Sun HP

Harwell is a special color map generator. It provides a standard fringe color set, based on a transfer equation. It is gamma corrected, and has essentially three regions, with linear interpolation between the regions. The regions are blue –¿ green –¿ yellow –¿ red The cutoff points are controlled by the three float dial parameters. The gamma is also adjustable. The default values are designed for the "standard fringe". Version 2.0 update created by Ian Curington of Advanced Visual Systems, Incorporated.

if_2

Author :	**John Tee**
	AVS Inc. Consultant
Number :	1337
Language :	FORTRAN
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	HP

Two modules, "if" and "endif" can be used to make a logical branch in a flow network. The "if" module copies the input to one of two output ports, based on a boolean flag, while "endif" copies one of two input ports data to the output port, closing the condition. Different processing modules can then be placed along the two logical branches of the network. Both modules are in Fortran, and dynamically allocate new space for the output fields.

image_snap

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1661
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

This module uses the CLI to cause image zoom and pan operations on one image to be echoed to a second image. A "snap" button is used to cause the transformations. Two string type-in controls specifiy the original and target image names in the image viewer.

int_positiv

Author :	**Richard Albrecht** Max-Planck-Institute for Biochemistry
Number :	1725
Language :	C
Submitted :	12/02/93
Version :	1.000
Last Updated :	12/02/93
Ported To :	Kubota Sun

This is a short module to adjust the 'tracer' to 'downsize'. It works in connection with 'downsize points'. The first parameter is a replacement of 'integer'. Only positive values are allowed. The second parameter is multiplied with the first to give an adjustment of the alpha value in 'tracer'. Now it is possible to use 'downsize' with the 'tracer'. The size and the opacity of the image is independent from the downsize value.

integer_list

Author :	**John Stephen** Tessella Support Services
Number :	1663
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

Converts a list of integers within a string (blank-delimited) to a field of integers (a 1D array).

integer_step

Author :	**Young S. Ham** Naval Research Laboratory
Number :	1771
Language :	C
Submitted :	02/09/94
Version :	1.000
Last Updated :	02/09/94
Ported To :	HP Kubota Sun

The integer_step module supplies an integer value to one or more receiving modules from an intial integer. The current integer is incremented by step_size by means of a mouse click on a button. This module is very useful when viewing images or to fire an input value in order is desired. One does not have to keep typing an integer or play with the dial widget for a new input value.

jigsaw

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1665
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

Jigsaw creates a simple geometric object of extruded shapes, with input file control over contour of shapes. An AVS field controls the height, color, or visibility of each block. This module is similar to "Read Blokjes" at the IAC. Some degree of concave outline structure is allowed. Each piece has a unique name, and can be manipulated independenty in the geometry viewer, if lock mode is not selected. The color range is defined externally in the colormap data structure.

lathe

Author :	**Ian Currington**
	Stardent Computer
Number :	1048
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Sun HP

This module is a C co-routine, that includes an X-window application. It has an interactive pallet window, for defining the profile of a "lathe" object. This profile is turned into a 3D geometric object by surface of revolution.

lines_to_geom

Author :	**Tim O. Robinson**
	UC Berkeley
Number :	1272
Language :	C
Submitted :	05/27/92
Version :	1.000
Last Updated :	05/27/92
Ported To :	DEC IBM Sun Kubota HP

This module reads an ascii input file and creates a set of 3D line segments in a single geometry object. The module expects the file to have an extention of ".xyz". Each line segment is assumed to have the same number of vertexes. Duplicate vertexes can be used to overcome this but they are not ignored in this version.

load_md

Author :	**David Bock** North Carolina Supercomputing Center
Number :	1109
Language :	C
Submitted :	12/05/91
Version :	1.000
Last Updated :	12/05/91
Ported To :	Kubota DEC IBM Sun HP

AVS subroutine module source to read into memory a series of time steps from a specified input file and provide the user with the ability to select an individual time step or cycle through a series of selected time steps. Module has been used to visualized the dynamics of a given set of molecules over a specified period of time. Module simply loads the user-specified time steps of molecular coordinates and generates a sphere for every molecule at the stored coordinates.

load_mdn

Author :	**Ken Flurchick** North Carolina Supercomputing Center
Number :	1583
Language :	C
Submitted :	02/11/93
Version :	1.000
Last Updated :	02/11/93
Ported To :	DEC IBM Kubota HP

This is an enhancement of the load_md module already available. This module is designed to read in one time step of data from a series of input files and provide the user with the ability to select an individual time step or cycle through a series of selected time steps. Module has been used to visualized the dynamics of a given set of molecules over a specified period of time. Module simply loads the user-specified time steps of molecular coordinates and generates a sphere for every molecule at the stored coordinates. This module has the option of specifying the data file through animated filename, for multiple files, or with a file browser widget for a single file. The default is to use the animated filename input port, but if this value is set to $NULL then the module will use the file browser.

macro_menu

Author :	**John Stephen, Tessella Support Services** and Ian Curington, Advanced Visual Systems, Inc.
Number :	1719
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP Kubota Sun

This module navigates a menu structure which can be designed and configured by the user. Each menu level is described by a single ASCII text file. A menu level contains several buttons with user-configurable names. Each button results in one of three basic kinds of action in the module - load a new menu level, pass a command to downstream modules, or go to a file to find a series of commands to be passed to down stream modules. Option 3 allows multiple commands to be played in background as the result of a single button press. The file containing the commands to be played in background is the second type of file associated with this system. Thus there are two types of file - the MENU FILE to define a menu level, and the AUTOMATIC FILE which contains a series of commands to be executed.

mandelbrot

Author :	**Larry Gelberg** Stardent Computer
Number :	1108
Language :	C
Submitted :	11/27/91
Version :	1.000
Last Updated :	11/27/91
Ported To :	Kubota DEC IBM Sun HP

Generates the mandelbrot set, at a specific location, for a certain number of iterations, at a certain size, all determined by paramter widgets.

mask_generator

Author :	**David E. Harris**
	Georgia Tech Research Institute
Number :	1201
Language :	C
Submitted :	05/08/92
Version :	1.000
Last Updated :	05/08/92
Ported To :	DEC Kubota

This module generates a field that is intended to be interpreted as a binary mask for filtering FFTs, a gray scale image, or a structuring element for various morphological operations. The field can be any size from 2x2 pixels on up and an ellipse of any size major and minor axes from one pixel on up. The background can be either black or white and the intensity value of the ellipse can be any gray level. The position of the ellipse within the field is arbitrary and can intersect the edges of the field or be outside the field entirely.

menu_example

Author :	**Larry Gelberg**
	Stardent Computer
Number :	1053
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

This is an example of how you can use the layout feature of AVS to implement multiple level menus. When the selection "level_1" is madea menu showing the choices level2a1, level2a2, level2a3 is displayed. When the selection "level_2" is made, the choices level2b1, level2b2, level2b3 are displayed. There are several ways in which this is accomplished in the source code (1) the call AVSadd_parameter_prop() is used to set up the layout (2) the call AVScommand is used to issue CLIO calls from within the module body.

microscope

Author :	**Hennekens Didier** University of Mons-Hainaut (Belgium)
Number :	1283
Language :	C
Submitted :	06/07/92
Version :	1.000
Last Updated :	06/07/92
Ported To :	DEC IBM HP

This avs subroutine transforms a picture of atomic force microscope in the "scalar uniform field" structure as defined by AVS. This picture consists of a set of heights on a square uniform grid. The informations on this picture are contained in a ascii file. The expected ascii format is very simple. It consist of a header of three lines (the file name and two experimental parameters of the atomic force microscope - the z_scale and the z_dsclale) followed by data.

mk_vel2

Author :	**Ian Curington** Advanced Visual Systems
Number :	1716
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP IBM Kubota Sun

This module generates a synthetic 3D field, with random perturbations in both coordinates and data values. The 3D scalar field is the square of the distance from a point on the center of the bottom plane. An isosurface of this is approximately a parabola. This was used in a demonstration of seismic velocity modelling.

mk_vel

Author :	**Ian Curington** Advanced Visual Systems
Number :	1715
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP IBM Kubota Sun

This module generates a synthetic 3D field, with random perturbations in both coordinates and data values. The 3D scalar field is the square of the distance from a point on the center of the bottom plane. An isosurface of this is approximately a parabola. This was used in a demonstration of seismic velocity modelling.

multiformat_read

Author :	**A. Jacobson** UCLA Dept. Pharmacology / Crump Institute
Number :	1849
Language :	C
Submitted :	10/12/94
Version :	1.000
Last Updated :	10/12/94
Ported To :	DEC HP IBM Sun

This module reads an image from disk into AVS in any of a variety of formats. Some of the formats which this module supports are from San Diego Supercomputing Center's image tools package, others are supported from John Cristy's ImageMagick package, still others require tools from GNU's Ghostscript, and the NetPBM package. Depending on how the module is configured, it can read and about 30 to 40 image formats, including , AVS X, Alias, Apple Macintosh MacPaint, Apple Macintosh QuickDraw/PICT, CCIR 601, Compuserve Graphics, Flexible Image Transport System (monochrome only), Hierarchical Data File, JFIF Rev1.02, Khoros VIFF, Magick, Mark VanDeWettering's MTV ray tracer image format, Microsoft Windows bitmap, Microsoft Windows cursor, Microsoft Windows icon, PIXAR picture, Photo CD, PostScript, Radiance, Radiance, SDSC Synu, SGI RGB, Sun Icon and Cursor, Sun Rasterfile, Sun TAAC Image, Tagged, Truevision Targa, Utah Run length encoded, Wavefront raster, X Window System window dump, X11 bitmap, X11 pixmap, ZSoft IBM PC Paintbrush, binary or ASCII PBM Portable Bit Map, binary or ASCII PBM Portable Gray Map, binary or ASCII PBM Portable Pixel Map, binary or ASCII PBM Portable aNy Map. This module contains a superset of the formats in the READ_ANY_IMAGE module, with several new formats added and improved support for several of the original file types. Several sample images (gif, pict, ppm, ps, rgb, rle, tiff, xwd, and yuv) are included with this module for your testing purposes. Please note that this module calls several executable files, including imconv, convert, and gzip. Platform specific binaries for several of these programs are available from the IAC ftp site with this module. See the .txt file of this module or the file multiio.h for further details on the sources for this module's tools. The tools are stored at the IAC in one large compressed tar file, which is 11MB and is stored at avs.ncsc.org in /misc_tools/multiio_tools/multiio_tools.tar.Z. NOTE - THIS MODULE IS AWESOME ... - Steve

named_toggle

Author :	**John Stephen**
	Tessella Support Services
Number :	1672
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

This creates a "toggle" parameter button, whose value is available as a toggle output port, for remote control of other modules with toggle parameters. The name on the button is controlled dynamically from a second string input parameter.

newfilla

Author :	**Ken Flurchick**
	North Carolina Supercomputing Center
Number :	1582
Language :	C
Submitted :	02/11/93
Version :	1.000
Last Updated :	02/11/93
Ported To :	DEC IBM Kubota HP

Inputs a field from "ccyl_pipe" (available via anonymous ftp) and fills a cylindrical geometry. This module will read each time step of data from a separate file using "animate filename"

noise_source

Author :	**David Haith**
	Power-Media
Number :	1105
Language :	C
Submitted :	11/26/91
Version :	1.000
Last Updated :	11/26/91
Ported To :	Kubota

Generates random 2D fields for creating synthetic backgrounds, or for mixing with other images for various effects. Parameters control size of image, and character of random field.

pdb_to_geom

Author :	**Advanced Visual Systems, Inc.**
Number :	1551
Language :	C
Submitted :	11/04/92
Version :	1.000
Last Updated :	11/04/92
Ported To :	Kubota HP

The pdb to geom module reads the description of a molecule from a file in the Brookhaven Protein Data Bank (PDB) data format. Typically, such files have a filename suffix. The output is an AVS description of the molecule.

phoenics_int

Author :	**Michael J. Rangitsch**
Number :	1175
Language :	FORTRAN
Submitted :	04/16/92
Version :	1.000
Last Updated :	04/16/92
Ported To :	Kubota IBM HP

This module converts a phi.dat data file produced by PHOENICS into an AVS field. The type of the field is computed by the header information in the phi.dat file. Cartesian, cylindrical and BFC cases are all supported. Null velocity fields are produced for 1 and 2 dimensional cases to fill out three velocity components needed for modules such as the particle advector.

pick_center

Author :	**Ian Curington** Advanced Visual Systems
Number :	1752
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This module allows the user to pick on anything in the GeometryViewer scene that is not already registered for upstream picking, and will set the current rotate / zoom center to be this picked point. It is most useful if you see a large complexe scene, pick on the piece you want to zoom into, then zoom into it. Without this module, if it was on the edge of the object it would disappear from view as soon as you zoomed in.

place_camera

Author :	**Jeff Vroom** Advanced Visual Systems
Number :	1676
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

This module is a prototype of a camera placement system. Two geometry viewers are used, where the first has the object list plus an iconic representation of the camera, with view frustrum shown in wire frame. The second view is the scene that camera "sees" as placed in the first scene. The module shows how to manipulate and control camera and scene transformation matrices.

plate

Author :	**Jeff Vroom**
	Advanced Visual Systems
Number :	1063
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

This is a simulation of a vibrating tympanic membrane written as a co-routine. The coefficients A,B,C,D, and E relate to the various harmonic strenths. The "sleep" button turns the simulation on and off.

plato_super

Author :	**Larry Gelberg**
	Stardent Computer
Number :	1080
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

This module generates GEOM versions of the five platonic solids. It also allows for both primary and secondary stellations of the faces and random coloring of the vertecies. The 'Stellate 1' knob pokes the centroid of each face in or out. The 'Stellate 2' knob then does the same thing to each of the resulting triangles. The coloring selections provide either pre-chosen coloring schemes or ither two or three random colors per triangle. Hitting these buttons again and again will give you continuously different colorings.

plot_xyz

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1064
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

This module reads an ASCII file consisting of lists of x,y,z points and constructs a polyline geometry object using the points in the file as vertices.

plot_xyz_col

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1065
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Sun HP

This module reads an ASCII file consisting of lists of x,y,z points and constructs a polyline geometry object using the points in the file as vertices. Additionally, color information is read in on a per-vertex basis, and the line segments are color coded as a function of the color index specified in the user file and the input colormap.

r_tif_ra

Author :	**Richard Albrecht** Max-Planck-Institute for Biochemistry
Number :	1852
Language :	C
Submitted :	10/17/94
Version :	1.000
Last Updated :	10/17/94
Ported To :	DEC_alpha HP IBM Sun

r_tif_ra reads 24 bit, 8 bit grey and 8 bit colormap TIFF-files. This is a special reader for uncompressed TIFF-files. It converts 24 bit, 8 bit grey, and 8 bit colormap TIFF-images to an AVS image. The output is in any case a normal AVS image. Opacity of the image is always set to 255, whereas the red, green and blue channels are copied directly (24 bit) or set to the values of the colormap. The Colormap itself (if is) is not copied. A short report is copied to 'stdout' with the toggle.

r_tifvol_ra

Author :	**Richard Albrecht** Max-Planck-Institute for Biochemistry
Number :	1853
Language :	C
Submitted :	10/17/94
Version :	1.000
Last Updated :	10/17/94
Ported To :	DEC_alpha HP IBM Sun

r_tifvol_ra reads a serie of TIFF-files of the form *.nnn This is a special reader for uncompressed TIFF-files. It reads only 8 bit grey TIFF-images. A Colormap is ignored. With a modification of the normal AVS-field descriptor '*.fld' a complete image serie is read in. The filenames must be in MS-DOS style (cccccccc.nnn). The 3 characters after the dot are the file number (*.000, *.001 ...). The output is an AVS-field.

readFLOW3D

Author :	**Michael Rangitsch**
Number :	1174
Language :	C
Submitted :	04/16/92
Version :	1.000
Last Updated :	04/16/92
Ported To :	Kubota HP

The read_flow3d module ascesses the dump files via a browser. Prior to selecting the filename, use the parameter toggles to identify which turbulence model, which combustion model, the number of scalar species were used in the model and whether density or viscosity are stored, if the convection coefficients are desired or if the problem was in cylindrical coordinates. The reader has no way of knowing from the dump file alone.

read_16_bit

Author :	**Larry Gelberg** Stardent Computer
Number :	1103
Language :	C
Submitted :	11/22/91
Version :	1.000
Last Updated :	11/22/91
Ported To :	Kubota DEC IBM Sun HP

READ 16 BIT IMAGE reads a 16 bit integer binary image (2D collection of bytes) off of disk into an AVS integer (32 bit) field. It does this by reading the 16 bit integers into a "short" buffer and then casting them into 32-bit integers. If you do not know what the resolution of the images are, you can use the WIDTH and HEIGHT controllers like "vertical" and "horizontal" hold to experiment with until the image looks like you expoect.

read_ACR_NEMA

Author :	**Andrea Marchetti** Istituto di Elaborazione dell'Informazione del CNR
Number :	1885
Language :	C
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_ultrix DEC_alpha

The read ACR-NEMA module reads ACR-NEMA SPI data files and converts them into AVS field format. SPI is a standard based on the old ACR/NEMA standard used by Siemens and Philips. SPI make use of ACR/NEMA data elements nd groups. SPI store the integer in little endian format (Least Significant before Most Significant).

read_AIFF

Author :	**Charles J Williams III**
	Naval Research Lab
Number :	1543
Language :	C
Submitted :	10/14/92
Version :	1.000
Last Updated :	10/14/92
Ported To :	Kubota

This module is meant to be a companion to the Write_MooV module. It operates rather simply, reading in an AIFF file (this is a rather Macintosh format, although many programs can translate understand AIFF) and generating a 1D field.

read_Dore_im

Author :	**Anonymous**
	Kubota Pacific Computer Inc
Number :	1153
Language :	C
Submitted :	03/12/92
Version :	1.000
Last Updated :	03/12/92
Ported To :	Kubota DEC IBM Sun HP

Reads a raster Dore image file output.

read_F3D

Author :	**Michael Rangitsch**
Number :	1537
Language :	C
Submitted :	10/03/92
Version :	5.000
Last Updated :	07/15/94
Ported To :	DEC HP Sun

This module reads an ascii formatted multiblock FLOW3D solution file (from Harwell FLOW3D version 3) and stores the cfd data as a ucd data set. The dummy nodes stored at the edges of each computational block are removed by the module. The n-vector data at each computational cell is stored as cell data, to use most of the ucd modules, this must be converted to nodal data. The cells of the ucd structure are all hexahedra. See the sample data directory on the International AVS Center's anonymous ftp site for two sample data sets to read with this module. Version 2.000 has several enhancements and debugging features added. Version 3.000 adds geometry representation of 2 dimensional patches. There are 7 types of 2D patches defined by FLOW3D. This code probably doesn't work for 3d patches. Version 4.0 handles unanticipated data sets, and processes cylindrical data sets and single block data sets, as well as several other enhancements. Version 5.0 handles unformatted as well as formatted data.

read_FLOW3D

Author :	**Anonymous**
	Stardent Computer
Number :	1035
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Sun

The read_flow3d module is used for reading/converting CFD data and Mesh files from the Harwell "FLOW3d" code into AVS field data types. The read_flow3d module ascesses the dump files for both turbulence and temperature information, separately.

read_GOCAD

Author :	**Ian Curington** Advanced Visual Systems
Number :	1844
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP Sun

This is a data import reader module for the GOCAD (ASCII) file format. The GOCAD application models, edits, and describes 3D geologic structures, containing horizon surfaces, faults, and wells. The GOCAD application is authored by a group in Nice, France. This module reads the contents of one GOCAD file, and converts this to the AVS geometry data type, which can be sent to the Geometry Viewer for interactive display. Axis specific scaling is allowed by using the floating point parameters, supplied with dial controls by default. The module ignores all other lines from the GOCAD file except for vertices and trianges. It looks at the first character of every line. If the the line starts with "V", then it reads the id, and the position x, y, z. If the line starts with "T", it reads three index numbers for which set of verts to tie together as triangles.

read_HDF_SDS

Author :	**Bill Sherman** National Center for Supercomputing Applications
Number :	1179
Language :	C
Submitted :	04/22/92
Version :	1.000
Last Updated :	04/22/92
Ported To :	DEC Sun IBM

The "read HDF SDS" module reads Scientific Data Sets (SDS) from a HDF files, outputting up to four AVS field data sets. HDF is the Hierarchical Data Format developed at the National Center for Supercomputing Applications (NCSA). The SDS is only one of several types of data that may be contained in an HDF file, but it is the only type recognized by this module.

read_Leica_confo

Author :	**A. Jacobson** UCLA Dept. Pharmacology / Crump Institute
Number :	1855
Language :	C
Submitted :	10/17/94
Version :	1.000
Last Updated :	10/17/94
Ported To :	HP Sun

Read Leica Confocal module, designed to read in output from the Leica CLSM in it's native file format (not TIFF). The module is looking for a imagename.info file which it parses for the image plane numbers and plane sizes. It uses these values to open each image file in order. The contents are deposited in a 3D scalar byte field at the output port. Lots of error checking is done to make sure the .info file is not corrupted, and the data files read correctly.

read_abekas

Author :	**Ian Currington** Stardent Computer
Number :	1002
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

This module provides I/O capability to iamges in the native machine format used by the Abekas A60 digital video disk recording system. The image file specification is for the PAL (50 Hz,625 line) version, but can easily be modified to support the NTSC version, or both. The Abekas A60 holds 750 frames, or 30 seconds of live broadcast standard video imagery, in the digital storage format specified by the CCIR-601 specification. Although the disk format is Y-uv, the disk has firmware processing to encode/decode to 3-byte Red-Green-Blue images on-the-fly during the tcp/ip file transfers.

read_blokjes

Author :	**Ian Curington** Advanced Visual Systems
Number :	1332
Language :	C
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	DEC Sun HP

Read Blokjes is a module that imports blocks, or bricks, and creates geometric objects appropriate for the Geometry Viewer. The format is from a CFD package called FLOWTHERM from Flowmerics Ltd., Kingston, Surrey, England, KT2 5AA. This module is only one part of a more extensive interface system. As each block is separate, the property on each one can be edited, changing colour, transparency, rendering mode, etc. Each block has a name, type, and start and end corner points.

read_compressed

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1128
Language :	C
Submitted :	03/15/92
Version :	2.000
Last Updated :	05/15/92
Ported To :	IBM Sun Kubota HP

The "read compressed image" module provides the same functionality as the AVS-supplied "read image" module, with the added twist that the image is in a "compressed" format. An AVS image may be compressed using the UNIX(tm) compress utility, or by using the sibling module "write compressed image." Compression ratios of 1 to 10 are easily achieved using this compression scheme.

read_dem

Author :	**anonymous**
Number :	1887
Language :	C
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_ultrix DEC_alpha

This AVS module reads in an ARC/Info Digital Elevation Model (DEM) into AVS.

read_dyna3d

Author :	**David Parker**
	Stardentand Livermore Software Technology
Number :	1100
Language :	CandFORTRAN
Submitted :	11/22/91
Version :	1.000
Last Updated :	11/22/91
Ported To :	IBM

DYNA3D is a finite element analysis program developed by This module interprets the ASCII input file for this program and generates GEOM representations of the desired aspects of the data sets.

read_eps_image

Author :	**Terry Myerson**
	International AVS Center (NCSC)
Number :	1213
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an eps image in the eps format into an AVS Network. Encapsulated PostScript color, grayscale, and black-and- white image files are used by a variety of PostScript tools in order to include PostScript diagrams and images within other documents. Encapsulated PostScript files are recognized by the follow- ing filename suffixes .eps, .epi, .epsi, and .epsf.

read_gif2

Author :	**John Langner**
	Stardent Computer
Number :	1303
Language :	C
Submitted :	07/11/92
Version :	1.000
Last Updated :	07/11/92
Ported To :	IBM Sun HP

read_gif2 is yet another module to read a Compuserve GIF image file.

read_gif

Author :	**Larry Gelberg**
	Stardent Computer
Number :	1101
Language :	C
Submitted :	11/22/91
Version :	1.000
Last Updated :	11/22/91
Ported To :	Kubota DEC IBM Sun HP

This module reads a GIF format file and creates a 32-bit ARGB image in the AVS "field 2D 4-vector byte" format. If the GIF file has an embedded color-map, that is used, otherwise, it creates a linear grey-scale ramp.

read_gif_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1214
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the gif format into an AVS Network. GIF (Graphics Interchange Format) is CompuServe's standard for generalized color raster images. This standard is a mechanism to exchange and display high-quality, high- resolution graphics images. CompuServe gif files are recognized by the filename suf- fixes .gif and .giff.

read_grd_to_fiel

Author :	**Andy Bingham** Oxford University, Dept. of Earth Sciences
Number :	1699
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP IBM Sun

The read grd to fld module reads a GMT v1.0 grd file from disk and outputs the file as "field 2D scalar float 2-coord uniform". The output data is in the AVS field format. The field extents are set to the values of east, west, south and north from the input file.

read_hdf_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1215
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the hdf format into an AVS Network. hdf is a generic, tagged Hierarchical Data File format developed by the National Center for Supercomputing Applica- tions (NCSA). hdf files may contain images, scientific data sets, and miscellaneous data items. Such files may be created by several NCSA tools. See the NCSA documentation for details on how to use these tools. hdf format handling within the SDSC image library is limited to images of certain depths and storage methods. hdf files are recognized by these filename suffixes .hdf, .df, and .ncsa.

read_icon_image

Author :	**Terry Myerson**
	International AVS Center (NCSC)
Number :	1216
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the icon format into an AVS Network. icon image files are used by Sun Microsystem's SunView win- dow system, NeWS window system, OpenWindows NeWS tool set, and X11 XView tool set for the storage of icons, cursors, fill patterns, and pieces of widgets (like button check-marks). Sun icon files can be most easily generated using Sun's iconedit(1) icon and cursor editor. The Sun operating sys- tem release includes a directory of standard icons, cursors, background patterns, and widget pieces in icon format in the directory /usr/include/images. See the Sun documentation set for details on how to use the tools dealing with Sun icon files. Sun icon files are recognized by these filename suffixes .icon, .cursor, and .pr.

read_iff_image

Author :	**Terry Myerson**
	International AVS Center (NCSC)
Number :	1217
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the iff format into an AVS Network. iff image files are generated by Sun Microsystems TAAC software such as voxvu(1) and cloudvu(1). See the TAAC-1 Application Accelerator Software Reference Manual for information on how to use these programs. Note that image files compressed with the Sun-TAAC utility make_movie(1) cannot be read by the SDSC image library. iff files are recognized by the following filename suffix .iff.

read_irregul

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1074
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	DEC Sun HP

This module reads data in ascii format into an "irregular field" data structure. The disk file is structured to contain information about the field at the beginning of the file, followed by the data.

read_jpeg

Author :	**John Langner** Stardent Computer
Number :	1099
Language :	C
Submitted :	11/22/91
Version :	1.000
Last Updated :	11/22/91
Ported To :	IBM Sun DEC HP

Read a file containing an image compressed with the JPEG standard. See Also - The best and most readily available introduction to the JPEG compression algorithm is Wallace's article in the April '91 CACM Wallace, Gregory K. "The JPEG Still Picture Compression Standard", Communications of the ACM, April 1991 (v. 34 no. 4), pp. 30-44.

read_mpnt_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1218
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the mpnt format into an AVS Network. The MacPaint mpnt file is the standard Apple Macintosh mono- chrome bitmap image file format. It can be read by many Macintosh graphics applications, and many Macintosh applica- tions that can export bitmap graphics do so in the MacPaint mpnt file format. mpnt files are recognized by these filename suffixes .mpnt, .macp, and .pntg.

read_netcdf

Author :	**Wright, E.L.**
	U.S Geological Survey
Number :	1564
Language :	C
Submitted :	01/22/93
Version :	2.000
Last Updated :	07/15/94
Ported To :	DEC Sun

This module reads data from netCDF files into uniform or rectilinear AVS fields. The data variables must be of dimension 8 or less. Variables that are one-dimensional and have the same name as a dimension name are assumed to be coordinate variables. Coordinate variables, if present, define the coordinate values along each axis of the data space. A maximum of 8 data variables that are NOT coordinate variables may be present.

read_netcdf_example

Author :	**Wright, E.L.**
	U.S Geological Survey
Number :	1210
Language :	C
Submitted :	05/20/92
Version :	1.000
Last Updated :	05/20/92

The read netcdf module is a working example of a module that can read netCDF files. This version can only read netCDF files that contain specific dimension, variable, and attribute names. Also, the coordinate values for the irregular field output are computed in a way that is data specific. The netCDF files to be read by this version of read netcdf must contain the following dimension names xpos, ypos, zpos, and time, where the size of dimension time is declared as UNLIMITED. The following variables must be in the file float variables sigma(zpos), x(ypos,xpos), y(ypos,xpos), and depth(ypos,xpos) and short variables elev(time,ypos,xpos) and salt(time,zpos,ypos,xpos). Variables elev and salt must have the following float attributes scale_factor and add_offset, where the scale factor is to be multiplied first and then the offset added to the data.

read_p_2

Author : **John Langner**
Stardent Computer
Number : 1290
Language : C
Submitted : 07/08/92
Version : 1.000
Last Updated : 07/08/92
Ported To : IBM Sun HP

Read a PC Paintbrush PCX image file.

read_pbm_image

Author : **Terry Myerson**
International AVS Center (NCSC)
Number : 1219
Language : C
Submitted : 05/21/92
Version : 2.000
Last Updated : 05/21/92
Ported To : Sun DEC IBM

This module reads an image in the pbm format into an AVS Network. pbm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. pbm files are recognized by the filename suffix .pbm.

read_pcx_image

Author : **Terry Myerson**
International AVS Center (NCSC)
Number : 1220
Language : C
Submitted : 05/21/92
Version : 2.000
Last Updated : 05/21/92
Ported To : Sun DEC IBM

This module reads an image in the pcx format into an AVS Network. The pcx image file format was invented by ZSoft for use in its IBM PC PC Paintbrush tool series. It has become a defacto standard in the IBM PC world and is regularly used for the storage of monochrome and color pixel information by paint-type tools. See the documentation for each of the IBM PC tools for details on how to use them. ZSoft pcx files are recognized by these filename suffixes .pcx and .pcc.

read_pgm_image

Author : **Terry Myerson**
International AVS Center (NCSC)
Number : 1221
Language : C
Submitted : 05/21/92
Version : 2.000
Last Updated : 05/21/92
Ported To : Sun DEC IBM

This module reads an image in the pgm format into an AVS Network. pgm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. pgm files are recognized by the filename suffix .pgm.

read_pic_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1222
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the pic format into an AVS Network. pic image files are generated by PIXAR programming software, such as ChapVolumes and ChapReyes, the PIXAR Image Runtime Library called Pirl, and the PIXAR rendering tool RenderMan. See the PIXAR documentation set for details on how to use these applications and tools. PIXAR's pic file format is sometimes referred to as PICIO in PIXAR documentation. pic and PICIO mean the same thing. PIXAR's xpic is not the same as pic and is _Φn_Φo_Φt supported by the SDSC image library. _ΦN_Φo_Φt_Φe PIXAR's RenderMan always saves its image files with .pic filename suffixes. However, depending upon output defaults, RenderMan can generate .pic files with PICIO (same as pic) data or TIFF data. .pic files with TIFF data will confuse the SDSC image library. For you to avoid generating this type of file, we recommend that you configure RenderMan defaults to generate .pic files with PICIO data. PIXAR pic files are recognized by the following filename SDSC Last change March 4, 1991 1 IMPIC(3IM) SDSC IMAGE LIBRARY IMPIC(3IM) suffixes .pic, .picio, and .pixar.

read_pict_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1223
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the pict format into an AVS Network. The PICT file is the standard Apple Macintosh image file format. It can be read by almost any Macintosh graphics application. Most Macintosh applications that can export graphics do so in the PICT file format. pict files are recognized by these filename suffixes .pict and .pict2.

read_pix_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1224
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the pix format into an AVS Network. pix image files are generated by the rendering and painting tools of Alias Research, Inc., such as renderer, raytracer, and paint. See the Alias documentation set for details on how to use these tools. _ΦN_Φo_Φt_Φe Alias quickpaint, available on Silicon Graphics, Inc., IRIS workstations, uses Silicon Graphic's rgb image file format rather than the Alias pix image file format. See the imrgb(3IM) man page for details on the rgb format. Alias pix files are recognized by these following filename suffixes .alias and .pix.

read_plot3d

Author :	**Michael Rangitsch** Dow Chemical
Number :	1854
Language :	C
Submitted :	10/17/94
Version :	1.000
Last Updated :	10/17/94
Ported To :	DEC_alpha HP IBM Sun

This module reads either a formatted or unformatted multiblock PLOT3D data set consisting of the grid and data vector (Q file). No corrections are made for block edge overlap and no co-incident node removal is attempted between blocks. The cells of the ucd structure created are all hexahedra. Version 1.0 of new_plot3d reads in only one of the possible plot3d file types, a multiblock (with or without iblanks) stored in the whole field format. The Q vector is labeled with the default names of the primative variables. A second module similar to the unsupported 'extract cfd values' module is planned to compute derived properties from the primative variables. A crude test is made to determine whether or not a file is formatted. This may fail on some platforms.

read_pnm_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1225
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the pnm format into an AVS Network. pnm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. pnm files are recognized by the filename suffix .pnm.

read_points

Author :	**Penny Rheingans** US EPA Scientific Visualization Center
Number :	1311
Language :	C
Submitted :	08/30/92
Version :	1.000
Last Updated :	08/30/92
Ported To :	DEC Sun HP

Read observation locations and variable values from an ASCII file. From these observations build a 1D irregular (scatter) field with one element for each observation.

read_ras_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1227
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the ras format into an AVS Network. ras image files are used by various Sun Microsystems Inc. tools, such as screendump(1) and screenload(1). See the Sun documentation set for details on how to use these tools. Sun ras files are recognized by any of the following filename suffixes .ras, .scr, .sr, and .sun.

read_rectilinear

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1296
Language :	C
Submitted :	08/04/92
Version :	1.000
Last Updated :	08/04/92
Ported To :	DEC IBM Sun HP

This module reads data in ascii format into an “rectilinear field” data structure. The disk file is structured to contain information about the field at the beginning of the file, followed by the data.

read_rgb_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1228
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the rgb format into an AVS Network. rgb image files are generated by Silicon Graphics, Inc., software such as icut(1) and snapshot(1). See the Silicon Graphics documentation for information on how to use these and other Silicon Graphics programs. Silicon Graphics rgb files are recognized by the following filename suffixes .rgb, .iris, and .sgi.

read_rla_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1229
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the rla format into an AVS Network. rla image files are generated by Wavefront Technologies, Inc., software such as image and tdv. See the Wavefront documentation for information on how to use these and other Wavefront programs. rla is a subset of the newer Wavefront rlb specification. Programs that can read and write rlb files can also read and write rla files. The SDSC image library reader/writer is written to accommodate both types of files. Wavefront rla files are recognized by the following filename suffixes .rla and .rlb.

read_rle_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1230
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the rle format into an AVS Network. rle image files are generated by the tools of Utah's Raster Toolkit. See the Utah documentation set for details on how to use these tools. Utah rle files are only recognized by the following filename suffix .rle

read_rpbm_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1231
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the rpbm format into an AVS Network. rpbm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. rpbm files are recognized by the filename suffix .rpbm.

read_rpgm_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1232
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the rpgm format into an AVS Network. rpgm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. rpgm files are recognized by the filename suffix .rpgm.

read_rpnm_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1233
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the rpnm format into an AVS Network. rpnm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. rpnm files are recognized by the filename suffix .rpnm.

read_rppm_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1234
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the rppm format into an AVS Network. rppm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. rppm files are recognized by the filename suffix .rppm.

read_semper

Author :	**Michel DROZ**
Number :	1339
Language :	C
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	DEC Sun HP

Read Semper accesses an image file format from the SEMPER image processing system. The entire file is read into memory, and sent to the Image output port for further processing by other AVS modules. The Y scan line order is reversed compared to other AVS images, so the image will appear upside down unless reversed with "mirror". The images are generally in "xxx.pic" named file.

read_shak

Author :	**Keith Refson** Department of Earth Sciences, Oxford University
Number :	1176
Language :	C
Submitted :	04/16/92
Version :	3.000
Last Updated :	09/30/93
Ported To :	Sun Kubota

The read shak module and the shak_to_geom filter are crystallography front-ends for AVS. Both read an input file containing a description of the atomic co-ordinates of the structure or molecule to be visualized and output AVS geometry representing a ball-and-stick or space-filling model of it. The read shak module is suited to interactive viewing whereas the shak_to_geom filter is useful for off- line creation of a sequence of .geom files for animation by fast_animate(6). Please note - there are a number of sample data file for this module available via anonymous ftp from avs.ncsc.org, in the directory sample_data/avs_data/read_shak.

read_sunras

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1132
Language :	C
Submitted :	01/29/92
Version :	2.000
Last Updated :	01/29/92
Ported To :	Sun HP

Reads an image stored in the SUN Rasterfile format, placing the results into an AVS image field. Most sun rasterfile formats are supported (the exceptions being RT_FORMAT_TIFF and RT_FORMAT_IFF). Run-length-encoded, BGR and RGB formats are supported. Most depths, as well are supported. The exception is that depth=1 (monochrome) is NOT supported. This module will also decompress files which have been compressed using the UNIX(tm) "compress" utility (by checking for the existance of a ".Z" suffix.

read_synu_image

Author :	**Terry Myerson**
	International AVS Center (NCSC)
Number :	1235
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the synu format into an AVS Network. synu is the image file format output by SDSC's synu (Syn- thetic Universe) portable renderer. synu files are recognized only by the following filename suffix

read_tiff_image

Author :	**Terry Myerson**
	International AVS Center (NCSC)
Number :	1236
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the tiff format into an AVS Network. tiff is a generic Tagged Image File Format developed by Aldus and Microsoft in conjunction with leading scanner and printer manufacturers. tiff files may contain images and miscellaneous other image-related items. Such files may be created and manipulated by a variety of Tagged Image File Format tools. See the Tagged Image File Format documenta- tion for details on how to use these tools. tiff support within the SDSC image library is limited to images of certain depths and storage methods. tiff files are recognized by the following filename suf- fixes .tiff and .tif.

read_tiff_sectio

Author :	**Joe Panico** Dept. of Neuroscience, University of Pennsylvania
Number :	1616
Language :	C
Submitted :	05/01/93
Version :	1.000
Last Updated :	05/01/93
Ported To :	IBM

This module reads single planes (slices) from a multi-plane TIFF file and provides at the output port an AVS image (2-D scalar field). A dial widget allows one to interactively dial through the stack of slices.

read_tiff_vol

Author :	**Joe Panico** Dept. of Neuroscience, University of Pennsylvania
Number :	1617
Language :	C
Submitted :	05/01/93
Version :	1.000
Last Updated :	05/01/93
Ported To :	IBM

This module reads in a multi-plane TIFF file (a volume) and provides at the output port an AVS volume (3-D scalar field).

read_ucd_points

Author :	**Penny Rheingans** US EPA Scientific Visualization Center
Number :	1310
Language :	C
Submitted :	08/30/92
Version :	1.000
Last Updated :	08/30/92
Ported To :	DEC Sun HP

Read observation locations and variable values from an ASCII file. From these observations build a ucd structure field with one node for each observation. The cells of the structure are of type UCD_POINT.

read_uniform

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1075
Language :	C
Submitted :	10/10/91
Version :	2.000
Last Updated :	08/30/92
Ported To :	DEC Sun HP

This module reads data in ascii format into a “uniform field” data structure. The disk file is structured to contain information about the field at the beginning of the file, followed by the data.

read_xbm_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1237
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the xbm format into an AVS Network. xbm bitmap image files are generated by MIT's X Window Sys- tem, version 11 (hereafter referred to as X11). xbm files are generated by the X11 bitmap(1) bitmap editor and used by most X11 tools to define cursors, icons, and other mono- chrome glyphs. See the X11 documentation set for details on how to use tools and subroutines dealing with X11 bitmaps. X11 xbm files are recognized by the following filename suf- fixes .xbm and .bm.

read_xwd_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1238
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module reads an image in the xwd format into an AVS Network. xwd window dump image files are used by the xwd(1) and xwud(1) tools of MIT's X Window System, version 11 (hereafter referred to as X11). See the X11 documentation set for details on how to use these tools. X11 xwd files are recognized by the following filename suf- fixes .xwd and .x11.

river_bed

Author :	**Ken Flurchick** North Carolina Supercomputing Center
Number :	1579
Language :	FORTRAN
Submitted :	02/11/93
Version :	1.000
Last Updated :	02/11/93
Ported To :	IBM Kubota HP

This module is used to construct the outline of a river bed. The length parameter specifies the length of the river in grid cells and coordinates. The river starts at y coordinate 0 and extends downward to y coordinate length at y index length. The river is constructed in the xy plane. Width specifies the width of the river in the x direction in grid cells. Width has nothing to do with the shape of the river. The outline of the river is built from a set of sinusoidal line segments of the form A * Sin (wt) + O, where A - Amplitude w - Omega t - Length O - Offset. The data set that is constructed is filled with 0's initially. This module subroutine can be conected to the "river fill" coroutine to create a simulation of river contaminants.

shaker

Author :	**Janet Jensen**
Number :	1144
Language :	C
Submitted :	03/03/92
Version :	2.000
Last Updated :	05/19/92
Ported To :	DEC IBM Sun Kubota HP

Shaker reads an ascii data file defining a molecule of atoms, and allows the user to modify the vibrational mode and frequency for that molecule. This module includes a command line filter to interface the avs module with gaussian 90 output.

sololoadmd

Author :	**Ken Flurchick and Mark Reed** North Carolina Supercomputing Center
Number :	1604
Language :	C
Submitted :	04/09/93
Version :	1.000
Last Updated :	04/09/93
Ported To :	DEC IBM Kubota

sololoadmd - read velocity and coordinate information for N point objects and represent them as spheres color coded by velocity This module is a modified version of the genloadmd module that is intended for use when memory restrictions don't allow the entire data set to be read into an array. Therefore this modules only reads in one data set at a time and then discards it when a new set is read in. It has several enhancements to facilitate use with a single data set at a time. Note also that the data is expected in a slightly different format than in the genloadmd mdoule. This module reads in position and velocity information for a group of point particles in three dimensions, from a file specified as an input parameter. This parameter can be connected to either the file browser or the animated filename module depending on the user requirements. The program represents each point with a sphere at the position specifed and color coded by the magnitude of the velocity. The colors 0-255 are specified in the colors.h file and range from blue (slowest) to red (fastest). Initially the module will compute the minimum and maximum speeds from the data set read in and echo these back to the parameter widgets. The user can change the size of the spheres and the min and max values used to determine the colors of the sphere. Note that if the minimum speed is greater than or equal to the maximum speed then a message is echoed back to standard output (the screen) and no action is taken. Sphere color is determined by a linear interpolation between the minimum and maximum speeds of the particle velocity magnitude.If the velocity magnitude of a particular sphere exceeds the maximum speed then it is displayed as white while those less than the minimum are displayed as black (note - change the background color if this also is black).

spaceball_master

Author :	**John Stephen** Tessella Support Services
Number :	1717
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP IBM Sun

spaceball master - use small geomviewer window to control views. This module is used with spaceball slaves, to create a small geometry viewer icon window, that can be interactively transformed, then used to cause mirror movements in other geometry viewer scenes. See also - spaceball slave.

spaceball_slave

Author :	**John Stephen** Tessella Support Services
Number :	1718
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP IBM Sun

This module is used with spaceball master, to create a small geometry viewer icon window, that can be interactively transformed, then used to cause mirror movements in other geometry viewer scenes. Individual views can be "snapped" to conform to other views, or cause other views to snap to current one. See also - spaceball slave.

sphere_to_geom

Author :	**Tim O. Robinson** UC Berkeley
Number :	1273
Language :	C
Submitted :	05/27/92
Version :	1.000
Last Updated :	05/27/92
Ported To :	DEC IBM Sun Kubota HP

This module began as a simple wrap around the sphere.c filter but has been extended to include some interesting features. It uses the same ascii file format as /usr/avs/filter/sphere.c, however, the user may also add a list of lines "bonds" to the end of the file and these will be picked up and added to the geometry output. The min and max dials have the effect of 3D cropping on location within the bounds while retaining the boundary box.

string_example

Author :	**Larry Gelberg**
	Advanced Visual Systems
Number :	1553
Language :	C
Submitted :	11/05/92
Version :	1.000
Last Updated :	11/05/92
Ported To :	DEC IBM Kubota HP

The string example module sends a single user-specified filename string to one or more string parameter ports on one or more receiving modules. Its purpose is to allow you to simultaneously control filename parameter input to more than one module using only a single File Browser input widget. This code was based on the AVS "file browser" module.

string_list

Author :	**Phil McDonald**
	NOAA/ERL/Forecast Systems Laboratory
Number :	1279
Language :	C
Submitted :	05/29/92
Version :	1.000
Last Updated :	05/29/92
Ported To :	Kubota

String list is a coroutine module that can be used to input a series of strings from a list, a single string from the list, or a string from the user. If a list of strings is supplied, the strings may be input "automatically" (read as fast as the network will accept them) or stepped through one at a time. The list may be editted by skipping over individual strings. If the user-supplied string contains an asterisk, the string is used as a template – each list string is substituted for the asterisk wildcard to create the actual string that is output.

string_macros

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1685
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

This module provides a string macro facility for annotated choice lists anywhere strings are used in an AVS network. The module reads a configuration macro list file, sets up a choice list of macros, then puts the contents of the macro file at the output port.

strip_chart

Author :	**Larry Gelberg** Advanced Visual Systems
Number :	1115
Language :	C
Submitted :	01/06/92
Version :	1.000
Last Updated :	01/06/92
Ported To :	Kubota DEC IBM Sun HP

STRIP CHART archives (collects) values from some other routine and packages them up into 1D fields appropriate for sending to the graph viewer. There are two internal arrays, each 1024 element long. One is for integer values and the other is for floating point values.
Each array is static and has its own counter associated with it. When the counters pass 1024, they wrap back to 0.

taperer

Author :	**Larry Gelberg** Advanced Visual Systems
Number :	1634
Language :	C
Submitted :	07/05/93
Version :	1.000
Last Updated :	07/05/93
Ported To :	HP IBM Kubota DEC Sun

The taperer module is one of a suite of four AVS modules which can be used to design simple woodworking projects. For more information on the purpose of the project, please read the file called "Poster_Session" and the man pages for each module. The four modules are "cabinet maker" - generates structures made of boxes, "taperer" - generates structures made of tapered boxes, "extruder" - creates capped, extruded tubes from lists of X/Y pairs, and "grid_gen" - creates measured grids so that accurate models can be constructed. Due to the current structure of the IAC's directories, this suite has been split into 4 separate directories on the ftp site. However, a compressed tar archive of all 4 mods is included with each of the directories as wood_mod_suite.tar.Z. Also note that there are several images contributed with this module suite. These have been stored on the ftp site in the directory at avs.ncsc.org called sample_data/avs_data/woodworking_ima ages.

teapot

Author :	**Anonymous**
	Stardent Computer
Number :	1085
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

This module creates a geom version of the famous Newell teapot as described by Jim Blinn. It creates polygons based on subdiving bi-cubic patch definitions and then lets you mangle the teapot in all sorts of creative ways. The 'Offset' knob controls the 'bloat' of the teapot (offsetting the polygons in the direction of their surface normals). The 'Twist' knob controls how far around the selected 'Twist Axis' the teapot will be wrenched by and the 'Mesh Res' knob controls the resolution of the bi-cubic patch division. While this is generally meant to be a fun demonstration module, it is a good road map for a module which turns bi-cubic patches into renderable polytriangle strips.

track_cursor

Author :	**Anonymous**
	Stardent Computer
Number :	1139
Language :	C
Submitted :	02/26/92
Version :	1.000
Last Updated :	02/26/92
Ported To :	Sun HP

Track Cursor uses X Window System calls to follow the cursor around the screen and grab N*M pixels off the screen (with the cursor positioned at the upper left hand corner of the snagged window). It then outputs the (32-bit, true color) window contents as an AVS field 2D 4-vector byte which may be displayed using DISPLAY IMAGE, etc.

transform_xyz

Author :	**Lisa Traub**
	US Geological Survey
Number :	1726
Language :	C
Submitted :	12/14/93
Version :	1.000
Last Updated :	12/14/93
Ported To :	Sun

This module reads an ASCII file consisting of lists of x,y,z points, performs a scale and offset on each dimension and constructs a polyline geometry object using the points in the file as vertices. This program was developed through the combining of Wes Bethel's plot_xyz and Convex's transform_points modules.

tree

Author :	**Larry Gelberg** Stardent Computer
Number :	1104
Language :	C
Submitted :	11/26/91
Version :	1.000
Last Updated :	11/26/91
Ported To :	Kubota DEC IBM Sun HP

TREE generates a 2D binary tree-like geometry using recursive techniques. You can control things like the level of recursion, the average angle between the branches, the average branch length, and random Gaussian deviations from the average. You can also tell the module to generate green "leaves" on the leaf nodes.

trigger_int

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1601
Language :	C
Submitted :	03/23/93
Version :	1.000
Last Updated :	03/23/93
Ported To :	DEC IBM HP

The trigger int module waits for a field to be presented on its input port. When a field (new data) is detected, an integer is stuffed into the output port. The value of the integer is determined by the "start value" parmeter for the first execution. Subsequent invocations cause a static variable to be incremented according to the parameter "increment." This module is useful in creating networks to facilitate animations based upon a single coroutine.

trigger_int_2

Author :	**anonymous**
Number :	1893
Language :	C
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_ultrix DEC_alpha

The trigger int module waits for a field to be presented on its input port. When a field (new data) is detected, an integer is stuffed into the output port. The value of the integer is determined by the "start value" parameter for the first execution. Subsequent invocations cause a static variable to be incremented according to the parameter "increment." If the Reset Value button is pressed, then the output integer is reset to the start value. This module is useful in creating networks to facilitate animations based upon a single coroutine.

ucd_color_ra

Author : **Mike French**
Evans and Sutherland UK
Number : 1758
Language : C
Submitted : 12/15/93
Version : 1.000
Last Updated : 12/15/93
Ported To : Kubota Sun

The range of the UCD scalar node data is used to set the lo/hi values for the colormap. If the UCD min/max value is not available a warning is printed on the terminal output for AVS and the range is calculated from the UCD dataset. The behaviour of COLOR RANGE UCD is designed to match that of COLOR RANGE provided with AVS for use on fields. This module is design to be used with the ISOBAND UCD and ISOLINE UCD contouring modules.

ucd_comp

Author : **Jonathan Cox**
Department of Computer Science, University of Manchester
Number : 1712
Language : C
Submitted : 09/30/93
Version : 1.000
Last Updated : 09/30/93
Ported To : HP

The ucd comp module allows a component of model, cell or node data to be selected from a UCD structure. The selected component is made available on an output port. This output can be used to set the component parameters of other ucd modules via network connections. This allows a single set of radio buttons to control component selection in a number of UCD modules within a network. A component of model, cell or nodal data is to be selected. This is controlled through a choice widget. It is likely that in a network this parameter will be not be modified. Therefore, it is envisioned that the network editor will be used to delete this set of radio buttons in most networks. The module also outputs a suitable title string for the selected component.

uncompress_file

Author :	**Lee J. Silverman**
	Brown University
Number :	1589
Language :	C
Submitted :	02/21/93
Version :	1.000
Last Updated :	02/21/93
Ported To :	DEC IBM

The compress/uncompress module pair is designed to allow datasets or images or other files to be stored and kept in compressed format, but still be read into an AVS network. There must be enough space for this module to uncompress at least one file at a time, or it will bomb out. Generally speaking, a filename will be passed to the uncompress module, either through the browser or via input parameter string. The uncompress module will decompress the file, then pass the filename on to a data-input module such as Read Field.

uv_rect

Author :	**John Tee**
	AVS Inc. Consultant
Number :	1695
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

uv rect - Create a rectangle with longitude and latitude UV coordinates. UV RECT generates a geometry which is a rectangle, with UV coordinates set according from 0,0 at the top left to "U max" "V max" at the bottom right. UV RECT is a demonstration module to show how rectangular portions of scanned map data could be assembled as geometries on to which the map data is texture mapped.

uv_sector

Author :	**John Tee**
	AVS Inc. Consultant
Number :	1696
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

uv_sector - Create a mesh over a sector of an annulus. UV_SECTOR generates a geometry which is a sector of a circular annulus, with UV coordinates¿ The geometry of the mesh is determined by the input parameters. The Resolution parameter determines the number of mesh nodes in both directions. The UV coordinates are set from 0,0 to 1,1 over the extremities of the mesh. The purpose of the module is to demonstrate the ability to warp an image (eg cartographic dataa) by texture mapping it to a geometry whose shape can be varied.

waves_demo

Author :	**Dr. David Faux** Physics Department, University of Surrey
Number :	1191
Language :	FORTRAN
Submitted :	05/05/92
Version :	1.000
Last Updated :	05/05/92
Ported To :	DEC Kubota

This module is used to teach physics to a group of 2nd year students, and illustrates the transmission and reflection of electromagnetic waves from transparent media. The transparency of the media represents the magnitude of their refractive indices, which can be altered. The incident angle may be changed and all other angles, reflection and transmission coefficients, amplitudes of waves, phase changes etc are calculated and diplayed. The waves propagate when the animate button is clicked. Interesting physical effects such as the Brewster angle may be demonstrated.

secno = ount0

FILTER MODULES

International AVS Center

4D_to_3D_Sli

Author :	**Annette Walsh**
Number :	1143
Language :	FORTRAN
Submitted :	03/03/92
Version :	1.000
Last Updated :	03/03/92
Ported To :	IBM HP

Accepts 4D data and extracts a 3D "slice". It is essentially doing the same processing as the "orthogonal slicer" but for 4D data rather than 2D or 3D. The slice plane parameter determines the value of the array index to be held constant. This value is reset to zero each time a new data field is input. The axis parameter selects the dimension (I, J, K or L) in which the array index is to be held constant.

ColorToBW

Author :	**Steve Thorpe**
	International AVS Center
Number :	1599
Language :	C
Submitted :	12/15/92
Version :	1.000
Last Updated :	12/15/92
Ported To :	DEC Kubota HP IBM

This module converts an image from color to black and white, keeping the alpha channel the same in the resulting image.

Color_Legend

	Stardent Computer
Number :	1049
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota

Provides a color map data type for other mapping modules to use, and provides an annotated display of the current map. It makes a discrete map, with 16 entries, where each entry color can be edited individually. The map definition is read from a file (with file browser) including annotation details.

Cylinder

Author :	**D. LE CORRE** CONVEX S.A
Number :	1199
Language :	C
Submitted :	05/08/92
Version :	1.000
Last Updated :	05/08/92
Ported To :	DEC IBM Kubota HP

This module maps a 2D input field along a (portion of a) cylinder. If the input field is UNIFORM, the y dimension gives the number of subdivisons of the circle and the x dimension is the depth of the cylinder. If the input field is rectilinear, the x positions are taken along the cylinder axis, while the y values are positions around the circle, given in degrees. direction give the sphere subdivison.

Delta_Int

Author :	**Stephen H. Price** Loma Linda University Medical Center
Number :	1629
Language :	C
Submitted :	07/02/93
Version :	1.000
Last Updated :	07/02/93
Ported To :	HP IBM Kubota DEC Sun

This module takes in an integer and outputs an integer which is the differnce based on the value of an parameter dial- thus the name Delta_int. The range of the delta is -200 to +200. This module is really useful if you need to input i.e. N and N-1 to two ports down stream. Also useful to animate a crop range i.e. 5 to 10, then 6 to 11, then 7 to 12, etc...

Downsize_2

Author :	**Ian Curington (AVS Inc.), and Mike French (E and S)**
Number :	1746
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

The Downsize module changes the size of the input data set by subsampling the data. It extracts every Nth element of the field along each dimension. This one works with 1D, 2D, 3D, uniform or irregular. It has corrections for non-integral downsize coordinates in Uniform fields, and will work with 1D scatter data. There is a bug in the DOWNSIZE modules supplied by AVS Inc. The problem arises when using DOWNSIZE for uniform fields when the downsize value is not an exact factor of all the (dimension-1) values in X, Y and Z. There should be an empty region for the 'remainder', but the standard version always scales the output to the full extents. The result is that spatial positions within field slices will be slightly wrong. This will be most noticeable when two contour surfaces intersect - the bands will not match exactly. A fixed version of the DOWNSIZE module is supplied with the contouring suite, it's module name is 'Downsize', the upper case initial is used to distinguish it from the standard version.

E01SAF

Author :	**Steve Larkin** Computer Graphics Unit, University of Manchester, UK
Number :	1568
Language :	C
Submitted :	01/28/93
Version :	1.000
Last Updated :	01/28/93

This module provides an interface to the NAG Fortran routines E01SAF and E01SBF. The first routine (E01SAF) generates a 2D surface interpolating a set of scattered data points, using the method of Renka and Cline. The second routine (E01SBF) then evaluates the surface at a given point.

Extract_label

Author :	**Penny Rheingans** US EPA Scientific Visualization Center
Number :	1299
Language :	C
Submitted :	08/26/92
Version :	1.000
Last Updated :	08/26/92
Ported To :	DEC Sun HP

Extracts the label and units for one variable in a UCD structure and concatenates them into a string.

Field_Concat

Author :	**Krzysztof (Chris) S. Nowinski** Institute of Applied Mathematics and Mechanics, Warsaw University
Number :	1704
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	

Field Concat concatenates Out Veclen instances of input scalar field into one vector field of dimensions and type same as the first scalar field (if a field of another dimensions and/or type occurs at the input, the module starts creation of a new vector field as in the case of changing of Out Veclen or resetting). If increment is on, neach new input increments the index parameter filling sunsequent components of the field. The user can control the index parameter manually usinf set button to fill in new components. If all components of the outfield are set, the module spits it downstream.

Field_Formula

Author :	**Ronald E. Stogner** Computer Sciences Corporation
Number :	1545
Language :	C
Submitted :	11/03/92
Version :	1.000
Last Updated :	11/03/92
Ported To :	DEC IBM

The “Field Formula” module accepts a reverse polish notation (RPN) formula, specified by the input string. This formula is then applied to all elements of the field and a field of the same dimension is output with a vector length of one. This field contains the result of the applied formula for all locations in the field. “Variables” are defined by their field labels and are delimited by “[]”’s.

Field_Math

Author :	**Krzysztof (Chris) S. Nowinski** Institute of Applied Mathematics and Mechanics, Warsaw University
Number :	1770
Language :	C
Submitted :	02/03/94
Version :	1.000
Last Updated :	02/03/94
Ported To :	HP Sun

The module does some field mathematics – I am intended to talk about it at the AVS94. The module interprets a short (up to 60 lines) program written in AVSFOL (A Very Simple Field Operation Language). AVSSPL admits two types of variables - simple variables with values being single float numbers; field variables with values being float arrays of dimensions determined by the output field.

GRID_MATH

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1801
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

NAME GRID MATH – create a single gridded representation of statistical functions over multiple time steps. The GRID MATH module takes 4-D input and creates 3-D output based upon the selected statistical function. The effect is to reduce multiple time dimensions to a single one representing the chosen math function. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

GeomXYZtoFld

Author :	**Dominique Le Corre**
	Tethys S.A.
Number :	1618
Language :	C
Submitted :	05/01/93
Version :	1.000
Last Updated :	05/01/93
Ported To :	Kubota HP

This module takes as input a geometry, gets all vertices and generates a 1D 3-space field with the coordinates of these vertices. It can be used for instance as the "sampler" input field to hedgehog,streamlines, particle advector, ucd streamlines (AVS5), scatter dots, etc... Optionally, 2 integer dials (From and To) can be used to limit the output field to a portion of the geometry.

Geom_Dup_n_Mirro

Author :	**Janet Haswell**
	Rutherford Appleton Laboratories
Number :	1657
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

Copy and symtetric mirror geometry data. Given axis symetric geometry, this will duplicate and create mirror copies, and produce one geometric object out of many, performing geometry concatonation. Used "geom_duplicate.c" from Ian Curington (AVS Inc) as a basis to generate this module. Take up to 6 geom input ports, combine into 1 geometry then duplicate and mirror according to flags (can mirror in x and/or y and/or z and all 3). Also allows the user to set transparency level for all objects being mirrored, can also - select wireframe mode, cause AVS to compute the normals, and hide probes (anything with "probe" in the name). Lighting, rendermode, perspective, are controlled by toggles. This was used in to represent a section of a magnetic field, showing the whole from the input of just one quadrant.

Geom_Duplicate

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1335
Language :	C
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	DEC Sun HP

Merge up to 4 seperate geometries into one, with the multiple geom input ports, then replicate the combined object N times with a positional offset between them. Each copy is a unique object in the Geometry Viewer, so may be transformed or have properties edited independently.

Geom_Unique_Name

Author : **Krzysztof (Chris) S. Nowinski**
Institute of Applied Mathematics and Mechanics, Warsaw University
Number : 1706
Language : C
Submitted : 09/30/93
Version : 1.000
Last Updated : 09/30/93
Ported To : HP IBM Kubota Sun

The module passes through any geometric input appending to its name a string uniquely describing each instance. When no additional input is supplied, the string is just the current number (increased by one at each call, when Increment button is on, reset to 0 when Reset button is pressed or alternatively set to the value typed in). Otherwise, the value at input is converted to string and appended to the name of the input object. This allows e.g. to obtain a series of isosurfaces named "isosurface_75.0", "isosurface_75.5", "isosurface_76.0" etc.

Geos_Image

Author : **Terry Myerson**
International AVS Center (NCSC)
Number : 1112
Language : C
Submitted : 12/10/91
Version : 2.000
Last Updated : 12/10/91
Ported To : Kubota DEC IBM Sun HP

Filters images; originally this was intended to be used with satellite imagery. The module was written for artistic coloration of imagery. If the red channel is ever greater than the blue and green channels, individually, then the red and green channels are switched in the input image. To insure that the difference is not to great, red and blue can then be added back into the image as a proportion of the red-green difference. The difference after which color is added is the G-R Threshold parameter. The proportion of this difference which is added to the blue and red channels is the Add Factor parameters.

IDL

Author :	**Joshua Goldstein** Research Systems Incorporated
Number :	1782
Language :	C
Submitted :	02/15/94
Version :	1.000
Last Updated :	02/15/94
Ported To :	IBM Sun

This is one of a suite of seven AVS/IDL modules. The modules are all in the FILTERS subdirectory at the IAC. They are IDL - a UNIX utility that gets IDL and AVS communicating; and six AVS modules that convert data between IDL and AVS - field_to_float_parm, field_to_int_parm, field_to_string_parm, float_parm_to_field, int_parm_to_field, and string_parm_to_field. These modules have only been tested on the RS 6000 and Sun Sparcstation. Binaries have been compiled on the HP 700 but its usability is unknown. Binary executables are stored at the IAC for all three of these platforms. There are also some demo files at the IAC, stored in the file demo.tar. You must be running IDL 3.1 or later in order to use these. No guarantee is made that this version of avsidl will work with future versions of AVS and/or IDL. In fact, I know that avsidl will be completely reworked for for IDL 4.0.

IMAGE_WINDOW

Author :	**Terry Myerson** International AVS Center(NCSC)
Number :	1209
Language :	C
Submitted :	05/15/92
Version :	1.000
Last Updated :	05/15/92
Ported To :	Kubota HP

This module allows the user to specify the size of the image they wish to clip out of the input image, and then lets the uses interactively move a window of that size around the image, in order to crop out the appropriate section. This is very similar to the crop module, and in fact provides the same functionality for images, but makes it alot easier if you know the size of your final image.

Input_Formula

Author :	**Ronald E. Stogner**
	Computer Sciences Corporation
Number :	1546
Language :	C
Submitted :	11/03/92
Version :	1.000
Last Updated :	11/03/92
Ported To :	Sun

The “Input Formula” module accepts a input field and uses labels in the field as variables or “species” in an algrebraic expression. The user can select various parts of the expression from the available widgets and build a formula interactively. The module parses this expression and outputs a RPN sting expected by “Field Formula” when the user selects the “Parse Formula“ widget.

Int_to_string

Author :	**Penny Rheingans**
	US EPA Scientific Visualization Center
Number :	1301
Language :	C
Submitted :	08/26/92
Version :	1.000
Last Updated :	08/26/92
Ported To :	DEC Sun HP

Converts an integer input to its string representation. This is really just the AVS equivalent of sprintf.

Intervals_Field

Author :	**Mike French**
	Evans and Sutherland UK
Number :	1748
Language :	FORTRAN
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota

A set of data values is taken from the range of an input data field. The first output value is the minimum of the field, the last output value is the maximum of the field. If the ’Use Zero Level’ flag is OFF, or there is no zero point within the range of the data, then all intermediate values are taken at equal intervals within the data range. This is the default behaviour. If the ’Use Zero Level’ flag is ON and the data range includes zero, then one data level is explicitly set to zero and the remaining levels are partitioned between the positive and negative regions. This module is design to be used with the ISOBAND SLICE, ISOBAND BRICK, ISOLINE SLICE and ISOLINE BRICK modules.

Intervals_UCD

Author :	**Mike French**
	Evans and Sutherland UK
Number :	1759
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Sun

A set of N equally spaced data values are taken from the range of an input ucd node data. The first output value is the minimum of the field, the last output value is the maximum of the field. Any intermediate values are taken at equal intervals within the data range. This module is design to be used with the ISOBAND UCD and ISOLINE UCD modules. Other methods for selecting the data range could be added, e.g. logarithmic, cumulative frequency percentiles, interactive.

Iterate

Author :	**Annette Walsh**
Number :	1142
Language :	FORTRAN
Submitted :	03/03/92
Version :	1.000
Last Updated :	03/03/92
Ported To :	IBM HP

The first module in a two module pair demonstrating looping in an AVS network. The combination of modules ITERATE and STEPPER performs a very simple iterative convergence calculation. Seed data comes into ITERATE (see Figure 1). If any value in the input field is less than or equal to the parameter "Constant", the process has converged and the input field is placed on the "final result" output port. If the process has not converged, ITERATE sends the input data to STEPPER who decrements each value in the field by some amount. STEPPER sends its result back to ITERATE via an AVS upstream connection. ITERATE, having received data on its upstream input port, uses STEPPER's last result and tests for convergence. Modules connected to the final result output port of ITERATE will only receive data if the convergence criteria is eventually met. NOTE - The upstream connection from STEPPER to ITERATE will be automatically established when the downstream connection is made from ITERATE to STEPPER.

LatLon_UTM

Author :	**Jeff Wang** MCNC, Environmetal Programs Group
Number :	1721
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	

This module allows users to do conversion between UTM, or Universal Transver Mercator, coordinates and the regular Latitude/Longitude coordinates. The Input Data is any of the following - UTM Easting, UTM Northing UTM Zone, Latitude, Longitude The output data is any of the above.

MINMAX_2_3

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1804
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

The MINMAX module outputs floating point min and max values for setting legend color scale endpoints. The values are those selected by the user or the min-max of the input_species within the input_field. Output occurs only on the first invocation and when the values change to prevent continuous updates of the color table associated with the color legend. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

MOVING_AVG

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1806
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

The MOVING_AVG mod is a filter that accepts a field 4D 3-coordinate 1-vector uniform float as input, and outputs the same type of field. The module calculates a moving average of the input field. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

MathAVS

Author :	**Cleo Borac**
Number :	1628
Language :	C
Submitted :	07/02/93
Version :	2.000
Last Updated :	12/14/93
Ported To :	HP IBM Kubota Sun

MathAVS establishes a data transfer communication through MathLink. Having two modes of operation - 'send' mode and 'listen' mode, Mathematica and AVS may exchange data in turn. Note - the International AVS Center does not have Mathematica installed on the platforms listed, so was unable to test out this module completely.

MinMax_2

Author :	**Jeff Wang**
	MCNC, Environmetal Programs Group
Number :	1735
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This Module finds mininum and maxmium of a field. This is one of a suite of modules we used to plot the emission data for the state of North Carolina. This group of modules allows displaying points, lines and polygons over a map of the United States. The module suite is stored as a group in a compressed tar file at the International AVS Center, with some sample networks and datasets included. You may find it easier to access the modules this way. The compressed tar file is approximately 1.2MB in size, and can be found via anonymous ftp to avs.ncsc.org, in /sample_data/avs_data/Wang_GIS_mods s.tar.Z.

Nvect_to_2d

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1323
Language :	FORTRAN
Submitted :	08/31/92
Version :	2.000
Last Updated :	09/30/93
Ported To :	HP

Converts “field 1D uniform real N-vector“ to ”field 2D uniform real scalar“ by making the vector-length into the second dimension

Quantize_Image

Author :	**John Langner**
	Stardent Computer
Number :	1302
Language :	C
Submitted :	07/11/92
Version :	1.000
Last Updated :	07/11/92
Ported To :	IBM Sun HP

Quantize Image converts an image with 24 bits/pixel to 256 or fewer colors. This is done with something resembling the Heckbert mean cut algorithm.

Reset_Legend

Author :	**Jeff Wang** MCNC, Environmetal Programs Group
Number :	1742
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This Module recalculates the xoffset, yoffset and scale of the color legend accoding to the size of the graph. This is one of a suite of modules we used to plot the emission data for the state of North Carolina. This group of modules allows displaying points, lines and polygons over a map of the United States. The module suite is stored as a group in a compressed tar file at the International AVS Center, with some sample networks and datasets included. You may find it easier to access the modules this way. The compressed tar file is approximately 1.2MB in size, and can be found via anonymous ftp to avs.ncsc.org, in /sample_data/avs_data/Wang_GIS_mods s.tar.Z.

SIDE_BY_SIDE

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1197
Language :	C
Submitted :	05/06/92
Version :	1.000
Last Updated :	05/06/92
Ported To :	DEC IBM Sun Kubota HP

Side by side pastes input image 1 and input image 2 together, side by side, to produce the output image. image 1 and image2 must be the same height. The dimension of the output image is (width of image 1 + width of image 2)*(height of image 1). This was used to combine two images for comparison.

Shadow

Author :	**Jeffrey A Thingvold** National Center for Supercomputing Applications
Number :	1182
Language :	C
Submitted :	05/04/92
Version :	1.000
Last Updated :	05/04/92
Ported To :	DEC IBM Sun Kubota HP

The shadow module generates a set of geometric shadows by taking the polygon descriptions of other modules and replacing them with 2D versions projected onto the major axes as coloured polytriangle shadows. The shadow module interprets polytriangles, polyhedra, meshes, but not spheres or lines. The orientation is optionally controlled by three tri-state buttons, where the shadow can be placed at the min or max axis position, or inhibited on a particular axis.

SlowDown

Author : **Steve Thorpe**
International AVS Center
Number : 1723
Language : C
Submitted : 12/02/93
Version : 1.000
Last Updated : 12/02/93
Ported To : Kubota Sun

This Filter AVS Module adds a time delay in the execution of an AVS network. Each time it is executed, it uses up MilliSeconds worth of time before passing the input integer to the output integer. This can be useful to slow down an animation in order to more closely examine its data.

Sobel_operat

Author : **Anonymous**
Number : 1079
Language : C
Submitted : 10/10/91
Version : 1.000
Last Updated : 10/10/91
Ported To : Kubota DEC IBM Sun HP

Uses the Sobel Operator for finding edges in a 2D byte field. Typical usage is for finding edges in images prior to some segmentation operation. The input and output are both field 2d uniform scalar byte, and there are no parameters. There is a sample network described in the documentation accompanying the module.

Sphere

Author : **D. LE CORRE**
CONVEX S.A
Number : 1180
Language : C
Submitted : 04/28/92
Version : 3.000
Last Updated : 12/15/93
Ported To : Sun Kubota

This FILTER module maps a 2D input field around a sphere. If the input field is UNIFORM, the number of elements in each direction give the sphere subdivison. If the input field is RECTILINEAR, the coordinates arrays are taken as longitude and latitude (expressed in degrees). Thus, you will only see part of the sphere if the positions do not span 360 degrees. Optionally, you can specify a 2D floating-point scalar field. This scalar value will be added to the sphere radius at each point in the input field and can be used to enter terrain information. Version 2.0 bug fix 6/93 by M.Wheeler of ULCC. Version 3.0 fixes a bug which caused crashes on Sun workstations, and also it now handles byte fields (4 times more compact).

Stepper

Author : **Annette Walsh**
Number : 1141
Language : FORTRAN
Submitted : 03/03/92
Version : 1.000
Last Updated : 03/03/92
Ported To : IBM HP

Is the second module in a two module pair demonstrating looping in an AVS network. The combination of modules ITERATE and STEPPER performs a very simple iterative convergence calculation. Seed data comes into ITERATE (see Figure 1). A test is made to see if the process has converged. If the process has not converged, ITERATE sends the input data to STEPPER who decrements each value in the field by a percentage ("percent/step"). STEPPER sends its result back to ITERATE via an AVS upstream connection. ITERATE, having received data on its upstream input port, uses STEPPER's last result to again test for convergence. NOTE - The upstream connection from STEPPER to ITERATE will be automatically established when the downstream connection is made from ITERATE to STEPPER.

Toggle_Black_Whi

Author : **Steve Thorpe**
International AVS Center
Number : 1703
Language : C
Submitted : 09/30/93
Version : 1.000
Last Updated : 09/30/93
Ported To : HP Sun

Toggle_Black_White is a straightforward but useful AVS filter module. You can use this to change all black pixels in an image to white, or vica versa, depending on how you set a boolean parameter.

Topo2irreg

Author : **Wright, E.L.**
U.S Geological Survey
Number : 1565
Language : C
Submitted : 01/28/93
Version : 1.000
Last Updated : 01/28/93
Ported To : DEC IBM Kubota HP

This module converts a 2D scalar float 2-coordinate field to an irregular 3-D field by adding Z coordinates. The Z coordinates are the data values of the input field (typically topography) multiplied by the value of the vertical scale factor parameter. The input field may be uniform, rectilinear, or irregular.

UcdColorizeGeom

Author :	**Dominique Le Corre**
	Tethys S.A.
Number :	1627
Language :	C
Submitted :	07/02/93
Version :	1.000
Last Updated :	07/02/93
Ported To :	HP DEC Sun

This module takes as input a geometry, and colors all vertices according to a scalar Ucd component and a colormap. It is similar to the colorize geom module for fields.

UcdXYZtoFld

Author :	**Dominique Le Corre**
	Tethys S.A.
Number :	1619
Language :	C
Submitted :	05/01/93
Version :	1.000
Last Updated :	05/01/93
Ported To :	Kubota HP

This module converts the nodes of a UCD structure to a 1D 3-space (scatter) field. Node data are kept into the output field. Two integer dials (From and To) can be used to limit the output field to a subset of the input UCD.

Unjaggy

Author :	**Steve Thorpe**
	International AVS Center
Number :	1591
Language :	C
Submitted :	02/23/93
Version :	1.000
Last Updated :	02/23/93
Ported To :	DEC IBM Kubota HP

This filter module slightly blurs a color image in order to reduce aliasing effects, which are also affectionately known as “the jaggies”. It uses a 3x3 Bartlett window to filter an image. For each (non-edge) red, green, blue and alpha component of a pixel in the input image, the resulting component in the corresponding pixel in the output image is based on a weighted average.

WINDOW_MATH

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1820
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

WINDOW MATH – create a field that is a function of two input fields. The WINDOW MATH module takes two sets of 2D input data previously sliced on the specified "X", "Y", or "Z" input slice axis 1/2. The output field contains values that represent the chosen "window function" of the two input fields. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

WIND_MINMAX

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1817
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

WIND MINMAX – output either user selected min max values for use with a color legend or the min max values found within the input field of wind data. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

WIND_SPEED

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1819
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

WIND SPEED – create a vector for wind speed from u-v components. The WIND SPEED module accepts as input a 4D field of 2-vector u-v wind components and outputs the same information in a field with an additional vector specifying wind speed, r using the formula r = sqrt(u * u + v * v). The output field labels are set to UWIND, VWIND, WIND SPEED, and the units are all set to "m/s". The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

X_Crop

Author :	**Johan Wiklund**
Number :	1901
Language :	C
Submitted :	11/08/95
Version :	1.000
Last Updated :	11/08/95
Ported To :	DEC_ultrix DEC_alpha

X_Crop - This module is a ImageViewer Interface to crop it draws a rectange to crop and the rectangle's size can be changed and the rect. can be translated

X_ray

Author :	**Larry Gelberg**
	Advanced Visual Systems
Number :	1116
Language :	C
Submitted :	01/07/92
Version :	1.000
Last Updated :	01/07/92
Ported To :	Kubota DEC IBM Sun HP

Does simple, orthogonal volume visualization on 3D uniform byte fields. Outputs a 2D integer field which can be colorized and displayed as an image. Essentially, for a given direction, the module looks at the row of voxels "behind" each screen pixel and, depending of the selected operation, creates a new pixel based on those voxels. The processing options are SUM (each screen pixel is the sum of the stack of voxels), MEAN (each screen pixel is the sum of the stack of voxels divided by the number of voxels in each stack), MEDIAN (the stack of voxels is sorted by value and the screen pixel gets the center value in the sorted stack), MIN (the screen pixel gets the smallest value in the stack), and MAX (the screen pixel gets the largest value in the stack) This is a pretty fast volume visualization technique and you should be able to easily extend it to cover other cases.

Z_slicer

Author :	**Wright, E.L.**
	U.S Geological Survey
Number :	1566
Language :	C
Submitted :	01/28/93
Version :	2.000
Last Updated :	09/19/94
Ported To :	DEC IBM Kubota HP

Z_slicer interpolates a 2D irregular field at a specified Z value from a 3D irregular field with stretched ("sigma") vertical coordinates. This is a common need in 3D atmospheric and oceanic models, where irregular fields have fixed X,Y locations for each vertical index, but Z values vary with the thickness of the fluid layer.

aCNN

Author :	**Stefano Brozzi, Andrea Menozzi, Giulio E. Destri** Universita' degli Studi di Parma, 43100 Parma, Italy
Number :	1872
Language :	C
Submitted :	09/06/95
Version :	1.000
Last Updated :	09/06/95
Ported To :	DEC_ultrix DEC_alpha

The input image is first convolved with the control mask then added to the bias and to the convolution of the internal state and the feedback mask, then a function f is applied to this to get the output image. The output image is also the future feedback in case that more than a iteration is done.

add_2_ints

Author :	**Steve Thorpe** International AVS Center
Number :	1769
Language :	C
Submitted :	02/03/94
Version :	1.000
Last Updated :	02/03/94
Ported To :	HP Kubota Sun

This short AVS module was written to add 2 integers together. When creating an animation using animate_integer to 1) drive orthogonal slicer's slice plane, and 2) same integer to drive an output frame name (as created using Terry Myerson's animate_file), I wanted an offset to be added to the file name's frame number. This offset is provided by a parameter to this module.

add_coords_2

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1004
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Takes as input a 2d uniform field, then adds coordinate information to the uniform field, "changing" it into an irregular field. The notion is that 3-d coordinate information is assigned to each input grid point of a 2d uniform field. The user controls the range of the coordinates generated by changing the values input parameters. There are no restrictions on the size or range of the xmin, xmax values. A trilinear interpolating scheme is used.

add_coords_3

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1005
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Takes as input a 3d uniform field, then adds coordinate information to the uniform field, "changing" it into either an irregular or rectilinear field. The notion is that 3-d coordinate information is assigned to each input grid point. The user controls the range of the coordinates generated by changing the values of the dials, xmin, xmax, etc. There are no restrictions on the size or range of the xmin, xmax values.

add_cps

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1006
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Takes as input an irregular 3d 3-coordinate scalar field, processing it using the catmull-rom spline formulation, and produces a field suitable for consumption by the bezier volume curvilinear volume renderer. The input field consists of a 3d irregular, 3-coordinate floating point scalar field. From the input field, a "voxel" is defined by the eight vertices which surround it. For each "voxel", interior control points are computed which define a tricubic bezier solid. Thus, if N points (on a given data axis) are input, this module will produce 3N-2 output points.

add_extents

Author :	**Steve Larkin** Computer Graphics Unit, University of Manchester, UK
Number :	1602
Language :	C
Submitted :	03/23/93
Version :	1.000
Last Updated :	03/26/93
Ported To :	Sun DEC IBM HP

This module copies the input field to the output field whilst recalculating the extent information for the data and coordinates supplied in the input field. All extra field information is copied to the output field e.g., labels, units, data, coordinates, type The module will accept any type of field, with N dimensions and M data components per data element. This module is useful when modules further up a network produce fields which do not have the extent information added. EXAMPLE NETWORKS

adjust_rgb

Author :	**David Haith** Power-Media
Number :	1007
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Filters image to “tint” or modify color balance. Each component in RGB space can be altered, added or subtracted to adjust for overall effect. This takes the place of many other modules, combining “tint” into one operation.

alpha_mix

Author :	**David Haith** Power-Media
Number :	1008
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Places one image over another and graduates according to the alpha channel contents. Where the alpha channel permits, blends the images with the “mix” control.

and_or_merge

Author :	**A. Jacobson**
	UCLA Dept. Pharmacology / Crump Institute
Number :	1859
Language :	C
Submitted :	10/17/94
Version :	1.000
Last Updated :	10/17/94
Ported To :	DEC_alpha HP IBM Sun

And/or merge uses a variety of boolean operators to merge 2 or 3 images, with 2 or 3D ROI (with 2 source images). This module can be used to replace the use of several instances of field math ip threshold, ip rescale and similar modules. It allows facile merging of datasets with output values segmented to single values for ease of differentiation when displayed. This is actually 4 modules rolled into 1 to save disk and palette space. The routines are all similar and are only a subset of useful merge possibilities.
All data inputs must have the same extents and dimensions. However the routines that take an ROI will take either a 2D ROI, applying the ROI to each "Z" plane if 3D data fields are used, or a 3D ROI, applying each point of the ROI field to the data fields.
The ROI field must have the same dimensions as the data inputs, either in x,y (if 2D ROI) or x,y,z (if 3D). In either case, non-zero values in the ROI field are considered "within" the ROI.
These 4 routines are described in much greater detail in the documentation accompanying the source code.

arith_binary

Author :	**International AVS Center**
	North Carolina Supercomputing Center
Number :	1500
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

arith_binary is an avs module executable which contains 6 modules hacked from khoros. Binary Arithmetic Operations. ** DEC vabsdiff ** computes the absolute difference between two images ** vadd ** adds two images pixel by pixel ** vblend ** forms composite image by adding X*(img1 data) to (1-X)*(img2data) ** vdiv ** divides two images pixel by pixel ** vmul ** multiplies two images pixel by pixel ** vsub ** subtracts two images (img1 = img1-img2) pixel by pixel

arith_logical

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1501
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

arith_logical is an avs module executable which contains 6 modules hacked from khoros. Logical Operations. ** vand ** logically AND two images pixel by pixel ** vlshift ** shifts the pixel data in an image left s bit positions ** vor ** performs a logical OR between two images ** vreplace ** replaces pixels of first image with pixels of second image ** vrshift ** shifts the pixel data in an image right s bit positions ** vxor ** exclusively OR two images pixel by pixel

arith_unary

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1502
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

arith_unary is an avs module executable which contains 13 modules hacked from khoros. Unary Arithmetic Operations. ** DEC vabs ** computes absolute value of image ** vclip ** limits range of values that grey levels may occupy ** vconj ** computes the complex conjugate of the input image ** vexp ** computes the base 10 antilogarithm of an image pixel by pixel ** vfloor ** clips the lower levels of the input image to the specified value ** vinvert ** inverts the gray-levels of a byte type image ** vlog ** DEC compliments image by subtracting pixel values from const. level ** vnormal ** normalizes an image ** vnot ** comliments an image by subtracting pixel values from constant level **voffset ** adds a specified offset to each pixel in the input image ** vscale ** multiplies an image by a constant ** vsqrt ** computes the square root of an image pixel by pixel ** vsubstit ** computes the EROSION of a binary image by a structuring element

arithmetic_2

Author :	**Anonymous**
	Stardent Computer
Number :	1014
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Combines two fields of integer data, and outputs a new composite field. The two input fields are arithmetically combined, using one of several operators, selected by setting one of several choice buttons. Both input fields must be the same size, but need not be the same dimension. The output field is the same size and type as the input. This module is used to compare similar scalar fields, for instance the difference between the red and green channels of an image, subtract two volumes and visualize the difference, or normalize field data by removal of static background field. This module is open ended, many more buttons and types of operators could easily be added.

backgroundRGB

Author :	**John Stephen**
	Tessella Support Services
Number :	1639
Language :	C
Submitted :	09/24/93
Version :	1.000
Last Updated :	09/24/93
Ported To :	HP IBM Sun

backgroundRGB changes the background colour of the current camera as set by 3 red - green - blue dials.

byte_image_t

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1126
Language :	C
Submitted :	01/26/92
Version :	1.000
Last Updated :	01/26/92
Ported To :	Kubota DEC IBM Sun HP

Converts an image contained in a byte field to one contained in a floating point field. This is useful to in processing images in which the algorithms operate on floating point representations of images. Use of the AVS-supplied converters (field to byte, field to float) will produce incorrect results. The range of input pixels is 0 ¡= value ¡= 255.

classify

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1503
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

classify is an avs module executable which contains 9 modules hacked from khoros. Classification/Clustering Algorithms. ** DEC lrfclass ** classifies an image using the Localized Receptive Field (LRF) ** lrftrain ** trains on an image for the weights used with the LRF ** viso2 ** converts image into vectors and performs iso2clustering ** vkmeans ** converts image into vectors and performs kmeans clustering ** vlabel ** performs a labeling in a multiband image or a cluster image ** vmindis ** a simple minimum distance classifier ** vqerr ** computes RMS quantization error with two images ** vquant ** performs N-dimensional vector quantization ** vwmdd ** an Weighted minimum distance detector

closing

Author :	**David E. Harris** Georgia Institute of Technology
Number :	1172
Language :	C
Submitted :	04/16/92
Version :	1.000
Last Updated :	04/16/92
Ported To :	Kubota

This module performs the mathematic morphological closing operation on grayscale (or binary) images. See Haralick et al. "Image Analysis using Mathematical Morphology", IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI-9, No. 4, pp. 532-550, 1987 for a more complete discription. The closing of image A by structuring element B (a binary mask) is defined as the erosion (by B) of the dilation of A by B. This is done by feeding the output of the DILATE module into the input of the ERODE module and using the same mask for both. The CLOSING module just combines these two modules for convenience. See also the OPENING module.

collage

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1020
Language :	C
Submitted :	10/10/91
Version :	2.000
Last Updated :	10/17/94
Ported To :	Kubota DEC IBM Sun HP

Copies two images into an output image. The user can control the position of each of the input images within the output image. The portion of the output image which is not covered by either of the input images is set to the value of the Gray Fill dial. This was originally part of the HIPS image processing distribution. This particular version has been written from scratch to allow for images of varying format. That is, the input images may be grayscale format; single byte, ARGB format; four bytes per pixel, complex format; two bytes per pixel. Version 2 (also by Wes Bethel) provides bug avoidance whereby a percent character in a parm name string caused AVS to dump core.

collapse

Author :	**James Pipe** University of Michigan Medical School
Number :	1607
Language :	FORTRAN
Submitted :	04/10/93
Version :	1.000
Last Updated :	04/10/93
Ported To :	DEC IBM Kubota

Reduces an N-dimensional data set (1 ¡ N ¡ 5) to an N-1 dimensional data set by taking the (min—max—mean—std) of the data across dimension E (the dimension to be eliminated). This reduction can take place across the entire span of dimension E, or across some subset of it (as specified by the user).

collapse_2_num

Author :	**James Pipe** University of Michigan Medical School
Number :	1608
Language :	FORTRAN
Submitted :	04/10/93
Version :	1.000
Last Updated :	04/10/93
Ported To :	IBM

Reduces entire data set to a single floating point number. This number is either the minimum, maximum, mean, sum, standard deviation (std), or count (n) of the input data set. This module will only work for 4-byte real and 4-byte integer input data types.

colorizer_2d

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1587
Language :	C
Submitted :	02/11/93
Version :	1.000
Last Updated :	02/11/93
Ported To :	Sun

Similar to the “colorizer” module in function, this module will take scalar data along with a colormap, and produce color data. Whereas the “colorizer” module takes a single scalar field and a single colormap, “colorizer 2d” takes TWO scalar fields (of the same dimensions) and TWO colormaps. The resulting color vector produced is a function of the two scalar values and the two colormaps.

colour_cones

Author :	**Mark Pinches, Southampton Parallel Applications Centre, UK** and Ian Curington, AVS Inc.
Number :	1640
Language :	C
Submitted :	09/24/93
Version :	1.000
Last Updated :	09/24/93
Ported To :	HP IBM Kubota Sun

The ccone module transforms an AVS geometry, replacing a set of disjoint lines with “cones” constructed out of eight polygons. The cone tristrip is now upto 2x more efficient, and has a subdivision slider control. If the input has values the module will make colour coded cones based on the input lines and colormap. Min and Max controls allow lines of only certain bounded length to have cones, shorter or longer lines do not get cones.

combine

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1207
Language :	C
Submitted :	05/13/92
Version :	1.000
Last Updated :	05/13/92
Ported To :	DEC IBM Kubota HP

This module inputs three scalar arrays, and uses them to determine the coordinates of the output field. No data is copied over into the output field. This module in coordination with delay can produce fractal systems to be visualized with scatter dots.

combine_vecto_2

Author :	**James Pipe** University of Michigan Medical School
Number :	1609
Language :	FORTRAN
Submitted :	04/10/93
Version :	1.000
Last Updated :	04/10/93
Ported To :	IBM

This module is an extension of the supported module 'combine scalars'. This module combines a number of input data sets into a single output data set with one vector component of the output data set corresponding to each vector component of each input data set. Note that all of the input data sets MUST have same dimensions and data types. This is different from 'combine scalars' in the following ways - 1) The input data need not be scalar. For example, a 1-vector, 3-vector, and 2-vector data sets may be combined into a 6-vector output data set. 2) The total vector length of the output data set may be have up to 12 elements. Note there are 12 input ports, so that up to 12 scalar input data sets may be combined. If one or more of the input data sets are an N-vector (N ¿ 1), then not all of the input ports may be used. The limiting factor here is an output data set which is at MOST a 12-vector. 3) You may add labels to each vector component to 'keep track' of the output data's vector elements downstream. 4) The vector length of the output data set is automatically set to be the sum of the vector lengths of all of the input data sets.

combine_vectors

Author :	**Earl W. Hood** Convex Computer Corporation
Number :	1554
Language :	C
Submitted :	11/16/92
Version :	1.000
Last Updated :	11/16/92
Ported To :	IBM

The Combine Vectors modules combines up to 4 vector and/or scalar fields into one vector field. The input ports of the module are processed from right-to-left. The resulting output field will have a vector length equal to the sum of all the vector lengths of the input field. Any labels and units will be carried over to the output field. Written because I got tired of using a bunch of extract_scalar modules when I wanted to process a vector component separately, and then use the combine_scalars module. (eg. process the R,G,B components of an image together, and then combine the R,G,B components back with the Alpha channel).

complex_math

Author :	**Earl W. Hood** Convex Computer Corporation
Number :	1558
Language :	C
Submitted :	11/16/92
Version :	1.000
Last Updated :	11/16/92
Ported To :	IBM

The complex_math module performs binary math operations on 2-vector fields that represent complex numbers. The first vector component represents the real part of the complex number. The second vector component represents the imaginary part of the complex number.

compute_shade

Author :	**Robert Mazaika and Larry Gelberg** Advanced Visual Systems
Number :	1282
Language :	C
Submitted :	06/03/92
Version :	1.000
Last Updated :	06/03/92
Ported To :	Sun Kubota HP

This module combines the functions of the colorizer, compute gradient, and gradient shade modules into a single, memory efficient module. These modules are used primarily to make shaded, ray-traced images. The problem is that they are highly inefficient in terms of memory allocation colorizer takes in a 1 byte per voxel and outputs 4 bytes per voxel compute gradient takes in 1 byte per voxel and outputs 12 (3 floats) gradient shade outputs 4 bytes per voxel These three modules produce 20 bytes for every input data set byte. It is for this reason that some people have experienced problems trying to render ray castings of large data sets - the tracing code is pretty computationally efficient - it's just that most of the system resources go to swapping data, rather than computing the image. The COMPUTE SHADE module does gradient computation, colorizing, and shading on a per slice basis. All in all, it takes less time than running the original three modules in sequence. However, it does take longer than running only gradient shade alone.

concat_str

Author :	**Urs Meyer** Swiss Scientific Computing Center
Number :	1622
Language :	C
Submitted :	07/02/93
Version :	1.000
Last Updated :	07/02/93
Ported To :	Kubota HP IBM Sun

The "concat strings" module will concatenate two or more strings together according to the conversion specification string given. The format string parameter is a character string that controls the concatenation of the string inputs. The conversion specification is the same as for printf.

cone

Author :	**SH Lamson** General Electric Corporate Research and Development
Number :	1021
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun

This module replaces a set of disjoint lines in an AVS geometry with "cones" constructed out of eight polygons. The user can control the radius of the cone's base. Only values in the range 0.0 - 0.4 produce an acceptable result. The user can also use the "threshold" parameter to suppress cones for line segments that are shorter than the threshold. This is useful when ignoring the output of the direction of a vector field in regions where the magnitude is small enough as to make the field "negligible" in some sense.

cone_2

Author :	**Georg Thallinger, Joanneum Research Graz, Austria** and Scott Lamson, General Electri
Number :	1697
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	IBM Sun

The cone module transforms an AVS geometry, replacing a set of lines with "cones" constructed out of eight polygons. The cones are colored based on the colors of the input geomtry. Everytime the input geometry changes, cone tries to find a suitable radius by calculating the average length of a line segment. The min and max values of radius are set to 1/10*r and 10*r (where r is the calculated radius).

contour_perc

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1110
Language :	C
Submitted :	12/05/91
Version :	1.000
Last Updated :	12/05/91
Ported To :	Kubota DEC IBM Sun HP

This module inputs a floating point number and spits out 10%, 20%, 30%, 40%, 50%, 60%,7 0%, 80%, and 90% of that number. It was originally used for contour plots.

convert_color

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1504
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

convert_color is an avs module executable which contains 6 modules hacked from khoros. Color Transformations. ** vcltrans **specifies the linear transformation matricies ** vcluv ** converts to cieucs Luv color spaces ** vcmtrans ** performs linear color space conversions with a 3x3 matrix ** vcrgbhls ** conver6ts from RGB to HLS and vice versa ** vcrgbhsv ** converts from RGB to HSV and vice versa ** vgamut ** takes a 3 band image and generates a pseudo color image

convert_data

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1505
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

convert_data is an avs module executable which contains 10 modules hacked from khoros. Data Format Conversion. ** DEC vband2vect ** reformats data stored in a band type image format ** vbandcomb ** combines all bands from m images into one image ** vbandspt1 ** takes band from multiband image and creates image w/only that band ** vbandspt3 ** extracts 3 data bands from a multiband image ** DEC vctor ** performs simple mappings of COMPLEX data to REAL form ** vect2band ** rearranges the data stored in vector format to band format ** vect2row ** reformats data so that each vector is represented as a row ** DEC vmos2band ** converts mosaic image into image where subimage makes up a band ** vrect2pol ** converts data file from rectangular to polar coord. and vice versa ** vrtoc ** converts REAL data to COMPLEX data

crop2

Author :	**John Tee** AVS Inc. Consultant
Number :	1646
Language :	C
Submitted :	09/24/93
Version :	1.000
Last Updated :	09/24/93
Ported To :	HP IBM Kubota Sun

CROP2 crops 2-D fields by selecting a window of the input field. It differs from CROP in two ways. The more obvious difference is that the output field is defined by the position of a centre pixel, and a field width. The limits of the parameters are interdependent, and so are updated according to the current selections. The other reason for the creation of this module is that CROP limits the minimum dimension to 2. This is unacceptable when a 2-D field is required having one dimension set to 1. (ORTHOGONAL SLICER will produce the correct data, but reduces the dimensionality of the output field, which makes it unacceptable as input to many modules.)

cylinder_2

Author :	**SH Lamson, General Electric C RandD** and Ian Curington, Advanced Visual Systems, Inc
Number :	1714
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP IBM Kubota Sun

cylinder converts lines to cylindrical tubes. It transforms an AVS geometry data type, replacing a set of disjoint lines with “cylinders” constructed out of eight polygons. Unlike “tube”, if the input lines have colors, these are passed through as polygon colors. The number of polygons representing the cylinder is controlled by the resolution slider, from a 2D polygon strip to a prism tube, to square, up to a 20 sided cylinder.

delay

Author :	**Terry Myerson** International AVS Center (NCSC)	Pushes a 1-D array around its bounds, delaying the occurrence of a data value until a later point in the array. This was initially used with a business visualization prototype, trying to find patterns within a linear data set. A sister module named combine was used to combine the linear data sets into a 0-D 3-space field to visualize.uses a 1-D array around its bounds, delaying the occurrence of a data value until a later point in the array.
Number :	1138	
Language :	C	
Submitted :	02/26/92	
Version :	1.000	
Last Updated :	02/26/92	
Ported To :	Kubota DEC IBM Sun HP	

demultiplex

Author :	**John Tee** AVS Inc. Consultant	DEMULTIPLEX converts a line-interleaved scalar field to a vector field. This module is required when importing data which is interleaved on a row-by-row basis from a foreign format file using FILE DESCRIPTOR or DATA DICTIONARY. The constant stride variable provided in the preceding modules does not permit line-interleaved data to be decoded.
Number :	1649	
Language :	C	
Submitted :	09/29/93	
Version :	1.000	
Last Updated :	09/29/93	
Ported To :	HP IBM Kubota Sun	

dilate

Author :	**David E. Harris** Georgia Institute of Technology	"Expands" a grayscale (or binary) image by a structural mask. For binary images, this is an OR operation, for grayscale images it is a neighborhood maximum operator. Dilate will tend to connect islands and fill-in holes.
Number :	1171	
Language :	C	
Submitted :	04/16/92	
Version :	1.000	
Last Updated :	04/16/92	
Ported To :	Kubota	

downsize_points

Author :	**Richard Albrecht** Max-Planck-Institute for Biochemistry
Number :	1724
Language :	C
Submitted :	12/02/93
Version :	1.000
Last Updated :	12/02/93
Ported To :	Kubota Sun

The module divides the max points array by downsizeval. It is assumed that the min points are zero. Now it is possible to use 'downsize' in a correct way with the 'tracer'. 'Tracer' uses the dims-array for adjusting the view-port. In connection with 'downsize', 'tracer' enlarges the output by the downsizevalue (due to the unchanged points-array). We use this mainly for adjusting the viewing angle and the colors of the 'tracer' output at high 'downsize' values (4). The final calculation is done with 1. Our datasource is a confocal microscope and our normal data size is 256*256*100.

draw_pixel

Author :	**Ian Curington** Stardent Computer
Number :	1028
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Draws pixels into an image. Uses upstream transform information to draw pixels over an input image (paint). It is designed to take an input image, and copy it to the output port. This is a straight pass-through. If, however, once and image is passed, it can scribble on top of the image with a square-block brush in light grey. The mouse (brush) position is fed externally from this module, getting the current mouse position and button state from the "struct upstream transform" input port. This is designed to be used with "track_cursor", which displays the image and catches the X events.

dual_control

Author :	**A. Jacobson**
	UCLA Dept. Pharmacology / Crump Institute
Number :	1774
Language :	C
Submitted :	02/09/94
Version :	1.000
Last Updated :	02/09/94
Ported To :	HP Kubota Sun

This module was submitted as one of a four-module suite - 1) dual control - to control flow of field and integers to downstream modules, 2) field control - to control flow of an AVS image to downstream modules, 3) integer control - to control flow of an integer to downstream modules, and 4) real control - to control flow of an real to downstream modules. Dual control will be useful in keeping the flow of data controlable to modules that take integers and fields and fire when either input port (or parameter) is changed. This allows the user the flexibility to update either of the module inputs without the module firing. The only other way to do this with continuing changes would be to connect and disconnect either or both input ports or stop the network flow executive. This module is a combination of the field control and integer control modules. See the documentation for these modules for further details. An example of where this module will be useful is with modules that overwrite files everytime their input is updated. Dual control can be used to only triger such a module as needed.

erode

Author :	**David E. Harris**
	Georgia Institute of Technology
Number :	1170
Language :	C
Submitted :	04/16/92
Version :	1.000
Last Updated :	04/16/92
Ported To :	Kubota

The dual of dilation. It can be thought of as doing to the background what dilate does to the foreground. For binary images, this is an AND operation, for a grayscale image is it a neighborhood minimum operatior. Erode will remove islands that are smaller than the structuring mask.

excavate

Author :	**Anonymous** Stardent Computer
Number :	1031
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Copies a volume field from input to output, while zeroing out all data within a “corner”, as defined by the x,y,z parameters. This allows other mappers to “see” into the volume, to see data values along the interface of the “cut”. Typical uses would be “tracer” or threshold slice.

exponential

Author :	**Earl W. Hood** Convex Computer Corporation
Number :	1559
Language :	C
Submitted :	11/16/92
Version :	1.000
Last Updated :	11/16/92
Ported To :	IBM

The exponential filter module computes 'e' raised to each field value, i.e. newvalue = e$\hat{\text{v}}$alue

extend

Author :	**James Pipe** University of Michigan Medical School
Number :	1611
Language :	FORTRAN
Submitted :	04/10/93
Version :	1.000
Last Updated :	04/10/93
Ported To :	IBM

Turns an N dimensional data set into an N+1 dimensional data set. Does this by copying the N-dimensional input data set m times across the N+1th dimension, where m (the size of the added dimension) is specified by the user.

extrude_field

Author :	**Earl W. Hood** Convex Computer Corporation
Number :	1555
Language :	C
Submitted :	11/16/92
Version :	1.000
Last Updated :	11/16/92
Ported To :	IBM HP

Extrude field generates a 3D volume by applying a transformation matrix defined by the parameters to a 2D field. i.e. it "sweeps" a 2D field through 3-space to generate a 3D (irreguler) field. There were some researchers who had data sets that represented a slice of something axisymetric in 3-space. With extrude_field, one can recreate the 3D representation. This is also good for people who have 2D data sets, but the modules that they would like to use only accept 3D data sets.

feature

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1506
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

feature is an avs module executable which contains 8 modules hacked from khoros. Feature Extraction. ** vfractal ** performs a fractal feature extraction of an image ** vpml ** estimates the fractal dimension of an image ** vpolygon ** creates a vector image resulting of a linear approximation ** vrmatch ** analyzes two input images that are supposed to be similar ** vshape ** performs a shape analysis on two different types of images ** vspatial ** performs a spatial feature extraction on an image ** vstereo ** analyzes image produced by vrmatch and produces image ** vtexture ** extracts texture features from an image

fft2d

Author :	**David E. Harris** Georgia Tech Research Institute
Number :	1202
Language :	C
Submitted :	05/11/92
Version :	1.000
Last Updated :	05/11/92
Ported To :	Kubota

This module takes the forward two dimensional fast Fourier transform of a square array whose dimensions are integer powers of two. As a practical matter, realize that the result is a field with twice the number of values (real and imaginary components) and each is a floating point number. The second output is a grayscale byte field which represents the log(1+magnitude) of the FFT. The log(1+magnitude) plot shows more detail than a plot of just the magnitude. The companion module to this one is the inverse FFT module ifft2D.

field_2D_to_3D

Author :	**Chris Allen** Unisys Corporation
Number :	1292
Language :	C
Submitted :	07/11/92
Version :	1.000
Last Updated :	07/11/92
Ported To :	IBM Sun HP

Field 2D to 3D converts its input field from 2 to 3 dimensions, and sets the physical-space dimensions to 3-space. The data remains unchanged. This module is useful, for example, if you have a 3-vector 2D field that you would like to view with Hedgehog, which requires field 3D uniform data.

field_averaging

Author :	**Earl W. Hood** Convex Computer Corporation
Number :	1556
Language :	C
Submitted :	11/16/92
Version :	1.000
Last Updated :	11/16/92
Ported To :	IBM

The Field Averaging module computes the average field of a series of multiple fields. The module is useful if there exists multiple fields (eg. images) that represent a series of samplings of a given object, and an average of the samples will help cancel out the noisy data contained in each sample.

field_control

Author :	**A. Jacobson**
	UCLA Dept. Pharmacology / Crump Institute
Number :	1775
Language :	C
Submitted :	02/09/94
Version :	1.000
Last Updated :	02/09/94
Ported To :	HP Kubota Sun

This module was submitted as one of a four-module suite - 1) dual control - to control flow of field and integers to downstream modules, 2) field control - to control flow of an AVS image to downstream modules, 3) integer control - to control flow of an integer to downstream modules, and 4) real control - to control flow of an real to downstream modules. Field control will be useful in keeping the flow of data controllable to bottleneck points without requiring the user to repeatedly connect and disconnect modules to control their execution. This module allows the user to control the flow of an AVS field to downstream modules. Often there will be modules, (such as tracer), macros or whole branches of a network that take a long time to execute and may be undesireable in any given run of data through the network. Without this module, the only way to activate or deactivate such a module or branch would be to connect or disconnect it's input, suspend the flow executive of the whole network, or to deactivate it through its editor dialog. Another example of where this module will be useful is with modules that overwrite files everytime their input is updated. Field control can be used to only triger such a module as needed.

field_conver

Author :	**Anonymous**
	Convex Computer
Number :	1149
Language :	C
Submitted :	03/10/92
Version :	1.000
Last Updated :	03/10/92
Ported To :	Kubota DEC IBM Sun HP

Converts an input uniform field into a rectilinear or irregular field, based on the Field Type parameter. Non-uniform fields are passed through unchanged. The output field has the same dimensionality as the input field, but is non-uniform. This module has no effect on non-uniform input fields.

field_label

Author :	**anonymous**
Number :	1890
Language :	C
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_ultrix DEC_alpha

This module allows the user to modify the labels of a field's components. When the field is initially connected, the label parameters will be set to the original values in the data field. The the user should set the 'Apply typed in labels' choice and modify the labels to their new values. If the 'Apply typed in labels' parameter is not set, then each time the field is updated upstream of the field_label module, any typed in labels will be erased and replaced with the original field values.

field_mux

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1747
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

Given several (different size or types of) input fields, one is passed to the output port, given the choice selection. All the data from the input is passed to the output. All 10 input ports are optional, it will work with any subset, or just one input. Inputs can be different dimension, different data types, etc. By default, the names of the input ports on the choice list are "One, Two..." but a string input port can be used to set the choice names from an external procedure. This is an efficient module, as it only passes a pointer, and does not copy data. ADVANCED TECHNIQUE - This routine contains an advanced module writer trick. It does not make a copy of the data. It simply passes a pointer across to form the correct multiplex connection.

field_scatte

Author :	**Michael Rangitsch**
Number :	1147
Language :	C
Submitted :	03/10/92
Version :	1.000
Last Updated :	03/10/92
Ported To :	Kubota DEC IBM Sun HP

Converts a 3 dimensional field into a 1 dimensional field and uses the mapping information of the original field to compute the 3d mapping space. This works for any type of mapping in the original field (uniform, rectilinear or irregular). This module is useful in converting 3d data into a form useable by a 'bubble' type of visualization module.

field_to_float_parm

Author :	**Joshua Goldstein** Research Systems Incorporated
Number :	1784
Language :	C
Submitted :	02/15/94
Version :	1.000
Last Updated :	02/15/94
Ported To :	IBM Sun

This is one of a suite of seven AVS/IDL modules. The modules are all in the FILTERS subdirectory at the IAC. They are IDL - a UNIX utility that gets IDL and AVS communicating; and six AVS modules that convert data between IDL and AVS - field_to_int_parm, field_to_string_parm, float_parm_to_field, int_parm_to_field, and string_parm_to_field. These modules have only been tested on the RS 6000 and Sun Sparcstation. Binaries have been compiled on the HP 700 but its usability is unknown. Binary executables are stored at the IAC for all three of these platforms. There are also some demo files at the IAC, stored in the file demo.tar. You must be running IDL 3.1 or later in order to use these. No guarantee is made that this version of avsidl will work with future versions of AVS and/or IDL. In fact, I know that avsidl will be completely reworked for for IDL 4.0.

field_to_glo_2

Author :	**Raymond E. Flanery Jr., Director** Advanced Visualization Research Center, ORNL
Number :	1902
Language :	C
Submitted :	11/09/95
Version :	1.000
Last Updated :	11/09/95
Ported To :	DEC_ultrix Sun DEC_alpha

This module is used to wrap field data around a globe. It does a pretty good job of allowing the user to specify where on the globe to put the corners of the input field or to use the points data of an input irregular field as the longitude and latitude of each point. If the data is 3-D, then concentric globes are generated with the radius determined by the "radius min" and "radius max" parameters, unless the data is 3-space and the "use height data" toggle is on. Scalar fields can also "use height data" as an elevation map, and 2-vector or 3-vector fields can have their data converted from polar-coordinate direction vectors to cartesian vectors.

field_to_globe

Author :	**Ed Bender** Convex Computer Corporation
Number :	1595
Language :	C
Submitted :	03/04/93
Version :	3.000
Last Updated :	04/21/94
Ported To :	Sun IBM HP

This module is used to wrap field data around a globe. It does a pretty good job of allowing the user to specify where on the globe to put the corners of the input field or to use the points data of an input irregular field as the longitude and latitude of each point.
If the data is 3-D, then concentric globes are generated with the radius determined by the "radius min" and "radius max" parameters, unless the data is 3-space and the "use height data" toggle is on. Version 2 enhancements were made to this module by Scott Ferguson of Cray Computer Corporation, Colorado Springs, CO - Inner loops were restructured to allow vectorization. Also, for scalar input fields, the scalar value can be used as elevation data. And, for 2-vector or 3-vector input fields, the vector data can be converted from polar to cartesian coordinates as well. For example, a model of wind vectors in a global climate model needed to be overlayed on a globe. By selecting "Convert Data", the data can later be passed into hedgehog to show the arrows wrapped around a sphere. A parameter called "Elev Scale" was added to scale the elevation data when using a scalar input field as an elevation map.
Version 3 enhancements were made by Dr. Joachim Biercamp of the German Climate Computer Center. The module now updates the extent field, so downstream modules like hedgehog or sampler don't get confused

field_to_int_parm

Author :	**Joshua Goldstein** Research Systems Incorporated
Number :	1785
Language :	C
Submitted :	02/15/94
Version :	1.000
Last Updated :	02/15/94
Ported To :	IBM Sun

This is one of a suite of seven AVS/IDL modules. The modules are all in the FILTERS subdirectory at the IAC. They are IDL - a UNIX utility that gets IDL and AVS communicating; and six AVS modules that convert data between IDL and AVS - field_to_float_parm, field_to_int_parm, field_to_string_parm, float_parm_to_field, int_parm_to_field, and string_parm_to_field. These modules have only been tested on the RS 6000 and Sun Sparcstation. Binaries have been compiled on the HP 700 but its usability is unknown. Binary executables are stored at the IAC for all three of these platforms. There are also some demo files at the IAC, stored in the file demo.tar. You must be running IDL 3.1 or later in order to use these. No guarantee is made that this version of avsidl will work with future versions of AVS and/or IDL. In fact, I know that avsidl will be completely reworked for for IDL 4.0.

field_to_irr

Author :	**Kathleen Dyer** Lawrence Livermore National Laboratory
Number :	1118
Language :	C
Submitted :	01/22/92
Version :	2.000
Last Updated :	03/04/93
Ported To :	DEC IBM Sun HP

Converts fields to irregular fields by adding grid or coordinate information. This is useful for modules which will not process uniform or rectilinear fields. The input must be a 1D, 2D or 3D uniform field, but it may have any data type at each field point. The output is a field with the same data, shape and dimensionality as the original input field. It is irregular. Version 2.0 update (by Karin Schaefer) allows this module to convert uniform or rectilinear fields to irregular fields by adding a grid.

field_to_rec

Author :	**Kathleen Dyer**
	Lawrence Livermore
	National Laboratory
Number :	1119
Language :	C
Submitted :	01/22/92
Version :	1.000
Last Updated :	01/22/92
Ported To :	DEC IBM Sun HP

Converts fields to rectilinear fields by adding grid or coordinate information. This is useful for modules which will not process uniform or irregular fields. The input must be a 1D, 2D or 3D uniform field, but it may have any data type at each field point. The output is a field with the same data, shape and dimensionality as the original input field. It is rectilinear.

field_to_string_parm

Author :	**Joshua Goldstein**
	Research Systems
	Incorporated
Number :	1783
Language :	C
Submitted :	02/15/94
Version :	1.000
Last Updated :	02/15/94
Ported To :	IBM Sun

This is one of a suite of seven AVS/IDL modules. The modules are all in the FILTERS subdirectory at the IAC. They are IDL - a UNIX utility that gets IDL and AVS communicating; and six AVS modules that convert data between IDL and AVS - field_to_float_parm, field_to_int_parm, field_to_string_parm, float_parm_to_field, int_parm_to_field, and string_parm_to_field. These modules have only been tested on the RS 6000 and Sun Sparcstation. Binaries have been compiled on the HP 700 but its usability is unknown. Binary executables are stored at the IAC for all three of these platforms. There are also some demo files at the IAC, stored in the file demo.tar. You must be running IDL 3.1 or later in order to use these. No guarantee is made that this version of avsidl will work with future versions of AVS and/or IDL. In fact, I know that avsidl will be completely reworked for for IDL 4.0.

field_to_uni

Author :	**Kathleen Dyer** Lawrence Livermore National Laboratory
Number :	1120
Language :	C
Submitted :	01/22/92
Version :	1.000
Last Updated :	01/22/92
Ported To :	DEC IBM Sun HP

Converts fields to uniform fields by the simple expedient of stripping off their grid information. The input field can be any size, any data, any type. The output is a uniform field of the same type.

fill

Author :	**Anonymous** Stardent Computer
Number :	1033
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Does an area fill on a 2D scalar byte field. The X and Y parameters specify the starting position of the fill, the MIN and MAX parameters specify the range of data which will get filled in and the NEW parameter specifies the data value which will get substituted. For instance, to set all connected pixels in an image which lie in the range 200 to 255 to black, MIN = 200, MAX = 255, NEW = 0, and the the seed position must be set to somewhere in the interior of the region which you wish to fill.

filter_1Dfreq

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1507
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

filter_1Dfreq is an avs module executable which contains 5 modules hacked from khoros. 1D Frequency Domain Filters. ** dfiltbp ** a bandpass filter design function ** dfiltbs ** a bandstop filter design program ** dfilthp ** a high pass filter design function ** dfiltlp ** a low pass IIR filter design function ** dzresp ** calculates magnitude, power or phase response of a discrete sys.

filter_1Dtime

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1508
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

filter_1Dtime is an avs module executable which contains 7 modules hacked from khoros. 1D Time Domain Filters. ** dfilconv ** converts 1D filter coeff from transversal to lattice realization ** dfilter ** filters time domain signals ** dllatt ** generates an error, a signal estimate, and a PARCOR sequence ** dlsynth ** synthesizes an output sequence ** dltapp ** generates an error, a signal estimate, and a weight sequence ** dmlatt ** generates an error, a signal estimate, and a PARCOR sequence ** dmtapp ** generates an error, a signal estimate, and a weight sequence

filter_2Dfreq

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1509
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

filter_2Dfreq is an avs module executable which contains 8 modules hacked from khoros. 2D Frequency Domain Filters. ** vblse ** performs band-limited surface extrapolation or apodization ** vbpf ** performs a band-pass filtering operation ** vbrf ** performs a band-reject filtering operation ** vhpf ** performs a high-pass filtering operation on an image ** vinverse ** restores images in the spatial domain by inverse filtering ** vlpf ** performs a low-pass filtering operation on an image ** vsvpsf ** performs restoration of space-varying blurs ** vwiener ** restores images in the spatial domain by Wiener filtering

filter_2Dspatl

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1510
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

filter_2Dspatl is an avs module executable which contains9 modules hacked from khoros. 2D Spatial Domain Filters. ** DEC vconvolve ** computes two-dimensional spatial convolution of two images ** vdiff ** performs image differentiation by running 2 NxN weighted operations ** vdrf ** optimal filter for edge detection ** vgef ** first derivative operator for symmetric exponential filter ** vgrad ** differentiates an image to perform image sharpening ** vhmed ** computes a two dimensional median filter ** vqmed ** computes a two dimensional median filter ** vsdef ** second derivative operator for symmetric exponential filter ** vspeckle ** reduces the speckle index of an image

filter_morph

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1511
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

filter_morph is an avs module executable which contains 10 modules hacked from khoros. Morphological Operations. ** vm180 ** computes a 180 degree rotation of a morphological kernel ** vmcdilate ** computes the dilation of a binary image by a structuring element ** vmclose ** computes the closing of an image by a structuring element B ** vmcustom ** creates a new morphological kernel via keyboard entry ** vmdilate ** computes the dilation of a binary image by a structuring element ** vmerode ** computes the erosion of a binary image by a structuring element ** vmopen ** computes the opening of an image by a structuring element B ** vmoredge ** computes an edge extraction in an xvimage structure ** vmskel ** computes the skeleton of a binary byte image ** vmspan **

fix_field_edge

Author :	**Anonymous** Please forward avs@ncsc.org with author info if you have it
Number :	1613
Language :	C
Submitted :	04/10/93
Version :	1.000
Last Updated :	04/10/93
Ported To :	DEC IBM

For each cell that in a field (2D 2-space 1-vector rectilinear integer) that has a zero value, this module searches for the closest neighbor that has a positive value and replaces the zero value with that one.

float_image

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1161
Language :	C
Submitted :	01/26/92
Version :	1.000
Last Updated :	01/26/92
Ported To :	Kubota DEC IBM Sun HP

Converts an image contained in a floating point field to one contained in a byte field. This is useful to in processing images in which the algorithms operate on floating point representations of images. Use of the AVS-supplied converters (field to byte, field to float) will produce incorrect results. The range of input pixels is 0 ¡= value ¡= 255.

float_parm_to_field

Author :	**Joshua Goldstein** Research Systems Incorporated
Number :	1786
Language :	C
Submitted :	02/15/94
Version :	1.000
Last Updated :	02/15/94
Ported To :	IBM Sun

This is one of a suite of seven AVS/IDL modules. The modules are all in the FILTERS subdirectory at the IAC. They are IDL - a UNIX utility that gets IDL and AVS communicating; and six AVS modules that convert data between IDL and AVS - field_to_float_parm, field_to_int_parm, field_to_string_parm, float_parm_to_field, int_parm_to_field, and string_parm_to_field. These modules have only been tested on the RS 6000 and Sun Sparcstation. Binaries have been compiled on the HP 700 but its usability is unknown. Binary executables are stored at the IAC for all three of these platforms. There are also some demo files at the IAC, stored in the file demo.tar. You must be running IDL 3.1 or later in order to use these. No guarantee is made that this version of avsidl will work with future versions of AVS and/or IDL. In fact, I know that avsidl will be completely reworked for for IDL 4.0.

float_to_str

Author :	**Urs Meyer** Swiss Scientific Computing Center
Number :	1623
Language :	C
Submitted :	07/02/93
Version :	1.000
Last Updated :	07/02/93
Ported To :	HP IBM Kubota DEC Sun

The “float to string” module will convert a floating point number to a string according to the conversion specification given. The conversion specification string is given via a parameter, and it is the same as for printf.

fourier_filter

Author :	**Earl W. Hood and Allen Schiano** Convex Computer Corp. and UC Irvine
Number :	1548
Language :	C
Submitted :	11/04/92
Version :	1.000
Last Updated :	11/04/92

The FOURIER FILTER module takes a 1-3D input field and filters the field in the frequency-domain and outputs the processed field. The module is specifically designed for image and volume enhancement.

fourier_transfor

Author :	**Earl W. Hood and Allen Schiano** Convex Computer Corp. and UC Irvine
Number :	1549
Language :	C
Submitted :	11/04/92
Version :	1.000
Last Updated :	11/04/92

FOURIER TRANSFORM computes the Fourier transform of a field. If the input field is scalar, the modules assumes that the imaginary component of the Fourier transform is zero. If the field is 2-vector, the first component is the real part, and the second component is the imaginary part. In either case, the output field is 2-vector with the real part in the first component, and the imaginary part in the second component of the vector. The module uses Fast Fourier Transform algorithms to compute the desired Fourier transform. Therefore, the output field might have certain dimensions expanded as a result of the algorithms.

gamma_correct

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1550
Language :	C
Submitted :	11/04/92
Version :	1.000
Last Updated :	11/04/92
Ported To :	DEC IBM Kubota HP

The gamma correct module performs gamma correction on an AVS image. The user may control the amount of correction to apply. The user may also choose how correct the image, either in a manner which will result in a hue shift in the image, or so as to preserve hues.

geo_control

Author :	**Massimiliano Bavelloni** Radioastronomy department of C.N.R. Bologna Italy.
Number :	1884
Language :	C
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_ultrix DEC_alpha

This module is useful to create simple animation of any geometry object, provides simple rotation around selected axis, invert rotation sense, scale and translate geometry objects inside geometry viewer scene.

geomanip

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1512
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

geomanip is an avs module executable which contains 7 modules hacked from khoros. Geometric Manipulations. ** vexpand ** expands the size of the image pointed ** vflip ** performs reflection or "flip" operations on an image ** vresize ** magnifies or reduces an image by using billinear interpolation ** vrotate ** rotates an image by a specfied angle, about a specified point ** vshrink ** decreases the size of an image ** vtranslat ** computes the translation of an xvimage structure by vector (i,j) ** vtranspos ** computes the transpose of the rows and columns in an image

geometrySearch

Author :	**John Stephen** Tessella Support Services
Number :	1658
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

This module produces a text string list of items currently in a particular geometry viewer scene. The viewer and name search string are matched, and a full multi-line text string is sent to the output port.

glow_white

Author :	**David Haith**
	Power-Media
Number :	1039
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Blurs white to look like glowing light from input image to output. Controls specify threshold level for glow, and the mix level for replacement. Both are integer parameters.

gradual_mix

Author :	**David Haith**
	Power-Media
Number :	1040
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Places one image over another with a graduated fade from left to right. The degree of mix on each side of the image is given individual control. Both images must be the same size. Four parameters come with the module, all integer type - “left min”, “right min”, “left mix”, and “right mix”.

grey_scales

Author :	**Rob Kuyper**
	Univeristy of Technology
	Delft, The Netherlands
Number :	1315
Language :	C
Submitted :	08/30/92
Version :	1.000
Last Updated :	08/30/92
Ported To :	DEC Sun HP

The module takes an image as input an produces a black and white version of the same image as output. The number of grey scales is adjustable by a dial.
The grey level is produced in the same way as with the command xpr to print windows on a laserjet. The advantage of adjusting the number of grey levels is the fact that the produced image shows the edges between different colors better. The output can be printed on a laserjet which results in a much better picture then without this module. In order to have white as one of the grey levels, the roundoff works like a 'floor'.

grid_scale

Author :	**Kathleen Dyer** Lawrence Livermore National Laboratory
Number :	1122
Language :	C
Submitted :	01/22/92
Version :	1.000
Last Updated :	01/22/92
Ported To :	Kubota DEC IBM Sun HP

Allows the grids of 2D and 3D fields to be scaled. The I, J, and K grids may be scaled independently. There is no effect on a uniform field. This module simply multiplies the various grid values by the appropriate scale factor(s). Fields which are not entered at the origin may appear to shift.

histogram

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1513
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

histogram is an avs modules executable which contains 7 modules hacked from khoros. Histogram. ** venhance ** creates a new image g(x,y) from input image f(x,y) ** venhpatch ** ** vheq ** performs histogram equalization ** vhisto ** computes the histogram of a gray-levelimage ** vhsee ** outputs a histogram in a VIFF format in displayable form ** vhstr ** performs a histogram stretch of the grey levels ** vhxray ** enhances an image by using histogram stretch or equalization

hit_lumin_mi

Author :	**David Haith** Power-Media
Number :	1043
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Mixes according to pixel luminance. The left input is foreground , the right input background. Both images must be the same size. The mix parameter controls effect, blend is on pixel luminance of foreground image. This module is a variation taken from lumin_mix, where luminance is (green + blue *2) /3.0, instead of red, green, blue.

hsv_to_rgb

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1127
Language :	C
Submitted :	01/26/92
Version :	1.000
Last Updated :	01/26/92
Ported To :	Kubota HP

Converts an image, represented as HSVA (A = alpha) to ARGB (A = alpha). You might use this module in a network with read_image, field_to_float, rgb_to_hsv, extract_scalar, field_math, combine scalars, field_to_byte, and colorizer to attenuate the V component (luminosity), then recombine back into HSV space, and display the resulting image. The result is a "darker" image. See the .txt file accompanying the module for this example network.

image_combine

Author :	**Hideo Miyachi** Kubota Corporation
Number :	1615
Language :	C
Submitted :	05/01/93
Version :	1.000
Last Updated :	05/01/93
Ported To :	HP

image_combine generates a image from two image datas. This module has 3 input ports - two images and a colormap. Colormap is the optional port which is for background color if the two images have diffrent size. This is similar to "collage" but this is very simple. I used this module when I generate VTR tape to compare two visualized images showing time dependent data.

image_flip

Author :	**Bill Sherman** National Center for Supercomputing Applications
Number :	1192
Language :	C
Submitted :	05/05/92
Version :	1.000
Last Updated :	05/05/92
Ported To :	DEC IBM Sun Kubota HP

The "image flip" module take an image data field and duplicates the information by flipping it over the selected edge. This differs from the "mirror" module that is provided in the standard AVS library. The "mirror" module flips an image, but does not duplicate the data, so the original image is not part of the output. "Image flip" keeps the original image, placing the new mirrored data adjacent to the edge flipped over.

image_rotate

Author :	**William Hunter** GEC Ferranti DSL, Edinborough, Scotland
Number :	1184
Language :	C
Submitted :	05/04/92
Version :	2.000
Last Updated :	09/30/93
Ported To :	IBM Sun Kubota HP

This module is similar to mirror and transpose, but combines operations to provide image rotation in 90 degree steps, controlled by choice buttons. The source makes extensive use of macros to allow for any storage type, such as byte, float, integer, or doubles. This is contributed "AS-IS", and implies no liability of the author.

image_stack

Author :	**Scott Goodyear and Terry Myerson**
Number :	1835
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP IBM Kubota Sun

The "image_stack" module pastes input image 1 and input image 2 together, top to bottom, to produce the output image. Image 1 and Image2 must be the same width. The dimension of the output image is (height of image 1 + height of image 2)*(width of image 1). This was used to combine two images for comparison, or to create a composite mosaic for an aerial photograph GIS application. Derived from Terry Myerson's side by side module.

image_stats

Author :	**John Tee** AVS Inc. Consultant
Number :	1662
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

Computes field statistics, but in a multiband image context. Statistical measures are applied to each channel in the image, and displayed in an information panel. It computes min, max, mean, and standard deviation. The operator may apply to the whole image, or the ROI region, using the optional mask port, connected to the AVS5 image viewer.

image_tile

Author :	**Ian Curington** Advanced Visual Systems
Number :	1330
Language :	C
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	DEC Sun HP

Creates a replicated (tiled) version of the input image. It uses a simple horizontal pixel duplication, to make a small postage stamp image fill a box. The input image height and width is reported. This module was used in a demo to a design studio that manufactures ceramic tiles for kitchens and bathrooms. Maybe you know someone in wallpaper or fabric? It is also used to synthesize a Y dimension, to create an image from 1D data.

info

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1514
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

info is an avs module executable which contains 6 modules hacked from khoros. File Information. ** dpeakpick ** finds peaks or troughs in 1D data vectors ** dstats ** computes 1-D statistics of an input field ** varviff ** takes as input a field and extracts a single data value ** vfileinfo ** prints out the header information that describes input field ** vprdata ** prints field to the screen or to a file in ASCII format ** vstats ** performs statistical calculations

input_create1D

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1515
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

input_create1D is an avs module executable which contains9 modules hacked from khoros. Create Signals (1D). ** dgexpon ** creates one dimensional exponential distributed noise data set ** dggauss ** creates a one dimensional gaussian noise data set ** dgpoiss ** creates a one dimensional Poisson distributed noise data set **dgpwl ** creates a one dimensional set of signals **dgrleigh ** creates a one dimensional Rayleigh distributed noise data set **dgsin ** creates a one dimensional ensemble of signals **dgsinc ** creates a one dimensional ensemble of signals **dgunif ** creates a one dimensional uniform noise data set **dgwindow ** creates 1D data sequence based on user selected window function

input_create2D

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1516
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

input_create2D is an avs module executable which contains12 modules hacked from khoros. Create Images (2D). ** vcustom ** creates a custom kernel file in the VIFF format ** vgauss ** generates image that contains one or more Gaussian distributions ** vgbox ** fills buffer with greylevel rep. of parallelogram on CCD sensor ** vgcirc ** creates an image of a filled circle ** vgconst ** creates an image of dimension rows bands ** vgfractal ** creates a fractal image with a desired fractal dimension ** vggauss ** creates a two dimensional multi-band Gaussian (noise) image ** vgpwl ** creates a two dimensional piecewise linear image ** vgshot ** corrupts an image uniformly with spikes ** vgsin ** creates a two dimensional image of dimension rows*columns ** vimpulse ** creates image containing 1 or more unit impulses ** vmarr ** creates a convolution filter kernel

int_2_st

Author :	**Urs Meyer** Swiss Scientific Computing Center
Number :	1624
Language :	C
Submitted :	07/02/93
Version :	1.000
Last Updated :	07/02/93
Ported To :	HP IBM Kubota DEC Sun

The int to string module will convert an integer to a string according to the conversion specification given. It is very useful and flexible for generating series of filenames. The format parameter widget is a character string that controls the conversion of the integer value. The conversion specification string is the same as for printf.

int_parm_to_field

Author :	**Joshua Goldstein** Research Systems Incorporated
Number :	1787
Language :	C
Submitted :	02/15/94
Version :	1.000
Last Updated :	02/15/94
Ported To :	IBM Sun

This is one of a suite of seven AVS/IDL modules. The modules are all in the FILTERS subdirectory at the IAC. They are IDL - a UNIX utility that gets IDL and AVS communicating; and six AVS modules that convert data between IDL and AVS - field_to_float_parm, field_to_int_parm, field_to_string_parm, float_parm_to_field, int_parm_to_field, and string_parm_to_field. These modules have only been tested on the RS 6000 and Sun Sparcstation. Binaries have been compiled on the HP 700 but its usability is unknown. Binary executables are stored at the IAC for all three of these platforms. There are also some demo files at the IAC, stored in the file demo.tar. You must be running IDL 3.1 or later in order to use these. No guarantee is made that this version of avsidl will work with future versions of AVS and/or IDL. In fact, I know that avsidl will be completely reworked for for IDL 4.0.

integer_control

Author : **A. Jacobson**
UCLA Dept. Pharmacology / Crump Institute
Number : 1776
Language : C
Submitted : 02/09/94
Version : 1.000
Last Updated : 02/09/94
Ported To : HP Sun

This module was submitted as one of a four-module suite - 1) dual control - to control flow of field and integers to downstream modules, 2) field control - to control flow of an AVS image to downstream modules, 3) integer control - to control flow of an integer to downstream modules, and 4) real control - to control flow of an real to downstream modules. This module serves the same purpose as Field Control except for integers instead of fields. Its operation is identical. See Field Control for further info. An example of its use is to allow animate integer to remain connected to a number of integer ports on one or several modules while allowing the user to select which of these downstream modules will be effected. Provision is also made for manually sending single integer updates downstream.

integer_to_real

Author : **Lars M. Bishop**
National Center for Comp. Elect., Univ. of Illinois
Number : 1572
Language : C
Submitted : 02/11/93
Version : 1.000
Last Updated : 02/11/93
Ported To : DEC IBM

Integer to Real simply takes in an integer-valued input port, and passes its truncation to the output port as a real. This module is part of the the Real Math module group, which is a single file, process-sharing set of modules that allow basic operations on and between integer and real AVS data types. They have been most useful in dealing with linkage of several AVS parameters. AVS has no simple way of performing math operations on reals, even though it does allow math operations on fields.

interp_cmap

Author :	**Larry Gelberg**
	Advanced Visual Systems
Number :	1288
Language :	C
Submitted :	07/03/92
Version :	1.000
Last Updated :	07/03/92
Ported To :	Sun HP

interp_cmap interpolates between two colormaps. This is particularly useful when using the AVS Animator module because it does not interpolate between colormaps for you. The dial scale controls the blending between the colormaps. When scale = 0.0 the output is entirely cmap1, when scale = 1.0, the output is entirely cmap2 and in the middle, the output is interpolated between the two. Every value of every band (hue, saturation, value, and opacity) is evaluated. This module assumes colormaps which are 256 entries long.

irreg_2_reg

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1047
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Converts an irregular field to one which is uniform by removing the coordinate information from the irregular field. Converting an irregular field to one which is uniform is useful when one wants to create a "scalar mesh" (using field2mesh) where the elevation is derived from an associated scalar value in the field. Field 2 mesh creates a "scalar mesh" only when the input field is uniform (not when rectilinear or irregular).

last_cone

Author :	**Mark Pinches, Southampton Parallel Applications Centre, UK** and Ian Curington, AVS Inc.
Number :	1667
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

The lastcone module transforms an AVS geometry, replacing a set of disjoint lines with "cones" constructed out of polygons. The cone tristrip is now upto 2x more efficient, and has a subdivision slider control. If the input has values the module will make colour coded cones based on the input lines and colormap. The cone is only drawn at the end of a polyline structure, so it can be used with cylinder or tube, to create solid arrows. The cone is sent on one geom output port, while the original polyline is sent on the second port, such that it may be sent to cylinder. Min and Max controls allow lines of only certain bounded length to have cones, shorter or longer lines do not get cones.

linearop1D

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1517
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

linearop1D is an avs module executable which contains 3 modules hacked from khoros. Linear Operations. ** dacorr ** generates biased, unbiased or FFT based autocorrelation estimate ** dconvo ** generates time domain or FFT based circular convolution estimate ** dxcorr ** generates biased, unbiased or FFT based cross-correlation est.

local_area_stats

Author :	**John Tee**
	AVS Inc. Consultant
Number :	1666
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	IBM Kubota

LOCAL AREA STATS is an image processing module, similar to LOCAL AREA OPS, but including other measurements. For all choices, the pixel value is replaced by the appropriate statistical measurement of the local area around it, as determined by the kernel width parameter. The most useful measurement provided by this module is the standard deviation, which can be used to create adaptive thresholding fields. Skewness and Kurtosis are defined in the information for STATISTICS which calculates the global values of these parameters over an entire field. The kernel width is constrained to be an odd number, so that it can be centred on a pixel. The data type of the output field is the same as the input field in the cases of min, max, mean and median, but is always of float type for std dev, skewness and kurtosis.

log_field

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1304
Language :	C
Submitted :	08/26/92
Version :	1.000
Last Updated :	08/26/92
Ported To :	DEC Sun HP

log field computes a logarithm of the input field, and places the result into the data portion of the output field – the output field is the logarithm of the input field.

lumin_mix

Author :	**David Haith**
	Power-Media
Number :	1050
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Mixes two images according to pixel luminance. The left input is foreground , the right input background. Both images must be the same size. The mix parameter controls the effect. The blend is on pixel luminance of foreground image.

luminence

Author :	**Earl W. Hood** Convex Computer Corporation
Number :	1557
Language :	C
Submitted :	11/16/92
Version :	1.000
Last Updated :	11/16/92
Ported To :	IBM

Luminence computes the luminence of an image. However, it differs from the standard AVS luminence module in that one can control how much red, green, and blue contribute to the brightness of an image. I found this functionality useful when using the module with the replace_alpha and composite modules (eg. I can make any color with red be fully opaque while other colors are transparent).

matrix_algebra

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1518
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

matrix_algebra is an avs module executable which contains10 modules hacked from khoros. Matrix Manipulation. ** DEC mcovar ** computes maximum likelihood estimate of the covariance matrix ** meigen ** computes eigenvectors and eigenvalues of real or complex sq matrix ** minfo ** obtains information about a matrix ** minvert ** computes the inverse of a real or complex matrix ** mlde ** solves system of const. coeff. linear ordinay differential eqs. ** mlse * solves a system of linear silseaneous equations ** mlud ** computes the LU decomposition of a matrix ** mmult ** multiplies two matricies ** msvd ** computes the singular value decomposition of a matrix ** mtrans ** transposes the matrices contained in the input image

matte_3x3

Author :	**David Haith** Power-Media
Number :	1052
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Creates an alpha plane matte which is antialiased along with the rest of the image. Sample ratio is 9 into 1. The output image is 1/3 size in each direction. The input image height and width must be a factor of 3, or the indexing in this module falls apart.

matte_sample

Author :	**David Haith**
	Power-Media
Number :	1051
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Creates an alpha plane matte which is antialiased along with the rest of the image. The sample ratio is 4 into 1. The output image is 1/2 size in each direction. The input image height and width must be a factor of 2, or the indexing in this module falls apart.

min_max_fiel

Author :	**Terry Myerson**
	International AVS Center (NCSC)
Number :	1140
Language :	C
Submitted :	03/03/92
Version :	1.000
Last Updated :	03/03/92
Ported To :	DEC IBM Sun HP

Calculates the smallest and largest value in the data segment of a field. the field can be of any type, any vector. The outputs will always be floating point values. When used with continuous stats (another module in the ftp site), this was used to compile a running minimum/maximum tally of the data in a time series. This minimum/maximum was then used to fix the limits of a colormap.

mix2

Author :	**John Tee**
	AVS Inc. Consultant
Number :	1669
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

MIX2 mixes two fields together by combining corresponding elements, such that output = input1 * weight1 + input2 * (1 - weight1) If the two fields have different dimensions, the operation is performed on the overlapping portion of the data - ie using the smaller dimension. The module is used, for example, to produced a “sharpened” image by mixing high-pass filtered data with a proportion of the unprocessed data. It can also be used in conjunction with ANIMATED FLOAT to provide a gradual fade between two images.

modify_seq1D

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1519
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

modify_seq1D is an avs module executable which contains 9 modules hacked from khoros. Signal/Sequence Modification. ** dcpad ** performs a constant value padding of a 1D data sequence ** dextract ** extracts 1-dimensional subsignals from the input data file ** dinsert ** inserts 1-D vectors from input file (img2) into vectors in img1 ** dmirror ** generates a mirror data file ** dnormal ** normalizes the magnitude of the input data ** dscale ** rescales data for processing thus minimizing roundoff errors ** dshift ** shifts or rotates the 1-D signals in the field ** dsubsamp ** subsamples 1-D vector(s) from the field ** dwindow ** multiplies user selected 1-D data sequence by window function

my_mirror

Author :	**Michael Rangitsch**
Number :	1131
Language :	FORTRAN
Submitted :	01/27/92
Version :	1.000
Last Updated :	01/27/92
Ported To :	DEC IBM Kubota HP

Mirrors a 3D data set about one of the axes, selected by the user. This is a convenient way to fill out a 3D data set which was computed using a plane of symmetry. The AVS supplied mirror module mirrors the geometry correctly, but mirrors the data incorrectly. This version of my_mirror simply negates one of the coordinates, depending on the plane choice. For regular data, it inverts the mapping in the chosen direction.

natural_log

Author :	**Earl W. Hood** Convex Computer Corporation
Number :	1560
Language :	C
Submitted :	11/16/92
Version :	1.000
Last Updated :	11/16/92
Ported To :	IBM

The natural log filter module computes log base e of a real field, i.e. newvalue = LOG e (value)

no_format

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1520
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

no_format is an avs module executable which contains 3 modules hacked from khoros. No File Formats. ** asc2AVS ** transforms ASCII data into a field ** raw2AVS ** transforms RAW data into a field vconvert performs data storage type conversions ** AVS2raw ** writes ONLY the raw image data present onto descriptor output_fd

opening

Author :	**David E. Harris** Georgia Institute of Technology
Number :	1173
Language :	C
Submitted :	04/16/92
Version :	1.000
Last Updated :	04/16/92
Ported To :	Kubota

This module performs the mathematic morphological opening operation on grayscale (or binary) images. See Haralick et al. "Image Analysis using Mathematical Morphology", IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI-9, No. 4, pp. 532-550, 1987 for a more complete discription. The opening of image A by structuring element B (a binary mask) is defined as the dilation (by B) of the erosion of A by B. This is done by feeding the output of the ERODE module into the input of the DILATE module and using the same mask for both. The OPENING module just combines these two modules for convenience. See also the CLOSING module.

output_dither

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1521
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

output_dither is an avs module executable which contains 3 modules hacked fomr khoros. Dither. ** vdith45 ** performs 8x8 ordered block dither with major axes on 45' angles ** vdither ** performs 5x5 ordered dithers ** verrdif ** performs dithering operation using 4-direction error diffusion

output_print

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1522
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

output_print is an avs module executable which contains 2 modules hacked from khoros. Print Image Hardcopy. ** vlpr ** formats a byte image for output on a standard line printer ** vpostscr ** formats a byte image for output on a Postscript laser printer

panel_pan

Author :	**Anonymous** Stardent Computer
Number :	1060
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Uses upstream transform information to crop an input image and display one panel at a time. It is designed to take an input image, and copy it to the output port. This is a straight pass-through. The size of the output image is 1/panel_res of the input image. The module sensed mouse activity in borders near the edge of the window, and will flip to the next panel in that direction. The mouse position is fed externally from this module, getting the current mouse position and button state from the "struct upstream transform" input port. This module is designed to be used with "track_cursor", which displays the image and catches the X events.

pencil_sketc

Author :	**Larry Gelberg**
	Stardent Computer
Number :	1061
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Simulates a pencil sketched image by randomly sampling an image and generating blobs based on the color of the sample. The number of samples is controlled by the N-STROKE dial (5000-10000 makes nice pictures). The length and widths of the lines are controlled by the LENGTH and WIDTH dials. The lines are all placed down at the same angle (controlled with the ANGLE dial) unless the USE GRADIENT button is turned on in which case, the gradient of the luminance of the sampled pixel is used to determine the line angle. Since all the samples are added into each other, you will quickly max out the data values. In order to avoid this, there is a dial called FILTER SCALE which tones this down. Finally, you can control the color of the background with the GREY PAPER knob. 0.0 represents black and 1.0 represents white.

persp_field

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1062
Language :	C
Submitted :	10/10/91
Version :	2.000
Last Updated :	10/10/91
Ported To :	Kubota IBM

Applies a perspective transformation, as specified by its parameters, to each and every grid point in a 3 space irregular field. This module was designed to be used with the curvilinear volume renderer, "bezier volume." The input field consists of an irregular field with 3-space coordinates. The data type and vector length may be any configuration. The output field is the same size and type as the input field, but with the coordinates transformed according to the parameters.

real_control

Author :	**A. Jacobson** UCLA Dept. Pharmacology / Crump Institute
Number :	1777
Language :	C
Submitted :	02/09/94
Version :	1.000
Last Updated :	02/09/94
Ported To :	HP Kubota Sun

This module was submitted as one of a four-module suite - 1) dual control - to control flow of field and integers to downstream modules, 2) field control - to control flow of an AVS image to downstream modules, 3) integer control - to control flow of an integer to downstream modules, and 4) real control - to control flow of an real to downstream modules. This module serves the same purpose as Integer Control except for floats instead of integers. Its operation is identical. See Field Control for further info. An example of its use is to allow animate float to remain connected to a number of float ports on one or several modules while allowing the user to select which of these downstream modules will be effected. Provision is also made for manually sending single real updates downstream.

real_to_integer

Author :	**Lars M. Bishop** National Center for Comp. Elect., Univ. of Illinois
Number :	1573
Language :	C
Submitted :	02/11/93
Version :	1.000
Last Updated :	02/11/93
Ported To :	DEC IBM

The Real to Integer module simply takes in a real- valued input port, and passes its truncation to the output port as an integer. This module is part of the the Real Math module group, which is a single file, process-sharing set of modules that allow basic operations on and between integer and real AVS data types. They have been most useful in dealing with linkage of several AVS parameters. AVS has no simple way of performing math operations on reals, even though it does allow math operations on fields.

reconstruct_slic

Author :	**Joseph A. Oswald, III** Radiation Physics, Duke University Medical Center
Number :	1708
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	Kubota

The reconstruct slice module takes the result of a “slave slicer” operation, which is a 2D uniform scalar field representing a slice through a 3D scalar field, and constructs a 3-space field with identical data to this 2D uniform field, but with coordinates corresponding to the original 3D coordinate space. This output is suitable for use with “field to mesh” to render the 2D slice with a 3D geometry. The three modules in this package, “slave slicer,” “specify slice,“ and ”reconstruct slice,“ are a flexible and efficient alternative to “arbitrary slicer” and Wes Bethel's “new arbitrary slicer.” Slice specifications are generated by a module separate from the actual slicing module. That permits a single specifying module to specify identical slices through multiple data sets, and also for users to create their own modules to specify slices. The slicing algorithm takes 2D slices of 3D scalar fields (uniform or rectilinear) and outputs them in the most space efficient format, 2D uniform. 2D 3-space irregular output, similar to that output by “new arbitrary slicer” can be generated, if needed, using “reconstruct slice.” The slicing algorithm is generally faster than “new arbitrary slicer,“ at least on the Kubota architecture.

rect2pol

Author : **anonymous**
Number : 1891
Language : C
Submitted : 09/07/95
Version : 1.000
Last Updated : 09/07/95
Ported To : DEC_ultrix DEC_alpha

This module will convert any 2D or 3D, 2 space or 3 space field from rectangular coordinates to polar, and vice versa. The output field is always an irregular field, but the input can have any uniform or type setting. When converting into polar coordinates, the output angles will be in the range of 0 to 2*pi. When converting from polar to rectangular, any angles can be used, but it is possible to overlap data if the range of angles is greater than 2*pi. For cylindrical coordinates, points along the z-axis will have a radius equal to zero, and therefore an undefined angle. the atan2 function will usually set this angle to 0 or PI. For that reason, polar to rectangular to polar conversions will not always produce the original input field, though no significant data is lost. For spherical coordinates, this effect only concerns points located at the origin. The Polar Type parameter is removed for 2 space fields since cylindrical coordinates are assumed.

remap_field

Author : **Phil McDonald**
NOAA/ERL/Forecast Systems Laboratory
Number : 1268
Language : C
Submitted : 05/27/92
Version : 2.000
Last Updated : 07/05/93
Ported To : IBM

The remap field module provides the capability of changing a field's computational space to physical coordinate mapping scheme. For example, a uniform field may be remapped to an irregular field, or vice versa. Another example is to change the coordinates of an irregular field from one grid system to another. The data of one field (the data field) can be mapped in the same manner as an optional second field (the grid field). The updated version includes several changes to the module and the libraries it uses as well.

remote_gis

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1523
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

remote_gis is an avs modules executable which contains 7 modules hacked from khoros. Standard File Formats. ** big2AVS ** converts a BIG image file into a field ** dem2AVS ** converts digital elevation map data (USGS format) to a field ** dlg2AVS ** converts an optional format digital line graph file to a field ** elas2AVS ** converts an ELAS image file format into a field ** elasinfo ** reads and prints an ELAS image fiel format header ** AVS2big ** converts a BIG image file into a field ** vll2utm ** converts geographical to universal transverse mercator coordinates

rgb_to_cmyk

Author :	**Anonymous** Stardent Computer
Number :	1068
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota IBM

Converts from red-green-blue to cyan-magenta-yellow-black color space representation. Both AVS filters and offline standalone program versions are included. prepend.c and denude.c are used for conversion to and from SCITEX CT2T cmyk tape format files. The input and output are both field 2d 4-vector byte (image data).

rgb_to_hsv

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1130
Language :	C
Submitted :	01/26/92
Version :	1.000
Last Updated :	01/26/92
Ported To :	Kubota HP

Converts an image, represented as ARGB (alpha, red, green, blue) to HSVA (hue, saturation, luminosity, alpha). The H component ranges from 0.0 to 1.0. You might use this module to display only the V (or luminosity) component of an image, i.e. grayscale. A sample network to do this is provided within the .txt file accompanying this module.

river_fill

Author :	**Ken Flurchick**
	North Carolina Supercomputing Center
Number :	1580
Language :	C
Submitted :	02/11/93
Version :	1.000
Last Updated :	02/11/93
Ported To :	IBM Kubota HP

This module is a coroutine which continuously pads a two-dimensional array with data until it is turned off. This can be used with the "river bed" module which generates the coordinates for the array, to show a simulation of concentration over time.

scat_to_tri

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1681
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP Kubota Sun

This is a 2D grid generation routine, that takes random scatter data in a plane, and creates a delaunay triangulation from them. It sends node position data and connectivity information out via a UCD structure, containing only the 2D elements. It only grids based on the X and Y coordinates of the input data, but preserves the Z values on the UCD, so irregular 3D sheets may be generated. The core algorithm is from a public domain library for voronoi and delaunay triangulation at "netlib", called "sweep2". The only modifications to the package from netlib is the re-write of "main.c" and "output.c" to create AVS coroutine structures, and to pass data from shared memory rather than file I/O. All data stored at each point in the input field is transfered unchanged to the UCD, for further interpretation or processing.

scat_znmo

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1755
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This routine takes two 1D irregular fields, looks up the Z (depth) in the first one, finds the corresponding depth of the second one, and replaces the X-Y of the first by the second at that point. Linear or Quadratic interpolation is available by user choice. Both fields can be different lengths, and have no depths in common. The first must be contained within the range of the second, or the lookup will fail. All the data from the first is passed to the output. The underlying utility routines could easily be applied in other contexts. NOTE - Advanced AVS Module Techique used. This routine contains an advanced module writer trick, of modifying the input port data inplace, without explicitly or implicitly copying the data out of shared memory. This REQUIRES that a special flag be turned on in your .avsrc file - ReadOnlySharedMemory 0

scatter_downsize

Author :	**Michael J. Rangitsch**
Number :	1886
Language :	C
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_ultrix DEC_alpha

This module samples a scatter data set by the input rate. It is useful for downsizing samplers for use as vector locations or streamline/particle advection starting points.

scatter_to_spars

Author :	**Penny Rheingans** US EPA Scientific Visualization Center
Number :	1309
Language :	C
Submitted :	08/30/92
Version :	1.000
Last Updated :	08/30/92
Ported To :	DEC Sun HP

Convert a 1D field containing scattered data points into a (potentially) sparsely filled uniform field. The output field has as many dimensions as the input field did coordinate dimensions. For example, if the input is a 1D 3-space field, the output will be a 3D 3-space field. Each dimension of the output field will extend from the minimum value to the maximum value for that coordinate in the input field. Coordinates in the input field are truncated to integers. Accordingly, if a dimension of the input field extends from 15.3 to 19.0, the output field will have 5 elements in that dimension.

segment

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1524
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

segment is an avs module executable which contains 8 modules hacked from khoros. Segmentation. ** v1bgamut ** generates a 1 band image with map of displayable color levels ** vclose ** tries to close the boundaries ** vdistance ** computes the distance from the nearest boundary point ** vdyth ** performs dynamic thresholding on an image ** vgamth ** generates a binary image (value or 0) ** vgrow ** computes the median axis inverse transform ** vmediaxis ** computes the medial axis transform ** vthresh ** seperates binary image (value or 0) by thresholding input image

set_alpha

Author :	**David Haith** Power Media
Number :	1095
Language :	C
Submitted :	10/15/91
Version :	1.000
Last Updated :	10/15/91
Ported To :	Kubota DEC IBM Sun HP

Sets data in the alpha plane of the output image according to incoming luminance of an input image. Copies the R G B data across unchanged, but replaces the alpha channel with either the luminance or 0, based on a threshold key from the three (R G B) parameter dials.

set_extents

Author : **Wes Bethel**
Lawrence Berkeley Laboratory
Number : 1145
Language : C
Submitted : 03/10/92
Version : 3.000
Last Updated : 06/20/92
Ported To : IBM Sun Kubota HP

Maps uniform fields, by default, into world coordinates. Uses the formula (for each nspace dimension called _i_)- Min[i] = 0, Max[i] = dimensions[i] - 1. This arrangement will map a volume of data, where each of the logical i,j,k dimensions of data are equal (a 64**3 dataset, for example) and map it into a "cube." For most purposes, this is fine. However, consider an MR dataset in which the slices have a fine resolution, but there are only a few slices. The slices have thickness which is lost using the above mapping function.

shutter

Author : **Wes Bethel**
Lawrence Berkeley Laboratory
Number : 1571
Language : C
Submitted : 01/29/93
Version : 1.000
Last Updated : 01/29/93
Ported To : DEC IBM Kubota HP

The shutter module allows the user to control the flow of an AVS image to downstream modules. This module is designed for use in networks in which animation sequences are created using the geometry renderer. This module allows for the idiosyncracies of the AVS render geometry module which will perform several pixmap updates prior to finishing the rendering of the entire scene. Shutter will be useful in making sure the entire scene has been rendered prior to passing the resultant image downstream to a data output module (which may put the image onto a videodisk, or write the image to a file).

size_fit

Author : **John Tee**
AVS Inc. Consultant
Number : 1683
Language : C
Submitted : 09/29/93
Version : 1.000
Last Updated : 09/29/93
Ported To : HP IBM Kubota Sun

SIZE FIT is used to make 2 fields the same size, typically before passing them to a module which combines them in some way. The input field is rescaled, using linear interpolation, to be the same size as the template field. The template field is unaffected.

slave_slicer

Author :	**Joseph A. Oswald, III** Radiation Physics, Duke University Medical Center
Number :	1709
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	Kubota

The slave slicer module extracts a 2D slice of a 3D scalar field. The slice to be extracted is specified by the slice_spec structure input. The slice_spec structure will usually be produced by a separate module that contains control widgets for specifying the slice in a way suitable for the given application. The "specify slice" module, for instance, creates a slice_spec scaled to the input data extents, and specified by Euler angles. Other modules could be written if another method of specification is desired. A module has been written, for instance, to take slices perpendicular to a beam of radiation used to treat cancer. The output is a 2D uniform field, with coordinates specified in the 2D coordinate system. The minimum and maximum X and Y coordinates correspond to the extents given in the slice_spec. This data type is the most space-efficient representation for image data. If the coordinates of the data in the original 3D system are desired, the "reconstruct slice" module will reverse the transformation and produce a 2D 3-space irregular scalar field, suitable for visualization with "field to mesh." The geometry output is a simple rectangular polygon corresponding to the size and position of the 2D slice in 3D space. It is useful for quickly depicting the slice in relation to other objects in 3D. The three modules in this package, "slave slicer," "specify slice," and "reconstruct slice," are a flexible and efficient alternative to "arbitrary slicer" and Wes Bethel's "new arbitrary slicer." Slice specifications are generated by a module separate from the actual slicing module. That permits a single specifying module to specify identical slices through multiple data sets, and also for users to create their own modules to specify slices. The slicing algorithm takes 2D slices of 3D scalar fields (uniform or rectilinear) and outputs them in the most space efficient format, 2D uniform. 2D 3-space irregular output, similar to that output by "new arbitrary slicer" can be generated, if needed, using "reconstruct slice."

smooth

Author :	**anonymous**
Number :	1892
Language :	C
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_ultrix DEC_alpha

This module will take a 2 dimensional field, and perform smoothing operations on it. The input field will be smoothed in the Y direction, so the data is expected to be in several independent columns (graph viewer format). There are three smoothing methods; Average, Median, and Weighted Average.

specify_slice

Author :	**Joseph A. Oswald, III** Radiation Physics, Duke University Medical Center
Number :	1707
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	Kubota

The specify slice module creates a slice_spec structure, suitable for use with the “slave slicer” module to extract 2D fields from 3D scalar fields. The 3D field can be of any data type - the field is used only to scale the extents so that the 2D slice will cover the extents of the field. The slice orientation and resolution are controlled by the parameters. The X and Y resolutions are identical, and the scaling of the slice is selected to cover the “shadow” of the 3D field in this orientation and maintain the “aspect ratio” of the input. That is, parallel slices made by varying the “Z Translate“ parameter will all be the same size, and the geometrical proportions of the input are preserved. The three modules in this package, “slave slicer,” “specify slice,” and “reconstruct slice,” are a flexible and efficient alternative to “arbitrary slicer“ and Wes Bethel’s ”new arbitrary slicer.“ Slice specifications are generated by a module separate from the actual slicing module. That permits a single specifying module to specify identical slices through multiple data sets, and also for users to create their own modules to specify slices. The slicing algorithm takes 2D slices of 3D scalar fields (uniform or rectilinear) and outputs them in the most space efficient format, 2D uniform. 2D 3-space irregular output, similar to that output by “new arbitrary slicer“ can be generated, if needed, using “reconstruct slice.” The slicing algorithm is generally faster than “new arbitrary slicer,” at least on the Kubota architecture. RELATED MODULES - slave slicer, reconstruct slice

spectest1D

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1525
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

spectest1D is an avs module executable which contains 3 modules hacked from khoros. Spectral Estimation. ** darpsd ** determines the 1D autoregressive PSD and Phase spectral estimate ** dminorm ** implements the minimum norm for spectral estimation ** dmusic ** implements the MUSIC for spectral estimation

standard

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1526
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

standard is an avs module executable which contains 13 modules hacked from khoros. Standard File Formats. ** fits2AVS ** converts a FITS image into a field ** mat2AVS ** reads single matrix from file specified and places it in image ** pbm2AVS ** converts a PBM image into a field ** rast2AVS ** converts a SUN raster image into a field ** tga2AVS ** converts TGA image into a field ** tiff2AVS ** converts TIFF image into a field ** AVS2fits ** converts a field into a FITS fiel format ** AVS2mat ** outputs a matrix in desired format ** AVS2pbm ** converts a field into a PBM image ** AVS2rast ** converts a field into a SUN raster image ** AVS2tiff ** converts a field into a TIFF image ** AVS2xbm ** converts a field into a x11 bitmap format (xbm) ** xbm2AVS ** converts an x11 bitmap file (xbm) into a field

stereogram

Author :	**Larry Gelberg**
	Advanced Visual Systems
Number :	1700
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP IBM Kubota Sun

STEREOGRAM produces a version of a "random dot stereogram" based on an image which represents distance to the eye. An easy way to get this kind of image is to use the Z-buffer output from the AVS geometry viewer's software renderer. A random dot stereogram is an image where if you diverge your eyes while looking at it, the left and right eyes will see a different image and a three-dimensional surface will appear. Two squares are drawn in the top of the display. One method of viewing the stereogram would have you stare "past" the image so that the squares appear to overlap. Hold that for a few seconds and (for some people.) a shape will appear to either protrude from or recede into the image plane.

stick

Author :	**David Haith**
	Power Media
Number :	1083
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Sticks one image over another with a variable mix control, and with offset translation for image alignment. The mix has further level key control. The five parameters, all integer, are "x shift", "y shift", "mix", "level", and "key level".

stream_to_scatte

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1684
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

The stream_to_scatter module generates an output field with the coordinates of the stream path. This is used to create other visual or analytical processes at the positions along the streamline path. Scatter markers could be drawn, or the scatter field could control the sampling positions for other interpretation modules such as hedgehog.

string_parm_to_field

Author :	**Joshua Goldstein** Research Systems Incorporated
Number :	1788
Language :	C
Submitted :	02/15/94
Version :	1.000
Last Updated :	02/15/94
Ported To :	IBM Sun

This is one of a suite of seven AVS/IDL modules. The modules are all in the FILTERS subdirectory at the IAC. They are IDL - a UNIX utility that gets IDL and AVS communicating; and six AVS modules that convert data between IDL and AVS - field_to_float_parm, field_to_int_parm, field_to_string_parm, float_parm_to_field, int_parm_to_field, and string_parm_to_field. These modules have only been tested on the RS 6000 and Sun Sparcstation. Binaries have been compiled on the HP 700 but its usability is unknown. Binary executables are stored at the IAC for all three of these platforms. There are also some demo files at the IAC, stored in the file demo.tar. You must be running IDL 3.1 or later in order to use these. No guarantee is made that this version of avsidl will work with future versions of AVS and/or IDL. In fact, I know that avsidl will be completely reworked for for IDL 4.0.

strip_cfd

Author :	**Rob Kuyper** Univeristy of Technology Delft, The Netherlands
Number :	1319
Language :	C
Submitted :	08/30/92
Version :	1.000
Last Updated :	08/30/92
Ported To :	DEC Sun HP

strip cfd is a filter module which changes a cfd dataset in order to remove the staggering of velocity components (in finite volume methods) or to remove virtual points from the dataset.

subregion

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1527
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

subregion is an avs module executable which contains 5 modules hacked from khoros. Subimage/Subsignal Operations. ** vextcent ** extracts rectangular image from center of larger image ** vextract ** removes rectangular piece from an image to produce a subimage ** vinsert ** inserts subimage into the input image to form output image ** vpad ** pads a data sequence with a contstant ** vsamp ** samples image along line, given image and a line through it

super_sample

Author :	**David Haith** Power Media
Number :	1084
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Creates an anti-aliased (filtered and downsized) version of an input image. It used a simple box filter, averaging pixels in the sub-region. The sample ratio is 4 into 1. The output image is 1/2 size in each direction. The input image height and width must be a factor of 2, or the indexing in this module falls apart.

surface

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1528
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

surface is an avs module executable which contains 6 modules hacked from khoros. Surface Calculation and Modification. ** vcost ** computes surface arc length from one pixel to every other pixel ** vdetilt ** computes best-fit plane for image, then subtracts it from image ** vgettilt ** computes the least-squares best fit plane for an image ** vslope ** computes the slope and aspect images from input elevation data ** vsurf ** takes a surface image and produces two output images ** vtilt ** adds specified plane to image to correct for a tilt in luminance

switch_2

Author :	**A. Jacobson** UCLA Dept. Pharmacology / Crump Institute
Number :	1779
Language :	C
Submitted :	02/09/94
Version :	1.000
Last Updated :	02/09/94
Ported To :	HP Kubota Sun

This module was submitted as one of a two-module suite - 1) switchfire, which switches between two fields, and controls flow to downstream modules, and 2) switch, which switches between two fields to downstream modules. Switch will be useful in keeping the flow of data controllable to bottleneck points without requiring the user to repeatedly connect and disconnect modules to control their source field. The module will send data downstream whenever the chosen input is updated, or the A/B switch is switched. Both inputs are not required for the module to execute. This module allows the user to control the flow of AVS fields to downstream modules. Often it may be desireable to switch the input path between modules, macros or whole branches of a network frequently without having to connect and disconnect inputs repeatedly. This is especially true in larger networks wherein it would be impractical to duplicate the downstream module or tree, and switching inputs may require stopping the flow executive to allow for time to make the change.

switchfire

Author :	**A. Jacobson** UCLA Dept. Pharmacology / Crump Institute

Number :	1778
Language :	C
Submitted :	02/09/94
Version :	1.000
Last Updated :	02/09/94
Ported To :	HP Kubota Sun

This module was submitted as one of a two-module suite - 1) switchfire, which switches between two fields, and controls flow to downstream modules, and 2) switch, which switches between two fields to downstream modules. Switchfire will be useful in keeping the flow of data controllable to bottleneck points without requiring the user to repeatedly connect and disconnect modules to control their source field and execution. The module will switch between two input fields and send data downstream. In control mode, data is sent when triggered by the "send data" button, or, in continuous mode, whenever the chosen input is updated, the A/B choice is switched, or the send data button is triggered. Both inputs are not required for the module to execute. This module allows the user to control the flow of AVS fields to downstream modules. Often it may be desireable to switch the input path between modules, macros or whole branches of a network frequently without having to connect and disconnect inputs repeatedly. Similarly, it is often desireable to stop, start, or trigger data flow and execution to individual modules or branches of a network. This is especially a problem with modules that overwrite existing files whenever they are updated, or are time consuming to run when frequently not necessary.

sysModuleName

Author :	**John Stephen** Tessella Support Services
Number :	1686
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

sysModuleName - given the alias name of any module in the network this utility outputs the system's name for the module. This will return the system name for the module when its alias is typed into the parameter. For example if you use AVS CLI - "avs¿ module "geometry_viewer" -alias geomV", the system might name the instance of the geometry viewer as "geometry viewer".user.17. The module can be referenced by this name or by its alias "geomV" for most purposes, but some commands will only accept the system name. This module will get the system name from the alias which makes an application more robust. Thus in the example, if "geomV" is entered into the parameter, "geometry viewer".user.17 is output from the module.

table_editor

Author :	**Ian Curington** Advanced Visual Systems
Number :	1538
Language :	C
Submitted :	10/03/92
Version :	1.000
Last Updated :	10/03/92
Ported To :	Sun DEC HP

Like a mini-spreadsheet application, this module presents a segment of a 2D array and allows interactive editng of the field values. Takes an optional input field, displays the contents as a 2D array table of button cells. Each cell or group of cells may be selected (high-lighted) and new values entered. The modified field is then sent to the output port. Values can be replaced, added, multiplied, or divided into existing field values. The code shows how to create a page layout from within the module code. With no input port, this module can be used to synthesize new fields. If the input field is multi-vector, each vector can be selected from a choice list, and values can be edited in each channel, while synthetic data can only be created in scalar mode.

thresholdclamp

Author :	**A. Jacobson** UCLA Dept. Pharmacology / Crump Institute
Number :	1858
Language :	C
Submitted :	10/17/94
Version :	1.000
Last Updated :	10/17/94
Ported To :	DEC_alpha HP IBM Sun

This module will threshold or clamp the values of a field with or without a 2 or 3D ROI. This module does a few things that all preexisting AVS filters frustratingly don't do. 1. Thresholding (segmentation) can be done on a field, setting all values below Min to 0, and all values above Max to _255_, with all values inbetween unchanged. This can be used to create 3 data ranges, 0, min¡range¡max, and 255. 2. Clamping is performed such that all values below Min are set to Min, _except_ zero, and all values above Max, set to Max. 3. ROI's can be applied to this data. Any point that is _not_0_ in the ROI field is filtered in the data field. Areas that are 0 in the ROI can either be left unchanged or set to 0. This module will take either a 2D ROI, applying the ROI to each "Z" plane if a 3D data field is used, or a _3D_ ROI, applying each point of the ROI field to the data field. The ROI field must have the same dimensions as the data field, either in x,y (if 2D ROI) or x,y,z (if 3D).

transform1D

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1529
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

transform1D is an avs module executable which contains 6 modules hacked from khoros. Transforms (1D). ** dfft1d ** performs the 1D Fourier transform on a set of data sequences ** dfht ** performs a 1-D Fast Hadamard Transform on a signal ** dhart1d ** performs the 1-D forward Hartley transform of a sequence ** difft1d ** performs 1-D inverse Fourier transform on set of data sequences ** dihart1d ** performs 1-D inverse Hartley transform of a data sequence ** dmpp ** determines the magnitude, power, or phase of each 1-D data point

transform2D

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1530
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

transform2D is an avs module executable which contains 3 modules hacked from khoros. Tranforms (2D). ** vfft ** computes the 2 dimensional Fast Fourier Transform of an image ** vfht ** performs a fast hadamard transform on the image ** vmpp ** computes various representations of complex image data

transform_po

Author :	**Anonymous** Convex Computer
Number :	1148
Language :	C
Submitted :	03/10/92
Version :	1.000
Last Updated :	03/10/92
Ported To :	Kubota DEC IBM Sun HP

Performs a scale and offset of each dimension in coordinate space, according to the input parameters "scale x", "offset x", "scale y", and "offset y". It handles 1D, 2D, or 3D input fields, with rectilinear or irregular mapping. The input data may have any vector length, but must be non-uniform. The output field has the same dimensionality as the input field.

transpose_coords

Author :	**Phil McDonald** NOAA/ERL/Forecast Systems Laboratory
Number :	1269
Language :	C
Submitted :	05/27/92
Version :	1.000
Last Updated :	05/27/92
Ported To :	Kubota

The transpose coords module provides the capability of transposing the coordinates of non-uniform fields. This capability is convenient for making a vertical slice horizontal, without having to resort to the transformation capabilities of the render geometry module.

tubeness

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1791
Language :	C
Submitted :	04/21/94
Version :	2.000
Last Updated :	07/15/94
Ported To :	DEC HP IBM Sun

The tube module transforms an AVS geometry, replacing a set of disjoint lines with "tubes." The tubes are polygonal approximations to cylinders. The number of sides used to represent the cylinder is under user control. Version 2 update - those line segments which are "degenerate" will be cause a sphere of radius "radius scale" to be generated rather than a flat cylinder.

ucdTri2prism

Author :	**Ian Curington** Advanced Visual Systems
Number :	1757
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

ucd_tri_to_prism - extrudes triangulated surface into 3D prism layers, given triangular 2D UCD elements and a 2D field, loft the elements into 3D prism elements, with information for the 3rd dimension coming from a 2D field, where the first dimension matches the input UCD node list, the second dimension replicates the structure, adding a Z offset and new data.

ucd_extent_range

Author :	**Ian Curington** Advanced Visual Systems
Number :	1688
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

The ucd extent range module converts model coordinate ucd extent positions into normalized dial range for ucd_slice_2d, etc. It has 3 dials, which have model coordinate ranges, these are mapped to the the extent in X, Y, Z, and normalized -1 to 1 range floats are output, with appropriate correction factors to drive ucd_slice2d.

ucd_load_node_da

Author :	**Advanced Visual Systems, Inc.**
Number :	1689
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

ucd_load_node_data adds a field of data to a UCD structure. It takes in a field of floats as UCD data, and a field of ints as node ids. It also takes in an existing UCD structure. For each node id in both the field and the UCD it adds the extra data. Hence, not all of the field float data will necessarily be added; nor will all of the UCD nodes be guaranteed to receive data (if they do not get data, zeros are inserted).

ucd_minmax

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1285
Language :	C
Submitted :	06/20/92
Version :	1.000
Last Updated :	06/20/92
Ported To :	IBM Sun Kubota HP

For Transient UCD data, where data ranges are changing, and/or the mesh is static, this module copies UCD input to UCD output, with almost everything unchanged, except for the data minimum and maximum header values, and the mesh_id. This is picked up by ucd_contour or ucd_legend, to colour the data, using user-specified values for min and max, as set on the parameter dials. This is important for animation, as the colour range must remain constant over time varying data. This module will work with either Node or Cell based data. Make sure the range is outside the actual data values.

ucd_particle

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1189
Language :	C
Submitted :	05/04/92
Version :	3.000
Last Updated :	12/15/93
Ported To :	Sun Kubota

This module animates spheres moving along the path created by either UCD streamlines or the field "streamlines" module. It post-processes the path trajectory data, and moves particles along each path. This module will work with either UCD or field data, because it inputs neither, it simply uses the other integration modules to create the paths. The animation will be faster than "particle advector", because it is not performing the path integration at the same time as the animation. The module has full up and downstream reverse transformation pass-through, so interactive positioning of the streamline control icon is still posible. This module is intended as a source example for other UCD module developers. Version 3.000 update (provided by Mike Rangitsch of Dow Chemical) adds a second (optional) input and moving particle tails.

ucd_polar

Author :	**anonymous**
Number :	1894
Language :	C
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_ultrix DEC_alpha

This module reads in a UCD structure and locates all components that have a vector length of 3. Those data vectors are then converted from x,y,z format to r,theta,z format. Only those data values are changed. Óytherwise the ucd is exactly like the original. This module always calculates theta around the Z axis. If the user changes the Polar Origin parameters, then the polar coordinates will be computed around this new Origin.

ucd_select

Author :	**Jonathan Cox** Department of Computer Science, University of Manchester
Number :	1713
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP

The ucd select adjusts the minimum and maximum values of a colormap to those of a selected scalar component of UCD nodal data. The module is functionally similar to the color range module which performs the same operation for fields. However, as ucd contour already normalises the colormap this module is not required to correctly color ucd structures. Instead, it is intended that this module be used to provide a alternative mechanism for using the color legend and label modules with UCD data to that offered by the ucd minmax module. In this respect the ucd color range module offers a superset of the functionality of the ucd minmax module - filtering the ucd input. The ucd minmax module relies on network connections to the generate colormap module in order to set the bounds of the colormap. In contrast ucd select filters the colormap output. This colormap can then be passed to the color legend module. There are two principal advantages to using this module over ucd minmax. Firstly, the module offers more control over what type of changes in the ucd input cause the min/max values to be reset. The other major advantage is that the chosen component is made available on an output port. This can then be used to control the 'Node Data' parameters of other ucd modules. This makes it possible to select a component throughout a network at a single stroke. This makes it much easier to design ucd networks with a clean user interface. The other output of the module is a title string which can be used with the label module.

ucd_split

Author :	**John Stephen** Tessella Support Services
Number :	1691
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

This module outputs a portion of the input UCD, by selecting lists of nodes, cells, from an integer field, or by typing in cell ranges.

ucd_to_field

Author :	**Ian Curington** Advanced Visual Systems
Number :	1692
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

This is a UCD filter to convert node data, component lists, node id's to AVS field and string structures for external processing. The data on the nodes of the UCD model are brought out to a floating field port, where the data can be further processed such as with extract scalar, field math, or statistics, or sent to a graph viewer. The difference between the nodal values on two different UCD structures could be computed, for instance. In "irregular" mode, the output value field is in "scatter" form, so spheres can be placed at node positions with bubbleviz and scatter dots. In addition, the node id's are converted to an integer field, and the component list is converted to and integer field, with component names and units converted to string output ports. This is very handy for debugging UCD applications, for processing UCD data using field modules, or other user written modules. Note that this is not the compliment of field_to_ucd, as this is topologicall y impossible.

ucd_tri_to_prism

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1693
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP

This is a UCD filter to extrude 2D triangle UCD's to 3D prisms. Given triangular 2D UCD elements and a 2D field, loft the elements into 3D prism elements, with information for the 3rd dimension coming from a 2D field, where the first dimension matches the input UCD node list, the second dimension replicates the structure, adding a Z offset and new data. The control field contains all information to create the structure, both coordinates and data, except for cell topology, which comes from the input 2D ucd. This is intended to be generated by a module like scat_to_tri (delaunay triangulation). Each "row" of the input field represents one 2D surface, where the 4-vector information is - 1) X-Loc, 2) Y-Loc, 3) Time (depth), and 4) Velocity. The X-Y pairs are un-ordered, and represent random surface positions.

un_jag

Author :	**Tim O. Robinson**
	UC Berkeley
Number :	1275
Language :	C
Submitted :	05/27/92
Version :	1.000
Last Updated :	05/27/92
Ported To :	DEC IBM Sun Kubota HP

This module searches for jagged regions and performs local averaging on pixels located at or near a "jag." A pixel is determined to be at a jag if either red, green, or blue values differ from one (of two) surrounding pixel r,g,b values by a threshold amount in both the x and y directions. The threshold value is set at 25 in the module. There is no widget for this parameter because adjustment is not required during normal operation. This module is intended for final processing of image data which is limited by screen resolution before printing, its research value is dubious.

unif2irreg

Author :	**John Tee**
	AVS Inc. Consultant
Number :	1694
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

This module takes an input field, such as a 1D n-vector uniform field, and interactively decide which "column" or channel should be used for each of X, Y, and Z coordinates of a scatter field. Any remaining quantities, if the field had more than 3-vectors, are placed on the output field as an n-3 vector field. This can be used to create multidimensional scatter plots, showing channel correlation by spatial clustering. It is much easier to import uniform fields using read_field or the ADIA, then use this module to create a scatter, than to define a scatter based on the external file format. This module will work with byte, float, int, double, or short data types.

unif2zscat

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1848
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP

unif2zscat - uniform 1D to Z-line scatter. This module takes an input field, such as a 1D n-vector uniform field, and creates a new field with all the same data values, vector length, etc., but in addition adds irregular coordinates, creating a scatter field. The coordinates added have X=0, Y=0, and Z changing as a function of the data index with uniform spacing. This module is a simple filter used to test the "scatter_ribbon" module, for a well log display. This module will work with byte, float, int, double, or short data types.

uniform_to_i

Author :	**Larry Gelberg**
	Stardent Computer
Number :	1089
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Converts a uniform field data set to an identical irregular one. It does this by creating explicit 3D coordinates for all the nodes of the field. This is sometimes desirable for doing things which the uniform field doesn't allow. In the documentation accompanying the module, there is a sample network shown which helps get around the fact that orthogonal slice creates a slice in 2d instead of 3d.

vec2_to_vec3

Author :	**Chris Allen**
	Unisys Corporation
Number :	1291
Language :	C
Submitted :	07/11/92
Version :	1.000
Last Updated :	07/11/92
Ported To :	IBM Sun HP

Vec2 to vec3 converts the input field 2-vector into a similar field, with the only difference being the data has been moved into a type 3-vector structure, with the extra vector component containing a 0. This module is useful, for example, if you have a uniform field of 2-d vectors and would like to view with Hedgehog, which requires 3-vector uniform data.

vec_mag_2d_v

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1090
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Computes a two dimensional gradient from a two dimensional scalar, floating point field (image). This operation is useful in many image processing applications. There are no parameters with this module. The documentation accompanying the module shows a network that reads in an image (a good test case is the "stardent.x" image), then extracts one of the rgba fields from the image, then computes a 2d gradient from the resulting gray image. Next, they extract either the x- or y-component from the gradient field and display the result as a grayscale image. Switching from channel 0 to channel 1 in extract scalar shows the y-component of the gradient.

vect_div_scal_2

Author :	**Adam Usadi**
	Rice University
Number :	1870
Language :	C
Submitted :	06/27/95
Version :	1.000
Last Updated :	06/27/95
Ported To :	DEC_ultrix DEC_alpha

Divides every element of a vector field by a scalar field

vect_div_scal_5

Author :	**Adam Usadi**
	Rice University
Number :	1866
Language :	C
Submitted :	06/27/95
Version :	1.000
Last Updated :	06/27/95
Ported To :	DEC_ultrix

Divides every element of a vector field by a scalar field

vect_div_scalar

Author :	**Adam Usadi**
	Dept. of Space Physics and Astronomy Rice
Number :	1862
Language :	C
Submitted :	06/27/95
Version :	1.000
Last Updated :	06/27/95
Ported To :	DEC_ultrix

Divides every element of a vector field by a scalar field

vector

Author :	**International AVS Center**
	North Carolina Supercomputing Center
Number :	1531
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

vector is an avs module executable which contains 1 module hacked from khoros. Multiband Conversion. ** vmapdlg ** converts the maps of a field created using dlg2AVS from actual geographical attributes to rgb of displayable color

vector_cross

Author :	**Adam Usadi**
	Dept. of Space Physics and Astronomy, Rice University
Number :	1728
Language :	C
Submitted :	12/14/93
Version :	1.000
Last Updated :	12/14/93
Ported To :	IBM Kubota Sun

This module computes either a 3D or 2D vector cross product. You should be able to modify this to allow the calculation of a dot product also.

vector_cross_2

Author :	**Adam Usadi**
	Dept. of Space Physics and Astronomy Rice
Number :	1861
Language :	C
Submitted :	06/27/95
Version :	1.000
Last Updated :	06/27/95
Ported To :	DEC_ultrix

INPUTS First_Vector - float field of space, any size , any veclen REQUIRED Second_Vector - float field of any space, any size , any veclen REQUIRED PARAMETERS OUTPUTS Cross_Product - input 3-vectors yield 3-vector output. input 2-vectors yield 1-vector output in the third direction

vector_cross_3

Author :	**Adam Usadi**
	Rice University
Number :	1864
Language :	C
Submitted :	06/27/95
Version :	1.000
Last Updated :	06/27/95
Ported To :	DEC_ultrix

Computes the vector cross product of two float fields

vector_cross_4

Author :	**Adam Usadi**
	Rice University
Number :	1865
Language :	C
Submitted :	06/27/95
Version :	1.000
Last Updated :	06/27/95
Ported To :	DEC_ultrix

Computes the vector cross product of two float fields

vector_cross_5

Author :	**Adam Usadi**
	Rice University
Number :	1869
Language :	C
Submitted :	06/27/95
Version :	2.000
Last Updated :	06/27/95
Ported To :	DEC_ultrix DEC_alpha

Computes the vector cross product of two float fields

vector_dot

Author :	**Adam Usadi**
	Dept. of Space Physics and Astronomy, Rice
Number :	1863
Language :	C
Submitted :	06/27/95
Version :	1.000
Last Updated :	06/27/95

Computes the vector dot product of any two float fields of any size and any field coordinates.

vector_dot_3

Author :	**Adam Usadi**
	Rice University
Number :	1871
Language :	C
Submitted :	06/27/95
Version :	1.000
Last Updated :	06/27/95
Ported To :	DEC_ultrix DEC_alpha

Computes the vector dot product of any two float fields of any size and any field coordinates.

vector_dot_6

Author :	**Adam Usadi**
	Rice University
Number :	1867
Language :	C
Submitted :	06/27/95
Version :	1.000
Last Updated :	06/27/95
Ported To :	DEC_ultrix

Computes the vector dot product of any two float fields of any size and any field coordinates.

vector_field

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1150
Language :	C
Submitted :	03/10/92
Version :	1.000
Last Updated :	03/10/92
Ported To :	Kubota DEC IBM Sun HP

Computes a vector field between two planes in physical space. The two inputs ports define a initial and terminating plane, perpendicular to the y axis. The vector field that is output is constructed such that the corresponding corners of each plane have vectors pointing to the other one. The input data sets consist of an x, y, z triplet in positions 0, 1, and 2 in the input array. The triplet defines the xz planes that the particles advect from/to.

warp

Author :	**International AVS Center** North Carolina Supercomputing Center
Number :	1532
Language :	C
Submitted :	10/01/92
Version :	0.100
Last Updated :	10/01/92
Ported To :	Sun DEC IBM

warp is an avs module executable which contains 4 modules hacked from khoros. Warp Images. ** vbilinco ** computes the coefficients for the two bilinear equations ** vexpwarp ** alters the spatial properties of an image ** vpolyco ** computes a set of bi-polynomial coefficients ** vwarp ** alters the spatial properties of an image

xform_field

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1038
Language :	C
Submitted :	10/10/91
Version :	2.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun

Applies a transformation, as specified from the parameters “X Rotate”, “Y Rotate“, ”Z Rotate“, ”Ortho Scale“, ”X Xlate“, ”Y Xlate“, ”Z Xlate“, and ”flip Z“. The transformation is applied to each and every grid point in a 3D, 3 space irregular field. This module was designed to be used with the curvilinear volume renderer, “bezier volume.“

y_join

Author :	**David Haith**
	Power Media
Number :	1094
Language :	C
Submitted :	10/14/91
Version :	1.000
Last Updated :	10/14/91
Ported To :	Kubota DEC IBM Sun HP

Joins up two images, with a split line running horizontally. The position of the split is controlled by the parameter "y value". Both input images (field 2D 4-vector byte) must be the same size.

secno = ount0

MAPPER MODULES

International AVS Center

2D_bounds

Author :	**KGT (Kubota Graphics Technologies) Inc.**
Number :	1876
Language :	CandFORTRAN
Submitted :	09/06/95
Version :	1.000
Last Updated :	09/06/95
Ported To :	DEC_alpha

The "2D bounds" module generates lines that indicate the bounding box of 2D field. It is used to get the 2D mesh of 3D field. For example, you can use this module to check CFD grid.

3D_contour_map

Author :	**Krzysztof (Chris) S. Nowinski** Institute of Applied Mathematics and Mechanics, Warsaw University
Number :	1781
Language :	C
Submitted :	02/15/94
Version :	1.000
Last Updated :	02/15/94
Ported To :	HP Sun

The 3D contour map routine creates contour map of a given 2D scalar float field together (optionally) with a surface equivalent to the surface generated by the field to mesh routine. The algorithm used is a subset of the more familiar 3d marching cubes algorithm. This is a modification of the field_to_con module by Wes Bethel.

Analog_Clock

Author :	**Lars M. Bishop** National Center for Comp. Elect., Univ. of Illinois
Number :	1626
Language :	C
Submitted :	07/02/93
Version :	1.000
Last Updated :	07/02/93
Ported To :	DEC IBM Kubota

The Analog Clock module is a 3-dimensional, graphical readout that presents a "stopwatch" or "piechart" dial display. It is designed to be connected to an integer parameter whose value is changing over time. The Analog Clock provides a qualitative way of seeing the value change. By setting the cyle length of the Analog Clock intelligently, the input parameter's change over time can be elucidated. For example, if the cycle length is set to 100, and the Max Value is set to 1000, the fast indicator will cycle for every 100, and the slow indicator will cycle at one tenth this speed. The Analog Clock produces two types of "clock" Pie Chart - This counter uses a "cheese wheel"-shaped clock as its fast indicator, and a change in color of the cheese wheel as its slow indicator. In this version, the colors wrap around past each multiple of Max Value. Analog Clock - This counter uses a standard two-handed analog clock face (without markings) as its fast and slow indicators.

CFD_object

Author :	**KGT (Kubota Graphics Technologies) Inc.**
Number :	1877
Language :	FORTRAN
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_alpha

The "CFD object" module inputs a scalar field and creates a geometry(Non-fluid Area) for CFD data. This module tells which digit (0 or 1) is in the node as flag data. The flag data of the nodes is as follows. 1 means the fluid node and 0 is the non-fluid node.

CFD_object_conto

Author :	**KGT (Kubota Graphics Technologies) Inc.**
Number :	1878
Language :	FORTRAN
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_alpha

The "CFD object contour" module inputs 2-vector field and creates a contour surface(on obstacle) for CFD data. This module tells which digit (0 or 1) is in the node as flag data. The first component data (of 2-vector) is the scalar value for color, and the second component is the flag data. The flag data is as follows. 1 means the fluid node and 0 is the non-fluid node.

CFD_objectline

Author :	**KGT (Kubota Graphics Technologies) Inc.**
Number :	1879
Language :	FORTRAN
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_alpha

The "CFD objectline" module inputs a scalar field and creates a line geometry (obstacle) for CFD data. This module tells which digit (0 or 1) in the node as flag data. The flag data is as follows. 1 means the fluid node and 0 is the non-fluid node.

ChairPlot_2D_Txt

Author :	**Scott Goodyear (AVS Inc.), Ian Curington (AVS Inc.), and Mike French (E and S)**
Number :	1641
Language :	C
Submitted :	09/24/93
Version :	1.000
Last Updated :	09/24/93
Ported To :	HP IBM Sun

The ChairPlot module is another alternative way of visualizing 3D volume data. The volume is displayed with 3 orthogonal slice planes, with the exterior boundary filled in behind, so it looks like a section has been removed from a corner of the volume. It is similar to excavate brick, except that it makes 2D polygons, with UV texture coordinates and resamples the input data into 2D images that glue onto the slice sections. This way, 2D texture map facilities are used, rather than 3D texture. Several hardware vendors, notably EandS, SGI, Kubota, Vistra, etc., have hardware support for this type of texture mapping. The software renderer also has support for this mode, where no accelerator is available. The controls are virtually identical to excavate brick.

Colored_bounds

Author :	**D. LE CORRE**
	CONVEX S.A
Number :	1200
Language :	C
Submitted :	05/08/92
Version :	2.000
Last Updated :	08/30/92
Ported To :	DEC IBM Kubota HP

This mapper module generates a geometry consisting of 6 orthogonal slices (2 in each dimension) from a 3D scalar field. By default, the 6 ortho slices are placed at the hull of the input field. The faces of this geometry are colored from the scalar field value.

Contour_Lines

Author :	**KGT (Kubota Graphics Technologies) Inc.**
Number :	1880
Language :	CandFORTRAN
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_alpha

The “Contour Lines” module inputs a 2D field and colormap. This module looks like the “contour to geom” module. It finds and creates 2D contour lines of similar value.

Contour_Mesh

Author :	**KGT (Kubota Graphics Technologies) Inc.**
Number :	1881
Language :	FORTRAN
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95

The “Contour Mesh” module inputs a 2D field and colormap. This module looks like the “field to mesh” module. This module creates the contour only Fluid AREA (of CFD).

Draw_Text

Author :	**Ronald E. Stogner, Computer Sciences Corporation**
Number :	1736
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

Creates a text geometry object from an Label String input. A user can also supply an optional Hour as an integer, which will be added into the label. This is useful if you are animating a time series of data. This is one of a suite of modules we used to plot the emission data for the state of North Carolina. This group of modules allows displaying points, lines and polygons over a map of the United States. The module suite is stored as a group in a compressed tar file at the International AVS Center, with some sample networks and datasets included. You may find it easier to access the modules this way. The compressed tar file is approximately 1.2MB in size, and can be found via anonymous ftp to avs.ncsc.org, in /sample_data/avs_data/Wang_GIS_mods s.tar.Z.

Field_To_Lines

Author :	**Jeff Wang** MCNC, Environmetal Programs Group
Number :	1737
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This module allows users to draw lines in the geometry viewer. The value of the line is mapped into color by the colormap of AVS. The input field is a 1D 6 space field with (x0, y0, z0, x1, y1, z1, value). The denomitor "0" means the starting points of the links, while "1" means the ending points of the links. This is one of a suite of modules we used to plot the emission data for the state of North Carolina. This group of modules allows displaying points, lines and polygons over a map of the United States. The module suite is stored as a group in a compressed tar file at the International AVS Center, with some sample networks and datasets included. You may find it easier to access the modules this way. The compressed tar file is approximately 1.2MB in size, and can be found via anonymous ftp to avs.ncsc.org, in /sample_data/avs_data/Wang_GIS_mods s.tar.Z.

Field_To_Points

Author :	**Jeff Wang**
	MCNC, Environmetal Programs Group
Number :	1739
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This module allows users to draw points in the geometry viewer. The value of the point is mapped into color by the colormap of AVS. The input field is a 1D 3 space field with (x, y, z, value). This is one of a suite of modules we used to plot the emission data for the state of North Carolina. This group of modules allows displaying points, lines and polygons over a map of the United States. The module suite is stored as a group in a compressed tar file at the International AVS Center, with some sample networks and datasets included. You may find it easier to access the modules this way. The compressed tar file is approximately 1.2MB in size, and can be found via anonymous ftp to avs.ncsc.org, in /sample_data/avs_data/Wang_GIS_mods s.tar.Z.

Field_To_Polygon

Author :	**Jeff Wang** MCNC, Environmetal Programs Group
Number :	1738
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This module allows users to draw shaded polygons in the geometry viewer. The value of the polygo is mapped into color by the colormap of AVS. The input field is a 1D 4 space field with (x, y, z, n, value), where (x, y, z) is the coordinate of a vertex in the polygon, n is the number of vertices in the polygon, value is the value assign to the polygon. This is one of a suite of modules we used to plot the emission data for the state of North Carolina. This group of modules allows displaying points, lines and polygons over a map of the United States. The module suite is stored as a group in a compressed tar file at the International AVS Center, with some sample networks and datasets included. You may find it easier to access the modules this way. The compressed tar file is approximately 1.2MB in size, and can be found via anonymous ftp to avs.ncsc.org, in /sample_data/avs_data/Wang_GIS_mods s.tar.Z.

Field_grid

Author :	**Penny Rheingans** US EPA Scientific Visualization Center
Number :	1307
Language :	C
Submitted :	08/30/92
Version :	1.000
Last Updated :	08/30/92
Ported To :	DEC Sun HP

Generates a labelled grid and/or axes for field data. The extents of the grid or axes are the extents of the data (or slightly larger).

Field_to_Isosurf

Author :	**Chin-I Huang** National Chiao Tung University, Taiwan, Republic of China
Number :	1630
Language :	C
Submitted :	07/02/93
Version :	1.000
Last Updated :	07/02/93
Ported To :	HP IBM DEC Sun

This module reads a uniform 3D volume data and passes a threshold level to it to generate isosurfaces. The algorithm is based on the Marching Cubes Method from Lorensen. Any interesting threshold level ranged from 0 to 255 can be input and the module will generate isosurfaces corresponding to that level. After the generation of isosurfaces, you can see the number counts of each table case on the console. Colormap is used for rendering the volume with one color level to a threshold. The module is limited to the data type of char, but the reader can change it into the other data type easily by modifying the source code a little.

GRAPH2D

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1800
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

GRAPH2D – generate an X-Y plot, control graph viewer parameters, and optionally output PostScript. The GRAPH2D module is an entire graphics management system for 2d plots. It creates a network including the AVS "graph viewer" and "image to postscript" modules, and it generates script files that are played to make ASCII files required as input to the graph viewer. Up to four 2d windows may be controlled by the module, and the settings will apply to the current window of focus, selected by the user and indicated by the red color band around the window. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

Get_Local_IP_Num

Author :	**Jeff Wang** MCNC, Environmetal Programs Group
Number :	1722
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	DEC IBM Kubota

This module allows users to get the internet IP number for their local machine. This module is useful when it is connected with an editor in the remote machine. See the related module Edit_File.

GraphSound

Author :	**Brian Kaplan** Center for Innovative Computer Applications
Number :	1636
Language :	C
Submitted :	07/05/93
Version :	1.000
Last Updated :	07/05/93

The GraphSound module generates a graph of the waveform of a sound produced by the Sonify module and is useful as a diagnostic tool.

Grid_2_TxtMesh

Author :	**Advanced Visual Systems, Inc.**
Number :	1834
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	HP IBM Sun

The texture mesh module serves two needs. First, the need for a robust module for mapping 2D data to quadrilateral meshes is served. The input 2D data may be a 2D uniform field, in which case the x,y coordinates for the output mesh is derived from the extents information (if present) or using a mapping from indeces to space if no extents information is present. The z values are obtained from the data in the field. The second need fulfilled by this module is support for texture mapping. Two types are supported. The first, called "dynamic" in AVS parlance, is the traditional type of texture mapping where the coloring of the surface is postponed until rendering time. The second type, called "explicit" here will color the vertices of the field based upon the image on the second input port, along with the uv coordinate ranges specified by the user.

Isoband_Slice

Author :	**Ian Curington (AVS Inc.), and Mike French (E and S)**
Number :	1749
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Sun

A set of contour bands are drawn on a 2D slice through a 3D field. The field input must be a slice through a uniform or irregular 3D grid. If the field is uniform, the plane of the output contour geometry will have its correct position in 3D space.

Isoband_UCD

Author :	**Ian Curington (AVS Inc.), and Mike French (E and S)**
Number :	1760
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

A set of contour bands are drawn on the exterior faces of a UCD. The UCD must have a single scalar component of node data. To extract a single scalar component of node data from a composite ucd use the UCD EXTRACT module. There is no module available to extract a scalar from a vector component of node data.

Isoline_Slice

Author :	**Mike French** Evans and Sutherland UK
Number :	1750
Language :	FORTRAN
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota

A set of contour levels are drawn on a slice through a field. The field input must be a slice through a uniform or irregular 3D grid. If the field is uniform, the plane of the output contour geometry will have its correct position in 3D space.

Isoline_UCD

Author :	**Ian Curington (AVS Inc.), and Mike French (E and S)**
Number :	1765
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota

A set of contour lines are drawn on the exterior faces of a UCD. The UCD must have a single scalar component of node data. To extract a single scalar component of node data from a composite ucd use the UCD EXTRACT module. There is no module available to extract a scalar component from a vector component.

Keyframe_Ani

Author :	**Brian Kaplan** Center for Innovative Computer Applications
Number :	1133
Language :	C
Submitted :	02/23/92
Version :	3.000
Last Updated :	07/05/93
Ported To :	Kubota

The Keyframe Animator module is used to animate objects which are displayed through the Geometry Viewer subsytem. It works by supplying a tranformation matrix to the Geometry Viewer module for any object which has been loaded into the Geometry Viewer or supplied to the Geometry Viewer's geom input. Generally the Keyframe Animator's geometry output (the transformation matrix) is supplied to the geometry input of the Geometry Viewer module. This is an update from version 2.4 to 3.0 Beta.

LABELS_UNITS

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1802
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

LABELS UNITS – create a label based upon the input field and input slice axis and output a label geometry. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

LBL_bezier_v

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1016
Language :	C
Submitted :	10/10/91
Version :	3.000
Last Updated :	05/19/92
Ported To :	IBM Sun

The bez_vol module is an object order volume renderer designed to work with volumes embedded in warped grids. This module assumes that voxels are topologically equivalent to hexahedra, the faces of which are not necessarily planar. This solid is manipulated and rendered using tricubic Bezier basis and rendered using a forward difference evaluation of the cubic triparametric Bezier representation of a solid.

MASK_TILE

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1803
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

MASK_TILE is a mapper module that produces a geometry from an input field and color map. If any data point falls below the color scale, this sets its color to the mask color. For optimization of Z slices, adjacent squares on the same row of identical color will be drawn as a single polygon of that color. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

MINMAX_REPORT

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1805
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

NAME MINMAX REPORT – generate statistics report with C-language. All floating-point values are written with the C-language "%g" format so that more decimals may be reported than those using the standard AVS statistics module. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

Marker_Draw

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1836
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	HP IBM Kubota Sun

marker_draw - generate marker bitmaps from marker input list. This module to takes a marker coordinate specification vector XY-pair list on an input port from another module, combines this with an an input image, and create a composite display image. The input image is copied to the output, then marker polygonal shapes are scan converted and filled using an edge fill algoriythm, with the fill color computed based on hue parameters. It is designed and tested with several other marker modules, such as marker_pick or marker_select as source The vector marker list is defined by structures in the "marker.h" header file, defining the structure and usage of the field data type for markers. Each marker has a compile-time defined shape, but runtime attributes of position, size, color, etc.

Marker_Gen

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1837
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP IBM Kubota Sun

marker_gen - generate marker bitmaps from marker input list. This module converts vector edge defined marker shapes into ROI bit masks coverage for AVS images. The input is compatable with any AVS image, and any fully specified marker list 1D integer field. The marker list can be specified from a stored field, or more often, marker_select, or marker_pick modules. This module scan converts the marker edge list into a binary bit pattern, where the bit pattern at each pixel is either zero for no marker present, or the marker shape index. The output is a full size scalar byte field, the same size as the input image.

Marker_Merge

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1838
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP IBM Kubota Sun

marker_merge - merge roi marker pattern with full colour image. This module modifies a full color input image by masking it with an ROI byte field. The marker type code in the ROI is used to control what color is replaced inside each marker area, and optionally a secondary texture image can be used as the fill color inside of the marker areas. The ROI field is compatable with other ROI field modules (IP Lib). This module is designed to work with other marker modules, marker_gen, marker_select, marker_draw, marker_pick, etc.

Marker_Pick

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1839
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	HP IBM Sun

Marker_Pick - Pick, Edit, Create, and Move Markers on Images. Allow marker pick identification and display from an input marker list and the image viewer system. Takes as input an AVS field with a list of coordinates for markers, displays them over an image (map) in the image viewer, and reacts to mouse picks on these markers, sending the pick selection to the output port. Markers are defined by geometric polyline vector lists, not ROI pixel masks. You can externally control position, shape, size, etc. Any marker field sent out on the output port can be read back in to the input field port. This way state save and restore are possible. New marker lists can be made using the “marker_pick” module.

Marker_Select

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1840
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	HP IBM Sun

Marker_Select - Define and allow Pick Selection of Markers Allow marker pick identification and display from an input marker list and the image viewer system. Takes as input an AVS field with a list of coordinates for markers, displays them over an image (map) in the image viewer, and reacts to mouse picks on these markers, sending the pick selection to the output port. Markers are defined by geometric polyline vector lists, not ROI pixel masks. New marker lists can be made using the "marker_pick" module. This module does not allow edition of position or shape, just allows query picking of pre-defined markers.

NEW_CROP

Author :	**Kathy Pearson**
	MCNC Environmental Programs
Number :	1807
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

NAME NEW CROP – crop 3d or 4d field to reduce the range of any of its dimensions. Unlike the standard AVS crop module, the NEW CROP module allows the user to crop a field down to a single grid cell in any dimension. Also unlike the standard crop module, the NEW CROP module values will "hold" between invocations if the previous settings are within the latest range for X, Y, Z, or Time. The standard crop module allows cropping on 3 dimensions only, but the NEW CROP module allows the user to crop either 3D or 4D fields.

NEW_LEGEND

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1811
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

NEW LEGEND – create a geometry of a linear scale using color polygons. The NEW LEGEND module creates a color scale bar of linear bins in either the horizontal or vertical with labeling controlled by C-language formats to give the user more control than that found in the standard AVS "color legend" module. Color bins are selected from the input color map based upon linear increments corresponding to the number of color bins selected. For a typical Newton AVS colormap, the lowest bin will be blue and the highest bin will be red. The bins are labeled according to the Data Minimum and Data Maximum values and the linear increments between. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

NEW_PRINTFIELD

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1808
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Sun

PRINTFIELD – display data in a matrix format. The NEW PRINTFIELD module outputs to a text browser (and optionally to an output file) the values of the input cropped slice of data. The module also requires that an input string specifying that the input data has been sliced along the "X", "Y", or "Z" axis. Each output value is written in the C-language format set by the user with optional labeling. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

NEW_SLICE

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1809
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

NEW SLICE – slice a field of up to 4 dimensions on X, Y, Z or time. Unlike the standard AVS slice module, the NEW SLICE module will accept 4D data input in addition to a 3D data field. It outputs a field of one dimension less than that input with the same coordinate space and the same number of vectors as the input. The NEW SLICE module also outputs the slice axis to indicate to downstream modules whether the slice has occurred in the "X", "Y", "Z", to "Time" dimension. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

NEW_TILE

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1810
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

NEW TILE – create a geometry of a tile plot displaying 2D 3-space data. The NEW TILE module accepts as input a field sliced on "X", "Y", or "Z" and displays a tile map in the other two dimensions. The NEW TILE module was needed because versions available from the International AVS Center do not work properly for cropped data and do not function well with any slice axis other than "Z". Still, much credit is due to David Bock, formerly of MCNC, who is the author of the original, for his pioneering efforts in creating an early AVS version of this module. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

Ortho_Texture

Author :	**Scott Goodyear and Ian Curington** Advanced Visual Systems, Inc.
Number :	1675
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

This is a special version of field to mesh, for uniform fields, that outputs just one quadrilateral, with texture from the field, rather than a pair of vertex colored triangles for each sample point. This allows exploitation of hardware texture mapping. This is prototype unsupported code. The source is NOT intended as a working example.

Polygon_Volume

Author :	**Jeff Wang** MCNC, Environmetal Programs Group
Number :	1731
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Sun

This module allows users to draw columns of the irregular shaded polygon. For example, users can make a 2 dimensional geographical county map become 3 dimensional volume using the population of the county as height. Another application includes building a house, showing downtown area of a city and plotting economic data. The value of the polygo is mapped into color by the colormap of AVS. The input field is a 1D 4 space field with (x,y,z,n,value), where (x, y, z) is the coordinate of a vertex in the polygon, n is the number of vertices in the polygon, value is the value assign to the polygon. Portions of the code based on Wes Bethel's hsv_to_rgb module, also on the IAC's anonymous ftp site.

Polyline

Author :	**Ian Curington** Advanced Visual Systems
Number :	1678
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

The polyline module interacts with the AVS5 image viewer, to allow interactive line drawing on top of an image. Points inside the image can be picked, and line segments are drawn with "rubber banding" along the pick sequence. "Undo" will back up the line segments, and clear will erase all lines. If multiple image viewer "views" are open, the rubber banding can be seen in each window at the same time. The resulting picked polyline list is output on a scatter field port, for external processing by other modules.

QUERY_TILE

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1812
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

QUERY TILE – create a geometry of a cross hair cursor over a tile plot and report the value of the point at that tile position.

RESET_2

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1798
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

RESET – resets geometry viewer to normalize and scale window to the current coordinate system, then rescales to reduce its size. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

Reset_Geom

Author :	**Jeff Wang** MCNC Environmental Programs Group
Number :	1740
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This Module allows users to reset, rescale and normalize the geometry viewer. This is one of a suite of modules we used to plot the emission data for the state of North Carolina. This group of modules allows displaying points, lines and polygons over a map of the United States. The module suite is stored as a group in a compressed tar file at the International AVS Center, with some sample networks and datasets included. You may find it easier to access the modules this way. The compressed tar file is approximately 1.2MB in size, and can be found via anonymous ftp to avs.ncsc.org, in /sample_data/avs_data/Wang_GIS_mods s.tar.Z.

Scatter_Rect

Author :	**Ian Curington** Advanced Visual Systems
Number :	1846
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP Kubota Sun

scatter rect - Draw Nested Rectangles from irregular 1D (scatter) fields. This module draws 4 variable height nested coloured rectangles from scatter input data showing 4-vector data at the specified positions. Each sample point gets 4 stacked rectangles. The first three are various colors, and fourth is a wire frame shell around the others. It can be used in place of "scatter dots". The Scale dial is simply a size multiplier. The height of each nested rectangle is controlled with one of the vector quantities.

TEXT_TO_GEOM

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1815
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

NAME TEXT TO GEOM – create a label geometry from the input string. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

TIME_SPAN_TEXT

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1816
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Sun

TIME SPAN TEXT – create a string containing the starting and ending timestamp labels; if they are the same, output only the starting timestamp label The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

Title

Author :	**Roy Wirthlin** Advanced Visual Systems Inc
Number :	1287
Language :	C
Submitted :	06/23/92
Version :	1.000
Last Updated :	06/23/92
Ported To :	DEC IBM Sun Kubota HP

This module creates a “label” style text string in GEOM format which can be used as a title. There are two advantages to using this module over the labelling facilities in the Geom Viewer - (1) the information will get saved with a network and (2) the optional floating point parameter (“Value”) can come from another module and can represent some important variable such as time, animation step, some parameter, etc. The Title String can contain a '%f' ala C programs to include this parameter - for example, the Title String can be “Time Step %f” and the value of “Value” will get transferred to the geom label. Other things you can control are the font number, drop shadows, justification, position, height, and color of the text string.

UCD_Node_Contour

Author :	**Woody Muller** Becton Dickinson Research Center
Number :	1830
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP Sun

UCD Node Contour creates an array of floating point RGB values, one color for each node in an unstructured cell data set. This module is a replacement for ucd-contour as supplied by AVS Inc. AVS Inc's ucd-contour module turns zones of out-of-range data black, rather than clamping it to min and max values as the user often prefers. This module is intended as a functional equivalent of ucd-contour, but which clamps out of range values to min and max colors. This allows more detailed coloring of data within the absolute range of the entire UCD without turning any of it black. UCD Node Contour is used to create a color contour of a UCD structure. Its output is passed to ucd-to-geom to produce a colored representation of a UCD structure. UCD Node Contour associates a color with each node of a UCD structure.

UCD_grid

Author : **Penny Rheingans**
US EPA Scientific Visualization Center
Number : 1308
Language : C
Submitted : 08/30/92
Version : 1.000
Last Updated : 08/30/92
Ported To : DEC Sun HP

Generates a labelled grid and/or axes for ucd data. The extents of the grid or axes are the extents of the data (or slightly larger).

VIEW_SHARE

Author : **Terry Myerson**
International AVS Center (NCSC)
Number : 1206
Language : C
Submitted : 05/12/92
Version : 1.000
Last Updated : 05/12/92
Ported To : Sun DEC IBM

This module is part of the collaboratory development being done at the North Carolina Supercomputing Center. This module allows AVS users, on up to 9 heterogeneous workstations, to perform shared visualization. In other words, the data can be located on one workstation's file system, the computation could be done on that user's workstation, and then the resulting geometry can be broadcast to all of the other users and then rendered on each user's individual workstation. Each user can then take control of the visualization, rotating and translating not only his view, but that of the other collaborators. In addition, integers, floats, and strings can be broadcast around the collaborative loop to allow for interactive distributed data slicing, contouring, and file switching.

Vector_Arrow

Author : **KGT (Kubota Graphics Technologies) Inc.**
Number : 1883
Language : C
Submitted : 09/07/95
Version : 1.000
Last Updated : 09/07/95
Ported To : DEC_ultrix DEC_alpha

The "Vector Arrow" module inputs a 3-vector field (1D,2D,3D) and creates arrows. This module looks like the "hedgehogs" module.

WIND_PLOT

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1818
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

The WIND PLOT module accepts as input a field 3-vector wind field and outputs a geometry representation of it. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

Wide_Arrow

Author :	**Wright, E.L.** U.S Geological Survey
Number :	1567
Language :	C
Submitted :	01/28/93
Version :	1.000
Last Updated :	01/28/93

This module reads a 1D 3-coordinate irregular field containing a single floating-point 2-element vector specifying X and Y velocity components at the specified (X,Y,Z) coordinate. It then creates the geometry to display a wide arrow indicating vector direction and magnitude at a specified (X,Y,Z) coordinate. The length of the wide arrow is proportional to the vector magnitude. The Scale Factor parameter allows the user to scale the data to produce an arrow of the desired length. This module is useful for displaying wind direction and velocity.

add_polar_2D

Author :	**Jeff Knerr and Shawn Mehan** North Carolina Supercomputing Center
Number :	1305
Language :	C
Submitted :	08/26/92
Version :	1.000
Last Updated :	08/26/92
Ported To :	DEC Sun HP

This module takes in a 2D field and adds coordinate data in (r,theta) spherical polar coordinates mapped to (x,y) cartesian as required by the geom_viewer. The added coordinates are either uniformly spaced or calculated from input coordinates, depending on whether the input field is uniform or rectilinear. Then, depending on what the original computational space grid was, the data can be mapped so that it covers the entire 2*pi grid.

area

Author :	**Anonymous** Stardent Computer
Number :	1013
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota IBM Sun HP

This module takes an input 2d scalar field, and assumes it has a “doughnut” or “0” shape in it. The simple evaluation algorithm searches for the interior space of the “0”, and computes the cross sectional area of the region. The result (a scalar) is passed in a 1-D output array. This module will “accumulate”, and pass the history along with the current value. The parameter input controls the current slice, ZERO will reset things. The design was to work with “tubes” with variable cross section. Typical input would come from 3D scalar fields -¿ ortho slice, while typical output would go to the “graph viewer.” This module was written to show how to compute the transverse cross sectional area of a blood vessel in a medical imaging application.

bin_field

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1178
Language :	C
Submitted :	04/20/92
Version :	2.000
Last Updated :	05/19/92
Ported To :	Kubota DEC IBM Sun

A grid is constructed in three-space which represents the extents of some input 3-space field. The grid has a user specified number of coordinates, or bins, along each axis. For every input point, the count in the three-space bin in which it lies is incremented.

bivar

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1017
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

The bivar module will resample 1d scatter data in 2 variables. A typical use for this module is to take a field of 2-d coordinate, scalar values, and resample over a "regular" grid. The algorithm used is one in which, for each output grid location, all input points are scanned to determine if they lie within the specified search radius. If so, that input point's contribution is weighted as a function of distance from the output grid point and scaled by 1./distance**weightfunction. As such, this module is best suited for use on input data which desn't necessarily fit a differentiable function. For such data, refer to the sact2d module.

carpet

Author :	**Ian Curington** Advanced Visual Systems
Number :	1569
Language :	C
Submitted :	01/28/93
Version :	3.000
Last Updated :	12/15/93
Ported To :	IBM Kubota HP

A Carpet Plot surface is made showing surface height variation from one field, with color variations from data in a second field. Each cell is drawn as per field to mesh, (linear interpolation), or in block mode with descrete non-interpolated colors. In block mode, it will optionally leave holes in the carpet where the data goes out of range for the colormap, allowing mask values to control cell visibility, or region boundaries. Version 3.000 adds upgrade to irregular 3space fields for a better hole algorithm.

city_scape

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1185
Language :	C
Submitted :	05/04/92
Version :	3.000
Last Updated :	07/15/94
Ported To :	IBM Sun Kubota HP

The city scape module converts a two-dimensional floating point field into group of 3D blocks, represented as a GEOM-format mesh. Each element of the field is mapped to a "building". The height of each block above each point is proportional to the scalar value of the field. This is my attempt at "Business Graphics", for a multi-dimensional telecommunications statistics data application, except in interactive 3D. Version 2 is a substantial upgrage from version 1 at the IAC. This module provides 3D block diagram, with averaged statistics on side shadow panels, and annotation. A transparent theshold sheet is superimposed on city. Blocks can be picked for interogation of cell number. parts may be turned on or off for clarity. Two scalar fields can be displayed, one contolling height, the other color, with two independent side stacked side bar statistics. Version 3 update draws the top face of picked block in white hi-lite, adds extra edit list for pick, no normals, new text input for edge labels, added string typeins for edge labels, and other fixes.

color_legend

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1019
Language :	C
Submitted :	10/10/91
Version :	2.000
Last Updated :	08/31/92
Ported To :	Kubota DEC IBM Sun HP

This module takes an input colormap and generates a "color legend," consisting of a number of geometry objects which may be subsequently rendered using the geometry viewer module. The range of colors in the color scale is controlled by the input colormap. All colors in the colormap are represented in the output color scale. In addition to the smoothly shaded "color bar", the user may control various parameters relating to labeling characterisitics, and legend orientation. Version 2 fixes (by Dr. John Tee, AVS Inc consultant) make the code more portable by using standard AVS routine calls.

color_tile

Author :	**David Bock** North Carolina Supercomputing Center
Number :	1151
Language :	C
Submitted :	03/11/92
Version :	2.000
Last Updated :	05/19/92
Ported To :	DEC IBM Sun Kubota HP

The color tile module accepts a 2D scalar field of either uniform or rectilinear coordinates of any data type and an input colormap and generates a geometric grid of colored cells. The resulting grid is generated by creating cells of 4 vertices SURROUNDING each data value rather than cells with the data values representing the ENDPOINTS of each cell as is done in other color mapping modules such as field to mesh. The data values within each cell are mapped to colors determined from the input colormap and the input number of steps. This color is used to color the entire cell of the respective data value. By selecting the min/max parameter, the color tile module will calculate the minimum and maximum data values from the input field and use these values to map the data to the color values. Without this selection, the module accepts type-in values for the low and high data values. By selecting the mesh parameter, the module generates and displays an output mesh overlaid upon the output color tile object.

cone_bounding

Author :	**Massimiliano Bavelloni** Radioastronomy department C.N.R. Bologna Italy.
Number :	1896
Language :	C
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_ultrix DEC_alpha

Draw geometric cone around 3D scattered points in order to display astronomical points (galaxies). Galaxy coordinates have been already converted from Spherical Coordinates into Cartesian Coordinates. In the computational section coordinates are converted into Spherical ones and then converted again in Cartesian coordinates. The cone is always centered in the origin (x = 0, y = 0, z = 0) point.

confocal_slicer

Author :	**Nick Salmon** European Molecular Biology Laboratory
Number :	1851
Language :	C
Submitted :	10/17/94
Version :	1.000
Last Updated :	10/17/94
Ported To :	DEC_alpha HP IBM Sun

Slices through a three dimensional field, with the slice plane at an arbitrary orientation. The confocal slicer module extracts a 2d slice from a 3d field. Output consists of a 2d field or a geometry. The position, size and resolution of the slice plane are under user control via the parameters. For all input fields, the module will determine "sane" parameter settings for extracting an initial slice. This is basically a hack from the module new_arbitrary slicer from IAC (apologies to Wes Bethel). The intention is to use it for slicing data collected from a confocal microscope. This is uniform byte data with one or more values per point in space.

contour_field

Author :	**Phil McDonald**
	NOAA/ERL/Forecast Systems Laboratory
Number :	1193
Language :	C
Submitted :	05/06/92
Version :	3.000
Last Updated :	07/05/93
Ported To :	IBM

The contour field module generates contour lines from a 2D field. These contour lines are output as a geometry object with one or more polylines in 3D space and will nest neatly with a mesh surface made from the same field. The contours may be at uniform intervals or at arbitrary levels. If an optional colormap is attached, the contour lines will be colored according to how the values of the contour lines are mapped in the colormap, otherwise, the contour lines are not colored. The contour lines may be optionally labelled. The interval between labelled contour lines may be uniform or, if arbitrary contour levels have been specified, only specified levels will be labelled. A separate geometry object for these labels is output. This allows the properties of the labels to be entirely different from those of the contour lines. However, the labels can be colored by value only if the contour lines are colored by value. Version 2 of this module includes bug fixes courtesy of Lyle Wiedeman of UC Irvine. Version 3 includes several changes to the module and the libraries it uses as well.

contour_to_g

Author :	**Anonymous**
	Stardent Computer
Number :	1022
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

CONTOUR TO GEOM finds and creates contour lines of similar value in a scalar byte field. The parameter "threshold" controls the level of contour being found. It differs from teh CONTOUR module found in the FILTERS directory in that it produces disjoint lines which can be rendered with the geometry viewer. It can also handle 3D data sets and irregular grids.

contour_to_posts

Author :	**Rob Kuyper** University of Technology Delft, The Netherlands
Number :	1586
Language :	C
Submitted :	02/11/93
Version :	1.000
Last Updated :	02/11/93
Ported To :	Kubota HP Sun

"contour to postscript" is the ultimate contour module. All the things you expect of a contour module are built in this one. Automatic scaling, min/max scaling, typein levels. Colormap port to color the contours. Postscript output for contours - straight lines on the plot . If the dataset is positioned in e.g. the y-z plane, only the y-z coordinates are used in the Postscript output. After compilation the world of interactive contour and Postscript generation opens up to you.

crop_and_slice

Author :	**James Pipe**
	University of Michigan
	Medical School
Number :	1610
Language :	FORTRAN
Submitted :	04/10/93
Version :	1.000
Last Updated :	04/10/93
Ported To :	DEC IBM

This module combines the functionality of series of the supported 'orthogonal slicer' and 'crop' modules and with some additional capabilities. For each dimension in the input data set (up to 4 dimensions), one may crop (preserve that dimension but limit its range), or slice (eliminate that dimension by choosing data from one index into that dimension) the data along that dimension. If the data is cropped along some dimension, one can specify the range either with 'begin' and 'end' indices, or by using 'center' and 'width'. Changing the value in one pair (begin/end or center/width) causes the module to the values of the other pair. If the data is sliced along any dimension, only the 'dimension center' widget remains visible; this is similar to the 'slice plane' parameter in the supported 'orthogonal slicer' module. For every dimension that is sliced, the dimensionality of the output data is reduced by 1. For instance, a 4D input data set that is sliced along dimensions 2 and 4 results in a two-dimensional output data set. The cropped/sliced data set is present at the LEFT output port. The RIGHT output port contains a 'mask' data set which is identical to the input data set EXCEPT for the data which was cropped/sliced into the left output port data set; this part of the mask (right port) data is set to zero, so that one can visualize the location in the original data set from which the cropped/sliced data came. For example, if the input is a 2D image, and dimension 2 is sliced, then the left output port will be a 1D 4-vector byte, and the right output port will be the same as the original image except for a vertical black line running through it where dimension 2 was sliced (plus perhaps some cropping along dimension 1).

draw_tracks

Author :	**John Tee**
	AVS Inc. Consultant
Number :	1650
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

DRAW TRACKS is a module provided to demonstrate the use of the geometry viewer to display two-dimensional track information provided as a floating point 2-vector (x,y) input field. If the input field has more than one dimension, multiple tracks are displayed. If the field is 3-vector, it uses the third vector component to color the lines, with the optional colormap input field. The hi-low dials of the colormap control the line color range. DRAW TRACKS also allows the user to select a track of interest by clicking the mouse on it. The selected track is highlighted by changing its colour, and the selected vertex is indicated by positioning a sphere over the vertex, and a line connecting the vertex to the sphere. The coordinates of the selected vertex are displayed in a text window.

field_lines

Author :	**Rob Kuyper**
	Univeristy of Technology
	Delft, The Netherlands
Number :	1533
Language :	C
Submitted :	10/01/92
Version :	1.000
Last Updated :	10/01/92
Ported To :	HP Sun

field lines is a mapper module which converts a 2D field in an object of lines and planes. This module will create black lines (in optionally i- or j-direction) with a 'no-lighting' plane below it, so the field will look like a serie of non-transparent graphs.

field_o_matic

Author :	**Phil McDonald** NOAA/ERL/Forecast Systems Laboratory
Number :	1265
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	07/05/93
Ported To :	IBM

Field o matic.. It slices.. It dices.. Actually, it slices and pares. The field o matic module provides the same capabilities as the orthogonal slicer, crop, and field to mesh modules. A 3D field may be sliced orthogonally in computational space and this slice may be cropped. The resulting slice "zone" (window) is output as a 2D field and as a mesh surface. If an optional colormap is connected, the mesh surface will be colored by this colormap, unless the field is a vector field, in which case the mesh will not be colored. The updated version includes several changes to the module and the libraries it uses as well.

field_params

Author :	**John Tee** AVS Inc. Consultant
Number :	1651
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

FIELD PARAMS is a module provided for the Oxford University Osney Labs custom software for visualisation of experimental data. FIELD PARAMS returns the X and Y dimensions of a field as integer parameters, and the minimum and maximum values of the field as float parameters. It allows each of these parameters to be scaled by its own scale factor. FIELD PARAMS is used to provide the GENERATE COLORMAP module with the range of the data values (cf COLOR RANGE), while also allowing the user to modify the colour range. It also provides the X dimension ofthe field as an upper limit to the ANIMATE INT module which controls the position of the viewable window.

field_size

Author :	**John Tee**
	AVS Inc. Consultant
Number :	1652
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

FIELD SIZE returns the dimensions of a field as integer parameters. It allows each of these parameters to be scaled by its own scale factor. FIELD SIZE is used to limit the range of parameters such as cropping limits to a proportion of the size of a field.

field_stats

Author :	**John Tee**
	AVS Inc. Consultant
Number :	1653
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

FIELD STATS returns the minimum, maximum and mean values of a field as floating point parameters. It allows each of these parameters to be scaled by its own scale factor. One application of FIELD STATS is to create a colormap which is adapted to the range of values in the field (as COLOR RANGE does), but allowing the user to adjust the range manually.

field_to_con

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1024
Language :	C
Submitted :	10/10/91
Version :	2.000
Last Updated :	02/15/94
Ported To :	IBM Sun HP

The field to contour module creates colored contour lines around regions in which the threshold specified by the "level" parameter is crossed. The module can process only 2d datasets, but these may may exist in 3-space (if the field is irregular) or in 2-space (rectilinear or uniform). The algorithm used is a subset of teh more familiar 3d marching cubes algorithm. Version 2 update provided by Krzysztof (Chris) S. Nowinski of Warsaw University's Institute of Applied Mathematics and Mechanics. Enhancements include Generating contours as polylines and not disjoint lines for faster rendering; Generating whole range of contours by a single module; Generating surfaces (these produced by field_to_mesh do not fit into dimensions, sometime); Optional plane and/or 3D maps; and Second field and colormap input to color surfaces.

field_to_map

Author :	**Andy Bingham** Oxford University, Dept. of Earth Sciences
Number :	1705
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	HP IBM Kubota Sun

The field to map module converts 2D scalar field into a surface in 3D space, represented as a GEOM-format mesh. The geometry is annotated with boundaries, tick marks and a colour scale, for geographic map visualisation. The input data must be a 2D field with a scalar data value at each element. The extents of the field are used to annotate the map.

glue

Author :	**James Pipe** University of Michigan Medical School
Number :	1612
Language :	FORTRAN
Submitted :	04/10/93
Version :	1.000
Last Updated :	04/10/93
Ported To :	DEC IBM Kubota

Continually pastes on incoming data onto the end of a growing output data set - always checking to make sure that the dimension lengths of the input and output data sets are identical EXCEPT for the dimension in which the glue operation is proceeding (also checks vector lengths). The parameter 'total sets' is used ONLY for reporting the total number of data sets glued together in the current output data set - changing its value manually will do nothing.

grid_mesh

Author :	**Michael Rangitsch**
Number :	1124
Language :	C
Submitted :	01/26/92
Version :	1.000
Last Updated :	01/26/92
Ported To :	Kubota HP

This module generates an outline or grid mesh at a constant topological plane. The coordinate information of the input field is used to generate the geometry by connecting adjacent nodal positions within a specified plane.

hog_2D

Author :	**Advanced Visual Systems and Evans and Sutherland**
Number :	1660
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

This module draws vector arrows on the lattice of a 2D field, at the node positions. It draws arrows in the 2D plane, given X-Y vector components. The downsize dial subsamples the mesh, so not all nodes get arrows, for clarity. There is never a Z component to the vectors, except that the arrow heads are 3D pyramids. Even if the input field is 3-vector or more, it only draws arrows assuming 2D vectors in first two vector components. The arrows are constructed from disjoint line segments.

interpolating_sl

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1198
Language :	C
Submitted :	05/07/92
Version :	1.000
Last Updated :	05/07/92
Ported To :	IBM Kubota DEC HP

Interpolating slicer is similar in function and form to “orthogonal slicer“ and ”new ortho slicer“. Orthogonal slices are extracted from a source dataset. Refer to the help for those modules on the base functionality. The difference is that non-integral slicer numbers are permitted. In the case of a non-integral slicer number, the output data is computed to be a linear average of adjacent slices from the input data. For example, if a value of 25.4 is used as the slice number, then the resulting data would be 60% of slice 25 and 40% of slice 26.

irreg2map

Author :	**John Tee**
	AVS Inc. Consultant
Number :	1664
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

IRREG2MAP is intended for applications in which data from scattered sensors is stored in irregular fields, and needs to be displayed, or processed , as uniform data. In particular, a time series of measurements may be stored in a 2D irregular field, with the one compute dimension representing time, and the other dimension the data values recorded at that time. For the data to be displayed in the Image Viewer, or as a texture map in the Geometry Viewer, this module is needed to create the uniform field required. IRREG2MAP produces a uniform byte field in which the pixels are set to zero, except where data from a selected row or column of the input field exists. The resolution of the output field is determined by the X_start, Y_start, X_stop, Y_stop, X_step and Y_step parameters. The module ignores any Z data in a 3-space field. The X and Y limits determine the range of coordinates, present in the input field, that are mapped into the output field. Values which fall outside the range of coordinates selected for the output field are ignored.

labeler

Author :	**Phil McDonald** NOAA/ERL/Forecast Systems Laboratory
Number :	1621
Language :	C
Submitted :	05/01/93
Version :	1.000
Last Updated :	05/01/93
Ported To :	Kubota

The labeler module may be used to create label objects from field data or from arbitrary text. Fields from which data labels are derived may contain data of any vector length and of any dimensionality. Scalar values may be translated into arbitrary character strings by using a value translation file. Title labels may be created from a single string entered through the module control panel or may be created from the text and attributes contained in an ASCII title file. Title labels may be fixed in the display window or attached to points in the object space. Title labels of this latter type are referred to as "object" labels. Object labels may be specified only through a title file. Object labels and data labels may be offset from their reference points. "Pins" may be used to connect these labels to their reference points. Labels may be colored individually, colored uniformly, or uncolored. Each instance of a labeler module will create either data labels or title labels (including object labels), but not both.

line_1D

Author :	**Kathleen Dyer** Lawrence Livermore National Laboratory
Number :	1121
Language :	C
Submitted :	01/22/92
Version :	1.000
Last Updated :	01/22/92
Ported To :	Kubota DEC IBM Sun HP

This module reads in a "field 1D 2-space or 3-space irregular" and produces a "line". The line is a GEOM object composed of disjoint lines, the vertices of which are determined from the grid points of the field. This module was written so that the location of an orthogonal slice of a 2D field could be shown via the "geometry viewer" module.

line_probe

Author :	**Phil McDonald** NOAA/ERL/Forecast Systems Laboratory
Number :	1620
Language :	C
Submitted :	05/01/93
Version :	1.000
Last Updated :	05/01/93
Ported To :	Kubota

The line probe module can be used to sample a 3D field along one or more lines or paths. The location of the probe may be entered manually, indicated by pointing and clicking the mouse, selected from a file of sampling locations, or determined by the sampling locations in a second field. If either of the sampling options is used, one sampling location may be selected, creating one probe, or all sampling locations may be selected, creating multiple probes, one for each sampling location. Except for the sample field mode, the probe will be a vertical line running through the height of the data volume at a constant X and Y position. By using the sample field mode, the paths of the probes may be non-linear, like trajectories. If the line probe module has created a single probe, the values found at its intersections with each of the fields layers will be listed in a text browser and output as a 1D field. If multiple probes are created, the output field is 2D (one "row" for each probe) but the values are not shown in the text browser. In either case, a polyline geometry object is created for the line probe(s) and spheres may be used to show the locations of the nodes (sampling points along the probe). If an optional colormap is provided and the field consists of scalars, the line probe and its nodes will be colored based on how the values along the probe map into this colormap. If a colormap is not provided or the contains vectors, the line probe and its nodes will not be colored.

local_coord

Author :	**John Stephen** Tessella Support Services
Number :	1668
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

This module gives you dial-based control over a probe/cropper, etc. It sets up a coordinate system relative to the transformation matrix of a named object (usually "top"). Dials are used to translate the derived origin from the parental coordinate system and also to rotate its axes. Other dials then allow the object to be translated and rescaled within the new coordinate system.

loop_objects

Author :	**Phil McDonald** NOAA/ERL/Forecast Systems Laboratory
Number :	1278
Language :	C
Submitted :	05/29/92
Version :	1.000
Last Updated :	05/29/92
Ported To :	Kubota

Loop objects is a coroutine module that provides a mild form of animation by providing the capability of looping a series of geometry objects. The result is that all of the controls (object, camera, and light transforms, etc.) that are available through the render geometry module remain available before, during, and after looping. The objects to be looped may be single objects or several objects grouped under a parent object. As each object (or family) is received, loop objects makes a note of it and creates a parent for it named "loop_objx", where x is the index of the object in the series. The entire series is given the parent "loop_objects". Animation is accomplished by altering the visibilities of the objects so that only one is visible at a time.

mesh_axis

Author :	**Tim O. Robinson** UC Berkeley
Number :	1276
Language :	C
Submitted :	05/27/92
Version :	1.000
Last Updated :	05/27/92
Ported To :	DEC IBM Sun Kubota HP

This module generates 3D coordinate axes for a surface mesh. It will also optionally generate tick marks along each axis, numeric labels for the tick marks, and movable mesh planes perpendicular to each of the axes.

mesh_slicer

Author :	**Anonymous** Kubota Pacific Computer Inc
Number :	1156
Language :	C
Submitted :	03/12/92
Version :	1.000
Last Updated :	03/12/92
Ported To :	Kubota DEC IBM Sun HP

Displays a lattice phase of field data.

mesh_to_ucd

Author :	**Ian Curington** Advanced Visual Systems
Number :	1751
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

A mesh_to_ucd makes a 2D UCD surface showing surface height variation from one field. Each cell is drawn as per field to mesh, (linear interpolation), or in block mode with descrete non-interpolated colors. It will optionally leave holes in the UCD output where the data goes out of range for the float dials, allowing mask values to control cell visibility, or region boundaries. It is basically a variation on field_to_mesh, allowing more advanced UCD type visualizations and data processing.

mesh_with_base

Author :	**Phil McDonald** NOAA/ERL/Forecast Systems Laboratory
Number :	1266
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Kubota

The mesh with base module generates a mesh surface from a field, much like field to mesh does. However, an optional base, or “skirt”, can be added to the surface to make it appear as a solid, rather than as a simple wavy papery surface. For 1D fields, the surface degenerates to a polyline and the base degenerates to a “curtain”. If an optional colormap is attached, the surface will be colored according to how the field data values map to the colormap, otherwise, the surface will not be colored. Also, the base color may be selected from the colormap.

new_abuffer

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1113
Language :	CandFORTRAN
Submitted :	12/12/91
Version :	2.000
Last Updated :	05/19/92
Ported To :	IBM Sun Kubota

The new abuffer module is a ray traced volume renderer. Volumes of three dimensional color information are composited on a pixel by pixel basis, resulting in an image of a translucent solid, the signature of volume rendering. The module will compute a set of faces which bound the rectangular shaped volume hull, and perform ray intersections with this volume at each pixel location. For those pixels in which an intersection occurs, the “volume function” is integrated throught the volume, the resulting color information is stored at that pixel location. This module passes the vertices defining the volume hull (boundary) through a viewing transformation, taking the vertices from their native data space into one defined by a canonical view volume. Further, the center of the volume is placed into the center of the resulting image. The user can control the orientation of the volume, as well as specify isometric volume scaling.

new_arbitrar

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1003
Language :	C
Submitted :	10/10/91
Version :	2.000
Last Updated :	02/09/94
Ported To :	Kubota IBM Sun HP

The new arbitrary slicer module extracts a 2d slice from a 3d field. Output consists of a 2d field (as opposed to geometry, as in the “normal” arbitrary slice module) which may be piped to other modules which process 2d fields. The position, size and resolution of the slice plane are under user control via the parameters. For all input fields, the module will determine “sane” parameter settings for extracting initial slice. The latest update provides PRELIMINARY support for irregular fields. This HAS ITS LIMITATIONS - it uses a brute-force algorithm WHICH IS VERY SLOW; as space subdivision on warped grids is still a research topic.

new_bubbleviz

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1318
Language :	C
Submitted :	08/30/92
Version :	2.000
Last Updated :	12/15/93
Ported To :	Sun HP

The bubbleviz module generates spheres of various radii and colors at the element locations of a 1D, 2D or 3D field. This is a "cuberille" style of volume visualization, except that it uses spheres rather than cubes. This module can be used for non-uniform input fields. Version 2 upgrade provided by Martin Serrano (mas@epsilon.com) allows this module to work with 1-coord fields and 2-coord fields.

new_ortho_sl

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1057
Language :	C
Submitted :	10/10/91
Version :	3.000
Last Updated :	05/19/92
Ported To :	DEC IBM Sun Kubota HP

The orthogonal slicer module takes an n-1 dimensional slice from an n-D array. It does so by holding the array index in one dimension constant, and does so by letting the other indices vary. For instances, a data set might include a volume of 5000 points, arranged as follows (using FORTRAN notation) DATA(I,J,K) I=1,10 J=1,20 K=1,25 You can take a 2D "I-slice" from this data set by setting I=4 and letting the otehr indices vary DATA(4,J,K) J=1,20 K=1,25.This module is capable of slicing through 4D fields.

non_ortho

Author :	**Tim O. Robinson**
	UC Berkeley
Number :	1605
Language :	C
Submitted :	04/10/93
Version :	1.000
Last Updated :	04/10/93
Ported To :	DEC IBM Kubota

This module applies a non-orthogonal transform to a uniform input field. The typein_real values determine the axis length and directions. For a typical 3D field, the a parameter corresponds to the x-axis, the b parameter the y-axis and the c parameter the z-axis. The angle alpha is the angle between vectors b and c, the angle beta is the angle between c and a, and the angle gamma is the angle between a and b. The default values correspond to no transformation whatsoever. This module has been tested briefly with 2D fields. The a, b, and gamma parameters are the only parameters that will have an effect on such a field.

ov_slicer

Author :	**Anonymous**
	Stardent Computer
Number :	1059
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

The orthogonal vector slicer module takes a 2D slice from a 3D array of data values and a 3D array of vectors, overlaying the vector slice on the data slice as a hedgehog display. It does so by holding teh array index in one dimension constant, and letting the other indices vary. See new_ortho.

plumb_line

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1677
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

The plumb_line module annotates the top surface of an irregular 3D field, with red cross hairs over each node. The cross hairs are pickable, and drive a line along the remaining axis, through the 3D space. The (I,J) index for the picked point is shown and can be controlled with the X-Y integer dials, and are output on two integer output ports. This can be used to control modules such as "orthogonal slicer" to create a cutting sheet or a graph plot of the selected data section. The picked cross hair changes color to white, and a polyline following the mesh is drawn below.

probe_path

Author :	**Anonymous**
	Stardent Computer
Number :	1071
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

The probe path module displays the numeric data values in a field at a location in space. It works for fields that have been rendered as a n AVS geometry. It works for uniform, rectilinear, and irregular coordinates, and for any data type. It works for both scalar and vector fields. probe_path can accumulate the 3D path where the cursor has been, showing a colored line in 3D space of the probe history. Three other colored lines are shown, where the path is "projected" in 2D onto the axis planes. The sampled data is also stored along the path, and passed out in the 1D field structure. This is typically sent to other filter modules or the graph viewer.

real_math

Author :	**Lars M. Bishop**
	National Center for Comp. Elect., Univ. of Illinois
Number :	1574
Language :	C
Submitted :	02/11/93
Version :	1.000
Last Updated :	02/11/93
Ported To :	DEC IBM

This module works with the same interface as the Field Math module, with two input ports (one required, one optional), and one output. If the operation selected from the control panel is unary, then the required input port is used as the operand. If the operation is binary, then both ports are used if they are both connected. If only the required port is connected, then the required port, and the scalar parameter are used as the two operands. It is the user's responsibility to assure that the values sent to this module are in the domain of the selected function. This module is part of the the Real Math module group, which is a single file, process-sharing set of modules that allow basic operations on and between integer and real AVS data types. They have been most useful in dealing with linkage of several AVS parameters. AVS has no simple way of performing math operations on reals, even though it does allow math operations on fields.

rgbtogray

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1536
Language :	C
Submitted :	10/03/92
Version :	1.000
Last Updated :	10/03/92
Ported To :	DEC HP

The rgbtogray module will receive an AVS image as input, then compute an output field which is a gray scale version of the rgb components of the input image.

rubberbonds

Author :	**Nihar Gokhale**
	Argonne National Laboratories
Number :	1730
Language :	C
Submitted :	12/14/93
Version :	1.000
Last Updated :	12/14/93
Ported To :	Kubota Sun

Rubberbonds forms a stretchable bond between two atoms, given a molecule, and the atoms within the molecule. The cylinders start with a default radius for the equilibrium condition. As the bond gets stronger (shorter distance), the radius is enlarged proportionately. As bonds get weaker (longer distance), the cylinders become long and thin, and, at some point "pop" (i.e. are no longer visible).

scat_2d

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1077
Language :	C
Submitted :	10/10/91
Version :	2.000
Last Updated :	10/10/91
Ported To :	DEC IBM Sun

The scat2d module will resample 1d scatter data in 2 variables. A typical use for this module is to take a field of 2 corrdinate, scalar values, and resample over a "regular" grid.

scat_3d

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1078
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Sun

The sact3d module will resample sacttered data in three variables. A typical use for this module is to take a field of 3-coordinate, scalar values, and resample over a "regular" grid.

scat_base_la

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1754
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

If your vector data is 1D n-vector float irregular, this module will draw a text label just above the first point, showing the data label in the AVSfield header. This was used in one application to show the names of wells, floating above the well heads.

scat_bub

Author :	**Michael Rangitsch**
Number :	1284
Language :	C
Submitted :	06/20/92
Version :	1.000
Last Updated :	06/20/92
Ported To :	DEC IBM Sun Kubota HP

Visualize a scatter data set (field 1D real 3-space) by drawing spheres at the coordinate locations of the field and optionally coloring the spheres based on the field values and optionally dropping vertical lines in each coordinate plane.

scatter_arrows

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1326
Language :	C
Submitted :	08/31/92
Version :	2.000
Last Updated :	02/21/93
Ported To :	Sun HP

This module draws vectors with arrow heads from scatter input showing vector data at the specified positions. The module will draw both 2D and 3D vectors, as required. The Scale dial is simply a length multiplier. Version 2.0 mods submitted by Steve Larkin of the University of Manchester.

scatter_bounds

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1334
Language :	C
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	DEC Sun HP

This module draws a white bounding box from scatter input (1D irregular fields, or random collections of points) showing data min/max positions. It can be used with "scatter dots".

scatter_cubes

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1333
Language :	C
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	DEC Sun HP

This module draws variable size coloured cubes from scatter input showing scalar data at the specified positions. It can be used in place of "scatter dots". The Scale dial is simply a size multiplier. The size of each cube is taken by the first vector if "Variable", otherwise it is set to "Scale" size in "Uniform" mode. The colour of each cube is independently controled from the remaining three vector quantities. If the input vector length is 1 (scalar), the colour will default to white.

scatter_rib_2

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1756
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

If your vector data is 1D n-vector float irregular, this module will draw a ribbon going along the side of the polyline path, from joining up the specified positions in the scatter field coord array. This module is used for Well Log data display, showing various properties down a deviated well shaft. The width, and color of the ribbon can be controlled to create a ribbon data representation. Tick marks and labels will annotate the graph. Interpolation can be turned on and off.

scatter_ribbon

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1682
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

This module may be used for 3D Well Log data display, showing various properties down a deviated well shaft. If your vector data is 1D n-vector float irregular, this module will draw a ribbon going along the side of the polyline path, from joining up the specified positions in the scatter field coord array. The width, and color of the ribbon can be controlled to create a ribbon data representation. Tick marks and labels will annotate the graph. Interpolation can be turned on and off.

show_flow

Author :	**Phil McDonald** NOAA/ERL/Forecast Systems Laboratory
Number :	1195
Language :	C
Submitted :	05/06/92
Version :	1.100
Last Updated :	04/05/93
Ported To :	IBM Kubota Sun

The show flow module generates a geometry object named "flow symbols" that depicts the flow in a field of 1-, 2-, or 3-vectors of any data type and in any grid configuration. Show flow provides the same basic function as hedgehog but it has been enhanced to serve the meteorolog- ical community where wind barbs, rather than arrows, may be the desired flow symbol. Flow speed values are the resultant values of the field n-vector nalues. The symbols may be rotated about their reference points and their aspect with respect to the viewer can be frozen. While symbols of either type can be arbitrarily sized, arrows can be further scaled by a divisor which is the maximum of the field, the average of the field, or entered manually. The symbols will be colored by their speeds or the values in a second, optional field if an optional colormap is supplied.

str_to_geom

Author :	**Urs Meyer** Swiss Scientific Computing Center
Number :	1625
Language :	C
Submitted :	07/02/93
Version :	1.000
Last Updated :	07/02/93
Ported To :	HP IBM Kubota DEC Sun

This module creates a "label" style text string in the GEOM format which can be used as a title. It takes a string value as input. It can be used to represent some important variable such as time, animation step, some parameter, etc. It has the advantage over the labeling facilities in the Geom Viewer in that the information will get saved with a network. Other things you can control are the font number, drop shadows, justification, position, height, and color of the text string.

strip_charts

Author :	**Alan Barnum-Scrivener** Advanced Visual Systems Inc.
Number :	1847
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP Kubota Sun

I got sick of people saying that AVS won't do mutiple strip charts, even though we all know "it can't not do it," so I whipped up this demo. (It only works with float data, and assumes a data range of 0.0 to 256.0, but it does make a convincing demo.) The module is a subroutine module that takes multiple input signal lines from a float 2D field, and creates a line map showing the progression of the signal amplitude. The representation is created using AVS geometry, and may be sent to the geometry viewer, "tube" module, etc. An extent box is drawn around the border of the plot area. By using an animated integer controlling a moving crop region, the signal can be animated to simulate a traditional chart recorder.

surface_mesh

Author :	**Phil McDonald** NOAA/ERL/Forecast Systems Laboratory
Number :	1270
Language :	C
Submitted :	05/27/92
Version :	1.000
Last Updated :	05/27/92
Ported To :	Kubota

The surface mesh module creates a 3D wire mesh from a polytriangle object of a 3D field. Typically, polytriangle objects are used to represent surfaces such as those created by the isosurface and field to mesh modules. The surface mesh module creates a wire mesh of such a surface by passing planes that are perpendicular to the coordinate axes and finding the line segments that are the intersections between the surface and these planes. The mesh resolution (or distance between the planes) along each axis is independent from that along the other axes. The edges of the polygons defined by the polytriangles may also be shown.

surface_web

Author :	**Phil McDonald** NOAA/ERL/Forecast Systems Laboratory
Number :	1271
Language :	C
Submitted :	05/27/92
Version :	1.000
Last Updated :	05/27/92
Ported To :	Kubota

The surface web module creates a 3D wire web from a surface object created from a 3D field. Modules that create surfaces, like the isosurface and field to mesh modules, represent the surfaces with polytriangle strips. The web is created by running "strings", or lines, from each vertex of each polygon defined by these strips to the line segment opposite the vertex. The web density is the number of strings run from each vertex. The edges of the polygons are always shown, even if the web density is zero.

symmetrize

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1137
Language :	C
Submitted :	02/24/92
Version :	1.000
Last Updated :	02/24/92
Ported To :	Kubota DEC IBM Sun HP

The symmetrize module inputs a list of r and z values (cylindrical coordinates) which define a axisymmetric object, and then constructs a mesh object around the Z axis using this list of coordinates to defines the vertices. The number of divisions cylindrically around the Z-axis, in the theta direction, is determined by the divisions parameter.

texture_mesh

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1295
Language :	C
Submitted :	08/04/92
Version :	2.000
Last Updated :	12/14/93
Ported To :	IBM Sun HP

The texture mesh module serves two needs. First, the need for a robust module for mapping 2D data to quadrilateral meshes is served. The input 2D data may be a 2D uniform field, in which case the x,y coordinates for the output mesh is derived from the extents information (if present) or using a mapping from indeces to space if no extents information is present. The z values are obtained from the data in the field.

tracer_bounds

Author :	**Lars M. Bishop** National Center for Comp. Elect., Univ. of Illinois
Number :	1570
Language :	C
Submitted :	01/28/93
Version :	1.000
Last Updated :	01/28/93
Ported To :	IBM

Tracer Volume Bounds - This module takes in a 3D Uniform 4 vector Byte field, and adds a white frame to the edges. It is designed to be applied to a volume of data right before it is sent to the tracer module, and allows the addition of lines that look like the geometry viewer's volume bounds, which are normally inaccessible to volume-traced data. It also includes the ability to fill in the min X,Y, or Z walls, as the original volume bounds module does.

transform_tool

Author :	**Tim O. Robinson** UC Berkeley
Number :	1606
Language :	C
Submitted :	04/10/93
Version :	1.000
Last Updated :	04/10/93
Ported To :	DEC IBM Kubota

The transform tool module is used to obtain a copy of the current transformation matrix from the geometry viewer and apply it to an oject in another geometry viewer window. For example, to capture the transformation for the top object in one geometry viewer window and apply it to the top object in another geometry viewer window, set both typein values to %top. Currently, the transformation is only used after motion of the control geometry stops. Experienced users may experiment with the flags setting in the GEOMedit_transform_mode call in the module code to alter this behavior.

trivar

Author : **Wes Bethel**
Lawrence Berkeley Laboratory
Number : 1088
Language : C
Submitted : 10/10/91
Version : 2.000
Last Updated : 05/19/92
Ported To : DEC IBM Sun Kubota

The bivar module will resample 1d scatter data in 3 variables. A typical use for this module is to take a field of 3-coordinate, scalar values, and resample over a "regular" grid. The algorithm used is one in which, for each output grid location, all input points are scanned to determine if they lie within the specified search radius. If so, that input point's contribution is weighted as a function of distance from the output grid point and scaled by 1./distance**weightfunction. As such, this module is best suited for use on input data which doesn't necessarily fit a diffentiable function. For such data, refer to the scat3d module.

ucd_cell_arrows

Author : **Ian Curington**
Advanced Visual Systems
Number : 1186
Language : C
Submitted : 05/04/92
Version : 1.000
Last Updated : 05/04/92
Ported To : DEC IBM Sun HP

This module draws vectors with arrow heads from UCD input showing CELL vector data. This is a special ucd mapper module designed to make representations based on CELL based values rather than the more typical NODE based values. If you have CELL based values, you should also consider the ucd_cell_to_node module. The module will draw both 2D and 3D vectors, as required. The Scale dial is simply a length multiplier. This module is intended as a source example for other UCD module developers.

ucd_cell_cones

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1187
Language :	C
Submitted :	05/04/92
Version :	1.000
Last Updated :	05/04/92
Ported To :	DEC IBM Sun HP

This module draws cones from UCD input showing CELL vector data. The length and direction of the cones are representative of the vector quantity passed in through the cell data ucd structure. Scale and width dials control magnification factors on the raw data, to get the cones sized for visibility. The subdivision control defines the course to fine tessalation. This is a special ucd mapper module designed to make representations based on CELL based values rather than the more typical NODE based values. If you have CELL based values, you should also consider the ucd_cell_to_node module. The module will draw both 2D and 3D vectors, as required. The Scale dial is simply a length multiplier. This module is intended as a source example for other UCD module developers.

ucd_cell_geometr

Author :	**Ian Curington** Advanced Visual Systems, Inc.
Number :	1188
Language :	C
Submitted :	05/04/92
Version :	2.000
Last Updated :	09/30/93
Ported To :	HP IBM Sun

This module draws geometric representation of UCD structures showing CELL data, CELL Material ID, or Cell Number Sequence, Node Number Sequence, scalar Node data, Cell topology type, Face order, node positions, or face dip, by color. It shows all cells in either solid or wire frame, unlike ucd_to_geom and colors the model directly, without using and external module such as ucd_contour or ucd_cell_colors. It allows picking and moving nodes within the model, and has transformation icons, drag lines, and position markers to aid position editing. The edit information is available in a small field, containing new x-y-z node position, node number, and cell number. Exterior faces only may be displayed with the remove interior option. Downsize will subsample the structure and only display a fraction of cells, for faster interaction. The color range is defined by the colormap hi-low, and DOES NOT autonormalize to the input UCD range, so allows user refinement of the colormap. The "Dip" option allows pseudo-color of surface angle or curvature away from each axis. This is a major update from version 1.0. This module colours each cell ONE color, without color interpolation, showing a single scalar quantity. If your cell data is n-vector, this module will only color the first one, so use ucd_extract to get the one you want.

ucd_line

Author :	**Jonathan Cox** Department of Computer Science, University of Manchester
Number :	1535
Language :	C
Submitted :	10/01/92
Version :	3.000
Last Updated :	09/30/93
Ported To :	Sun DEC HP IBM

This module produces an arbitrary line of sampled node data from a 3D UCD mesh and outputs it as a field. This can be a conventional AVS 1D field with irregular 1D coordinates or a 2D field suitable for connection to the graph viewer module. It is also possible to output a uniformly spaced 1D field. Version 2 fixes bug concerning disappearing input ports. Ported to IBM 12/18/92 SRT. Version 3 changes - Code is ANSI C, therefore it requires AVS 5. Modules have been restructured to make them cooperative and reentrant. They are now designed to be compiled in a single executable. If you want to split them up on the FTP site refer to the README file to find which files are required for each module.

ucd_line_2

Author :	**Jonathan Cox** Department of Computer Science, University of Manchester
Number :	1767
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Sun

This module produces an arbitrary line of sampled node data from a 3D UCD mesh and outputs it as a field. This can be a conventional AVS 1D field with irregular 1D coordinates or a 2D field suitable for connection to the graph viewer module. It is also possible to output a uniformly spaced 1D field. The position and orientation of the line is arbitrary and is controllable either by the direct manipulation of a probe object or by manipulating a collection of dial widgets. The original version of this module (also on the IAC's anonymous ftp site) only works with AVS 4.0, for this enhanced version AVS 5.0 is required.

ucd_planes

Author :	**Jonathan Cox** Department of Computer Science, University of Manchester
Number :	1534
Language :	C
Submitted :	10/01/92
Version :	3.000
Last Updated :	09/29/93
Ported To :	Sun DEC HP IBM

The UCD PLANES module supports the interactive sub-sampling of node data held within a 3D UCD structure. The region sampled is selected interactively in the region of a transparent 'probe' object. Several different types of probes are supported, all based upon a collection of planes. Version 2 fixes bug concerning disappearing input ports. Ported to IBM 12/18/92 SRT. Version 3 - Support for the generation of 2D contours has been added. Contour lines can be superimposed on the output of any of the probes inc. the 3D graph. The range of values used can be set using ucd legend in a similar manner to ucd isolines. Also, Sampled region can be output as a UCD structure consisting of triangular elements. A default layout is included in the description function. There are now too many widgets to fit comfortably onto a single control panel. The default layout splits the widgets into three subpanels. Parameter modifications are interpreted more intelligently to avoid needlessly regenerating the sample when it is not needed. Code is ANSI C, therefore it requires AVS 5. Modules have been restructured to make them cooperative and reentrant. They are now designed to be compiled in a single executable. If you want to split them up on the FTP site refer to the README file to find which files are required for each module.

ucd_planes_2

Author :	**Jonathan Cox** Department of Computer Science, University of Manchester
Number :	1768
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Sun

The UCD PLANES module supports the interactive sub-sampling of node data held within a 3D UCD structure. The region sampled is selected interactively in the region of a transparent 'probe' object. Several different types of probes are supported, all based upon a collection of planes. For each probe type, the sampled region can be output either as various forms of geometry, as a UCD structure or as both. When a geometry is output, 2D contours of a scalar data component can be superimposed on the sample object. The selection of probe type is made from the 'Probe Type' choice widget on the module's control panel. Possible probes include a single plane (SIMPLE), two and three crossed orthogonal planes (SHEAF and CROSSED) and multiple parallel planes (MULTIPLE). These probe types produce a 'sample' that is the same shape as the probe object but is coloured according to a scalar component of the structure's nodal data. The original version of this module (also on the IAC's anonymous ftp site) only works with AVS 4.0, for this enhanced version AVS 5.0 is required.

ucd_plumbline

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1690
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

The plumb_line module annotates the top surface of a UCD structure with red cross hairs over each node. The cross hairs are pickable, and change colour. The node number and x-y coordinate value of the picked node is shown and can be controlled with the an integer typein, and are output on one integer and two float output ports. This can be used to control modules that use position dials, perhaps to create a cutting sheet or a graph plot of the selected data section. Downsize control allows a subset of node positions to be visible, while size and offset controls specify the marker shape and position.

ucd_profile

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1761
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This module draws Vector Arrows from UCD Cell Surface Boundary data as intersected by a surface cross-section profile. Airflow Lift-Drag analysis plots from surface pressure can be manipulated or animated dynamically, by dragging the profile plane through a large model. The profile sampled data can be sent to the graph viewer, or other field modules. This mapper module computes the exterior boundary surface of the UCD from node connectivity and material regions, then cuts these faces with an interactive slicing profile plane. It works with 2D shell elements or 3D elements.

ucd_streak

Author :	**Ian Curington** Advanced Visual Systems
Number :	1762
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Sun

The ucd streak module generates surface constrained stream lines. It is an advanced variation of the ucd_streamlines module in AVS5. It contains an optional mode where the streamlines follow the boundary of the flow domain, tracking along the surface, as long at the velocity field is into the boundary surface, and are not displayed when they leave the surface. This simulates oil drop streak lines from exterior flow wind tunnel tests. The flow is integrated through the 3D boundary elements, and optional integration step markers can be shown. Although the streamlines are only displayed at the surface, the whole interior 3D streamline calculation is taking place, so this method is quite different from a 2D surface only particle track algorithm.

ucd_to_prim

Author :	**Penny Rheingans** US EPA Scientific Visualization Center
Number :	1312
Language :	C
Submitted :	08/30/92
Version :	1.000
Last Updated :	08/30/92
Ported To :	DEC Sun HP

This module takes an input ucd structure and generates a geometric primitive (sphere or prism) for each node of the structure. Primitives are colored according to a selected value at the node and an optional input colormap. If no colormap is provided, primitives will be mapped with a grey scale.

volume_mesh

Author :	**Ed Bender** Convex Computer Corporation
Number :	1593
Language :	C
Submitted :	03/04/93
Version :	2.000
Last Updated :	03/23/93
Ported To :	IBM Kubota HP

The volume mesh module is similar to the volume bounds module. The difference is that it generates a mesh using a colormap for the desired faces of the irregular field instead of a wireframe. This replaces up to six ortho_slice/field_to_mesh combinations. By placing the crop module upstream, it is easy to control the location of the slice planes. Version 2.000 update generates normals.

waffle

Author :	**Ian Curington**
Number :	1286
Language :	C
Submitted :	06/20/92
Version :	1.000
Last Updated :	06/20/92
Ported To :	IBM Sun Kubota HP

The waffle module converts a two-dimensional floating point field into group of 3D cross section planes, represented as a GEOM-format mesh, with the base at zero. Each X or Y line of the field is mapped to a "fence". The height of each fence post above each point is proportional to the scalar value of the field. This is my attempt at "Business Graphics", for a multi-dimensional telecommunications statistics data application, while allowing 3D interaction.

wire_bender

Author :	**Phil McDonald** NOAA/ERL/Forecast Systems Laboratory
Number :	1194
Language :	C
Submitted :	05/06/92
Version :	1.000
Last Updated :	05/06/92
Ported To :	Kubota

The wire bender module bends line vectors to conform with a surface defined by a 2D field. The source of the input vectors may be either a geometry object or an ASCII vector file. If no field is specified, the input vectors are passed through unmodified, retaining their original 2D or 3D shapes.

zone_fence

Author :	**Ian Curington** Advanced Visual Systems
Number :	1763
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Sun

This module draws a fence panel connecting two wells, joining up common zones, and texture map image data onto the fence panels. This routine takes three geometric field inputs, one the zonation table, containing a 2D array of wells and zones, and the deviation scatter path data for two wells. It creates a geometric representation of the zonation region between the wells. The depths are resampled using linear interpolation from the deviation data to create a 3D sheet. Two optional inputs, texture image A and B, if present will texture map onto alternate panels of the fence.

zone_picker

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1764
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This routine takes two 1D inputs, one the zonation table, and second, the deviation file for one 3D well track. It creates a geometric representation of the zonation picks along the well bore, as golden rings, and allows picking and dragging of the positions for editing. It looks at the labels on the well dev field to figure out which well it is, then indexes the zonations by this, so there is no menu selection needed here. The zone icons can be picked and dragged along the well, the new positions recorded in the memory file structure.

secno = ount0

DATA OUTPUT MODULES

International AVS Center

Bit_Movie

Author :	**Ian Curington** Advanced Visual Systems
Number :	1832
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP IBM Sun

The bit_movie module is used to create and playback single bit black and white movie loops. It takes source images from any image source in AVS. The bit_movie module saves up movie loop sequences in "The X server" memory, and then allows for playback after storage. The motivation is to create the longest animation sequences possible, at the highest possible display rate, without going to disk.

CREATE_MINI_DATA

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1794
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Kubota Sun

The CREATE MINI DATA module is used in conjunction with the AVS module "write field", and other specially designed modules include "FIND MAP", "MULTI_extract", and "WRITE TIMESTAMP". The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

CREATE_USER_MINI

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1795
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Sun

CREATE USER MINI DATA – creates a local data set from a remote data file and place it in the users own directory structure. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

Create_MPEG

Author :	**Alex Knowles** The University of Manchester
Number :	1789
Language :	C
Submitted :	04/21/94
Version :	2.000
Last Updated :	07/15/94
Ported To :	DEC HP IBM Sun

Create MPEG is a subroutine module which will create MPEG movies from a series of AVS images. It uses Andy Hung's 'MPEG' which is available by anonymous ftp from Stanford. Because of the way MPEG movies are encoded by 'MPEG', many temporary files need to be stored, bear this in mind when choosing which directory to make your movie in. Version 2 fixes bugs.

Draw_Map

Author :	**Jeff Wang** MCNC, Environmetal Programs Group
Number :	1741
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Sun

This module allows users to draw the United States map with state boundary and county boundary. With the distribution of the module, two map data files are also included. "allcounty.txt" is ASCII data for boundaries of ALL counties in the United States, including Alaska, Hawaii and Perto Rico. "countryd.txt" is ASCII data for boundaries of the states. Users may construct their own boundary map according to the format. This is one of a suite of modules we used to plot the emission data for the state of North Carolina. This group of modules allows displaying points, lines and polygons over a map of the United States. The module suite is stored as a group in a compressed tar file at the International AVS Center, with some sample networks and datasets included. You may find it easier to access the modules this way. The compressed tar file is approximately 1.2MB in size, and can be found via anonymous ftp to avs.ncsc.org, in /sample_data/avs_data/Wang_GIS_mods s.tar.Z.

GEOM2IV

Author :	**YOSHIKAWA, Hiroyuki** FUJITSU NAGANO SYSTEMS ENGINEERING LTD
Number :	1898
Language :	C
Submitted :	09/11/95
Version :	1.000
Last Updated :	09/11/95
Ported To :	DEC_ultrix DEC_alpha

This module interprets AVS geometry data into Inventor format(V2.0 ascii) and writes it out to a new file.

GenMask

Author :	**Stefano Brozzi, Andrea Menozzi, Giulio E. Destri** Universita' degli Studi di Parma, 43100 Parma, Italy
Number :	1874
Language :	C
Submitted :	09/06/95
Version :	1.000
Last Updated :	09/06/95
Ported To :	DEC_ultrix DEC_alpha

The module GenMask is an input module created to generate the templates, both the control and the feedback template, to apply to the image when the image proc essing is executing. As a input module GenMask does not have any input and it gives as output a field 2-D scalar float that is a 2-dimensional array of float. Since the module aCNN works with templates of dimension 3x3 even the output array of GenMask will be 3x3.

Geom_to_Wavefron

Author :	**Steve Chall** MCNC Visualization Group
Number :	1790
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Sun

"Geom to Wavefront" accepts as input an AVS GEOMedit_list struct which contains a pointer to a linked list of GEOMedit structs. The list is traversed and each element which is of type GEOM_EDIT_GEOMETRY is examined to determine its GEOMobj type–as opposed to GEOMedit type–where the possible types are mesh, polyhedron, polytriangle strip array, sphere, or label. An equivalent Wavefront .obj file is written to disk for each polyhedron and polytri in the edit list. Note that a single AVS geometry may map to multiple .obj files.

HdB1

Author :	**Han de Bruijn** TUD Computing Centre,The Netherlands
Number :	1297
Language :	CandFORTRAN
Submitted :	08/04/92
Version :	1.000
Last Updated :	08/04/92

This product was developed for C3820, ConvexOS/10, Fortran Compiler 6.0. A so-called GEOM driver for the Application Visualization System (AVS) has been built into the classical RC-plotroutines (Calcomp), as an AVS course material spin-off. This means that AVS *.geom files can be produced by the plotroutines, which can be read in and visualized by the Geometry Viewer.

Image_to_Postscr

Author :	**Lars M. Bishop** National Center for Comp. Elect., Univ. of Illinois
Number :	1578
Language :	C
Submitted :	02/11/93
Version :	1.000
Last Updated :	02/11/93
Ported To :	DEC IBM Kubota

The Image to Postrscript module is a module that prints an Adobe Postscript file containing the image that is connected to the input port. It is designed to give the user a wide variety of image sizes, orientations, positions, and qualities. Note that the Dump to File button must be pressed to dump the image to the file, as the dump may take several seconds.

Record_Anima

Author :	**Brian Kaplan** Center for Innovative Computer Applications
Number :	1096
Language :	C
Submitted :	10/16/91
Version :	2.300
Last Updated :	10/16/91
Ported To :	Kubota

Used to record AVS animations to videotape or laserdisc, one frame at a time. It works by talking with either a Lyon-Lamb Minivas videotape controller, or a Sony Laser Videodisc recorder/player. Record Animation sends the appropriate commands to initialize the recording device, to tell the recording device when to record a frame, and to wait for the recording device to complete the frame record before continuing with the next frame. Usually, the Record Animation module is used in conjunction with an animation module of some sort which produces geometry for each frame of the animation. This geometry is then fed into the “render geom” module to be rendered as a pixmap output. The pixmap would then be connected to “display pixmap“ and also to the pixmap input of the Record Animation module. The Record Animation module uses the pixmap to tell when the current scene has changed so that it will know to send the appropriate record command to the recording device.

Spline_Animator

Author : **Mitchell Roth**
University of Alaska
Number : 1860
Language : C
Submitted : 06/01/95
Version : 2.000
Last Updated : 06/01/95

Version 2.0 is basically just the text version (i.e. 1.0) with a Motif interface dropped on top. The goal is to provide easier script creation and adding, as well as a few new features. Interested users should consult the Instructions.ps file for a general guide on creating scripts with v2.0. Not covered in the instructions is the Convert script which is intended to allow v1.0 users to port their scripts to v2.0 format. The v2.0 format will be used in the final AVS/Motif version. The Convert script is simple to use - Convert source_file dest_file

Trace_Render

Author : **Ian Curington, Advanced Visual Systems, Inc.**
Danny Harvey, University of Colorado
Number : 1687
Language : C
Submitted : 09/29/93
Version : 1.000
Last Updated : 09/29/93
Ported To : HP Sun

This is a display module, it takes a 2D floating point field, treats each row as a 1D signal, and draws the signal traces next to each other. The offset, amplitude, signal spacing, etc. can be controlled for any range of data. The signals can be plotted vertically, horizontally, or in 180 degree polar configurations. Additional controls allow for Seismic moveout interpretation, given an Time Offset (T0) and a Velocity, linear or hyperbolic moveout functions may be overlayed and optionally applied to the trace data. The display is like wiggle trace, but the areas under the curve are not filled in. The module catches mouse events in the window, but these are currently commented out. One possibility is to provide dynamic pan and zoom under mouse control, or allow signal picking. If the input is 1D only, it will draw one signal line, and may be used like a real-time oscilliscope display.

WRITE_ANY_IMAGE

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1212
Language :	C
Submitted :	05/21/92
Version :	2.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM HP

This module writes an image from an AVS Network in a variety of formats. All formats which the San Diego Supercomputing Center's image tools support, this module supports. With this module you can write out an AVS image in MacPaint format for inclusion in a to-be published document .. Any of the following image file formats can be read in by this module eps Encapsulated PostScript file, gif Compuserve Graphics image file, hdf Hierarchical Data File, icon Sun Icon and Cursor file, iff Sun TAAC Image File Format, mpnt Apple Macintosh MacPaint file, pbm Portable Bit Map file, pcx ZSoft IBM PC Paintbrush file, pgm Portable Gray Map file, pic PIXAR picture file, pict Apple Macintosh QuickDraw/PICT file, pix Alias image file, pnm Portable aNy Map file, ppm Portable Pixel Map file, ps PostScript file, ras Sun Rasterfile, rgb SGI RGB image file, rla Wavefront raster image file, rle Utah Run length encoded image file, rpbm Raw Portable Bit Map file, rpgm Raw Portable Gray Map file, rpnm Raw Portable aNy Map file, rppm Raw Portable Pixel Map file, synu Synu image file, tiff Tagged image file, x Stardent AVS X image file, xbm X11 bitmap file, xwd X Window System window dump image file. Version 2.0 frees allocated memory after writing each image.

WRITE_TIMESTAMP

Author :	**Kathy Pearson** MCNC Environmental Programs
Number :	1821
Language :	C
Submitted :	04/21/94
Version :	1.000
Last Updated :	04/21/94
Ported To :	DEC HP IBM Sun

WRITE TIMESTAMP – write into the specified file name the input timestamp string. The WRITE TIMESTAMP module creates an ASCII file and writes into it the set of timestamp labels input as one string. This file is needed to indicate the date and time represented by AVS data fields that have been created by extracting pieces from original model files of type ROM, RADM, UAM and ALPHA. The MCNC Environmental Programs Group has created this module as part of a suite of AVS demo modules, networks, and sample data sets to demonstrate the functionality of its visualization prototype for an Environmental Decision Support System (EDSS). You can download these modules individually from the IAC using anonymous ftp to the appropriate directories /avs_modules at avs.ncsc.org. If you would like to download the entire suite of modules, networks, and sample datasets at the same time, the IAC recommends that you use anonymous ftp to grab the following file from avs.ncsc.org - /sample_data/avs_data/MCNC_Env_Pgms s/MCNC_Env_Pgms.tar.Z This contains all source code to all local and remote modules, as well as all of the sample datasets, networks, documentation, installation, and usage notes.

WrUcdAnim

Author :	**D. LE CORRE** CONVEX S.A
Number :	1317
Language :	C
Submitted :	08/30/92
Version :	1.200
Last Updated :	12/30/92
Ported To :	Sun HP

This DATA output module generates a sequence of UCD data time steps onto a (set of) file(s), which may then be read by the twin module, RdUcdAnim. For each time step, it can store node positions, node data, cell data, model data or a subset of these. It is fully compatible with the standard read ucd format, so the first time step may be read by read ucd.

avs2vrml

Author :	**John Evans**
	U.S. Geological Survey
Number :	1899
Language :	C
Submitted :	09/21/95
Version :	1.000
Last Updated :	09/21/95
Ported To :	DEC_alpha

The avs2vrml module saves an AVS geometry data structure as an equivalent VRML Version 1.0 compliant *.wrl file. Meshes, polyhedrons, spheres, and polytriangles are fully-supported. Labels don't quite work yet.

avs2vrml

Author :	**John Evans**
	U.S. Geological Survey
Number :	1900
Language :	C
Submitted :	09/21/95
Version :	1.000
Last Updated :	09/21/95
Ported To :	DEC_ultrix DEC_alpha

The avs2vrml module saves an AVS geometry data structure as an equivalent VRML Version 1.0 compliant *.wrl file. Meshes, polyhedrons, spheres, and polytriangles are fully-supported. Labels don't quite work yet.

comput_vol

Author :	**Andy R. Haas**
	National Center for Toxicological Research
Number :	1868
Language :	C
Submitted :	06/27/95
Version :	1.000
Last Updated :	06/27/95
Ported To :	DEC_ultrix DEC_alpha

This module accepts a geometry object (mesh, polyhedron, polytriangle, or sphere type) and outputs the object's volume and surface area. The name of the object and its and area values are also printed to the terminal window in which AVS was started.

compute_volume

Author :	**Andy R. Haas**
Number :	1897
Language :	C
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_ultrix DEC_alpha

This module accepts a geometry object (mesh, polyhedron, polytriangle, or sphere type) and outputs the object's volume and surface area. The name of the object and its volume and area values are also printed to the terminal window in which AVS was started.

continuous_s

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1136
Language :	C
Submitted :	02/24/92
Version :	1.000
Last Updated :	02/24/92
Ported To :	Kubota DEC IBM Sun HP

Keeps a running tally of the biggest and smallest numbers to come through its input ports. The text string parameters will update each time a smaller of larger value comes through one of its ports. If the reset button is turned on, then the running tally is reset to have to current minimum or maximum. For example, if on one execution the module received 0.020 and -0.020, and at the next execution the module receives 0.039 and -0.05, then the string parameters will display 0.039 and -0.020. If you have a time series of data and you want to find out the absolute minimum and maximum of the data values over time, this module allows that capability.

create_unix_pipe

Author :	**John Stephen** Tessella Support Services
Number :	1645
Language :	C
Submitted :	09/24/93
Version :	1.000
Last Updated :	09/24/93
Ported To :	HP IBM Sun

This module creates and sets up a unix pipe file, for communication between an AVS module and an external application. It makes unix system calls to create the pipe file, based on a typein name. It helps manage the use of the pipe, with options to flush, etc. It outputs the full pathname of the pipe, once established, for downstream modules to use for the communication.

deleteGeom

Author :	**John Stephen** Tessella Support Services
Number :	1647
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

This module deletes objects from the geometry viewer on command, based on the name (alias) of the module that created the geometry.

delete_object

Author :	**John Stephen** Tessella Support Services
Number :	1648
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

Deletes a geometry object having a user supplied name.

dira2avs

Author :	**Massimiliano Bavelloni** Radioastronomy department C.N.R. Bologna,Italy
Number :	1875
Language :	C
Submitted :	09/06/95
Version :	1.000
Last Updated :	09/06/95
Ported To :	DEC_ultrix DEC_alpha

Read data from ascii file, data may be integer or floating point numbers separated by spaces.

display_ARGB_vol

Author :	**Howard Stroyan** Hewlett-Packard Advanced Systems Division
Number :	1823
Language :	C
Submitted :	07/15/94
Version :	1.000
Last Updated :	07/15/94
Ported To :	HP

This AVS module is designed to provide access to the HP specific back-to-front Volume Rendering graphics functionality of the Starbase graphics API. This module can only be successfully built and executed on an HP700 workstation. There are actually two modules here, which have identical functionality.
They both takes in a word/voxel dataset (with packed ARGB) and a transform matrix and renders a volume in a window. The only difference is that the first module uses the PRE_SCALED_ALPHA data form, while the second does not. For an explanation of this CUBIC_POLYPOINT option see the man page for the module.

display_lit_volu

Author :	**Howard Stroyan** Hewlett-Packard Advanced Systems Division
Number :	1822
Language :	C
Submitted :	07/15/94
Version :	1.000
Last Updated :	07/15/94
Ported To :	HP

This AVS module is designed to provide access to the HP specific back-to-front Volume Rendering graphics functionality of the Starbase graphics API. This module can only be successfully built and executed on an HP700 workstation. This module takes in a byte/voxel dataset, a colormap, and a transform matrix and renders a volume in a window. The difference with this module is that it implements a form of gradient shading. For coding convenience and speed the light source direction has been limited to one of the 26 vectors which pass directly through an adjacent voxel. The gradient magnitude is approximated as the magnituded difference of voxel under computation and the adjacent voxel along the light source vector. This magnitude is store in a second byte (per voxel) along with the original voxel scalar magnitude byte. This 2 byte per voxel field is fed to the CUBIC_POLYPOINT gescape along with a table which contains a lighting model applied to the original colormap supplied to the model. This lighting model is driven by ambient, diffuse, and specular weighting and theta/phi light source angles supplied as input parameters.

display_tracker

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1327
Language :	C
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	

Track 2D cursor motion over displayed image. This module takes an input image and displays it in an X-window. Inputs an image and allows the user create mouse clicks and moves in the image window, with event information passed to other AVS modules using the "upstream data" concept. This module acts similarly to "display image" or "display tracker" in that it accepts an input image and displays this in an X-window. It also then accepts asynchronous mouse events in the window, and passes these mouse events (in an upstream transform structure) to an output port. This allows applications to be built using this module to do all the X dependent code, leaving the actual operation (such as pan, draw, etc.) to an upstream module, with no concern of X events.

display_volume

Author :	**Howard Stroyan**
	Hewlett-Packard Advanced Systems Division
Number :	1824
Language :	C
Submitted :	07/15/94
Version :	1.000
Last Updated :	07/15/94
Ported To :	HP

This AVS module is designed to provide access to the HP specific back-to-front Volume Rendering graphics functionality of the Starbase graphics API. This module can only be successfully built and executed on an HP700 workstation. This module takes in a byte/voxel dataset, a colormap, and a transform matrix and renders a volume in a window.

dump_icc

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1029
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Sun HP

This module converts an input pixmap to the so-called ICC format. This image format is used by local device drivers which communicate with the Kodak XL7700 printer. The single parameter is a file browser that allows you to specify the name of the ICC file to be created. After the file is written, the filename is reset to null. This prevents subsequent changes upstream in the network from automatically triggering the rewriting of the file. A new file is written only after the user has entered a filename.

dump_ps

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1030
Language :	C
Submitted :	03/20/92
Version :	2.000
Last Updated :	10/10/91
Ported To :	Sun Kubota HP

This module converts a pixmap to the PostScript page description language and stores it in a file. After the file is written, the filename is reset to NULL. This prevents subsequent changes upstream in the network from automatically triggering the rewriting of the file. A new file is written only after the user has entered a filename. Two types of PostScript output are supported, PostScript-compatible monochrome laser printer and PostScript-compatible color laser printer.

dump_sunras

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1015
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Sun HP

This module produces a sun rasterfile from an input AVS format pixmap. This module is familiar with 8,24 and 32-bit pixmaps. It will write an 8-bit rasterfile when presented an 8-bit pixmap as input, otherwise a 24-bit BGR ("RT_STANDARD") rasterfile is produced.

field2_to_Math

Author :	**Tom Wickham-Jones**
	Wolfram Research
Number :	1539
Language :	C
Submitted :	10/14/92
Version :	1.000
Last Updated :	10/14/92
Ported To :	Sun DEC Kubota

The "field2 to Math" module allows a two-dimensional scalar field to be imported into Mathematica from AVS. When the "field2 to Math" module is invoked an xterm is started running Mathematica with the input and output of the xterm directed to and from the Mathematica session main loop. Entering Mathematica commands into the window will result in the appropriate Mathematica results. The session is initialised with a package of Mathematica commands which have started a MathLink communication channel to AVS. Other initialisations are made so that the Mathematica command AVSReadField[ ] will read a two-dimensional scalar field from AVS. Please note that at the present time the IAC does not have Mathematica installed on a , Sun, or Kubota platforms. As a result this module was placed on the ftp site using the Makefiles provided by the author, without testing.

field_info

Author :	**Kathleen Dyer**
	Lawrence Livermore National Laboratory
Number :	1117
Language :	C
Submitted :	01/22/92
Version :	1.000
Last Updated :	01/22/92
Ported To :	Kubota DEC IBM Sun HP

Calculates the maxima and minima of input field data and of the grids. If the field has more than three dimensions, grid information is obtained for only the first three. The information is used to produce GEOM text objects. The min/max parameters allow individual objects to be toggled on or off. The "Text Height" parameter sets the height of the GEOM text as a fraction of the window height.

field_to_EXCEL

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1654
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

ASCII Excel spreadsheet format files are produces containing field data in a table format. These files can be imported into a suitable excel format for analysis and plotting. Separator characters may be specified for differing output versions, three possible floating point output formats are selectable, and the maximum number of values per line is adjustable. The module only outputs the file if the file browser changes. This keeps the output file from beeing written too often.

fix_points

Author :	**Larry Gelberg**
	Stardent Computer
Number :	1107
Language :	C
Submitted :	11/27/91
Version :	1.000
Last Updated :	11/27/91
Ported To :	Kubota DEC IBM Sun HP

"orthoslice" and "field to mesh" communicate through a secret code stored in the "points" array of the field structure. This will be an array of 6 floating point numbers representing [minx, maxx, miny, maxy, minz, maxz] for the 2D (3-space) field which orthoslice produces. FIELD TO MESH looks at these searching for when the min == max to determine which direction (I, J, or K) the slice was taken from. This is the means that FIELD TO MESH uses to orient and position the slices properly in the volume. The way to defeat this is to impose your own point information on the existing ones. For instance, you have a 64x64x64 volume and you're taking the 12th K slice. The points array will look like - [0, 63, 0, 63, 12, 12]. If you overwrite the Z mins/maxes with 0.0's, the slice will appear positioned on the XY plane (not 12 units above it). Similarly, the 23rd I slice points array looks like [23, 23, 0, 63, 0, 63] and will produce a slice parallel to the YZ plane at X=23. By overwriting the 23's with 0's, you will get it to lie right on the YZ plane. This should put between the orthoslice and the field_to_mesh modules in your network.

geom_parent

Author :	**Ed Bender**
	Convex Computer Corporation
Number :	1592
Language :	C
Submitted :	03/04/93
Version :	1.000
Last Updated :	03/04/93
Ported To :	DEC IBM Kubota HP

The geom parent module allows an interactive user to reparent any and all objects in the geometry viewer system. This is useful if you want some objects to move together, but not all. The "Group Name" parameter is used to define the group to add objects to. When the "Include Current Object" oneshot is pushed, the current object selected within the geometry viewer is reparented to the "Group Name".

gui_page

Author :	**John Stephen** Tessella Support Services
Number :	1659
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Kubota Sun

To help place, size, and maintain user interface panels and widgets, this module gives dial and text parameter control over the "layout" of any item, without using the layout editor. Once an application is built, this module can will enable the layout to be dynamically changed by the end user, or used to prototype a layout, which can then be saved. Panel visibility can be controlled, as well as parent, place, and size. It used the CLI as the control path.

image_2_icc

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1045
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	DEC IBM Sun HP

Converts its input image to the so-called ICC format. This image format is used by local device drivers which communicate with the Kodak XL7700 printer. The single parameter is a file browser that allows you to specify the name of the ICC file to be created. After the file is written, the filename is reset to NULL. This prevents subsequent changes upstream in the network from automatically triggering the rewriting of the file. A new file is written only when you enter a filename.

image_to_PPM

Author :	**P. Fulconis** CICG computer center of Grenoble University, France
Number :	1772
Language :	C
Submitted :	02/09/94
Version :	1.000
Last Updated :	02/09/94
Ported To :	IBM

This is one of two modules written to translate AVS animations into mpeg format - image_to_YUV and image_to_PPM. Both formats can be read by mpeg encoder. A ppm image will have a memory size twice bigger than the yuv one, but it can be visualised by xv. You will need to make use of mpeg_encode and mpeg_play, both public domain utilities available via anonymous ftp from the University of California at Berkeley. These modules also make use of the San Diego Supercomputer Center's Image Tools, also available via anonymous ftp.

image_to_YUV

Author :	**P. Fulconis** CICG computer center of Grenoble University, France
Number :	1773
Language :	C
Submitted :	02/09/94
Version :	1.000
Last Updated :	02/09/94
Ported To :	IBM

This is one of two modules written to translate AVS animations into mpeg format - image_to_YUV and image_to_PPM. Both formats can be read by mpeg encoder. A ppm image will have a memory size twice bigger than the yuv one, but it can be visualised by xv. You will need to make use of mpeg_encode and mpeg_play, both public domain utilities available via anonymous ftp from the University of California at Berkeley. These modules also make use of the San Diego Supercomputer Center's Image Tools, also available via anonymous ftp.

image_to_can

Author :	**Hideo Miyachi** Kubota Corporation
Number :	1766
Language :	C
Submitted :	12/15/93
Version :	1.000
Last Updated :	12/15/93
Ported To :	Kubota Sun

This module prints out an AVS image to a laser shot printer. You can specify the actual command to print in the parameter "Command". If you specify "cat ¿ /tmp/filename" in "Command", you can save the image in the file in lbp format.

multi_geom_view

Author :	**John Stephen** Tessella Support Services
Number :	1671
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP Sun

This coroutine takes upstream transform events from any geometry viewer scene, and outputs two transformation control ports, such that a small iconic scene can have remote control over two or more main geometry viewer windows. Once the position is defined from the icon window, pushing the "snap" button will send the transformation over to the other window.

multiformat_writ

Author :	**A. Jacobson** UCLA Dept. Pharmacology / Crump Institute
Number :	1850
Language :	C
Submitted :	10/12/94
Version :	1.000
Last Updated :	10/12/94
Ported To :	DEC HP IBM Sun

This module writes an image to disk from AVS in any of a variety of formats. It also includes the multiformat_read code, and both will compile into a single binary. Some of the formats which this module supports are from San Diego Supercomputing Center's image tools package, others are supported from John Cristy's ImageMagick package, still others require tools from GNU's Ghostscript, and the NetPBM package. Depending on how the module is configured, it can write about 30 to 40 image formats, including , AVS X, Alias, Apple Macintosh MacPaint, Apple Macintosh QuickDraw/PICT, CCIR 601, Compuserve Graphics, Flexible Image Transport System (monochrome only), Hierarchical Data File, JFIF Rev1.02, Khoros VIFF, Magick, Mark VanDeWettering's MTV ray tracer image format, Microsoft Windows bitmap, Microsoft Windows cursor, Microsoft Windows icon, PIXAR picture, Photo CD, PostScript, Radiance, Radiance, SDSC Synu, SGI RGB, Sun Icon and Cursor, Sun Rasterfile, Sun TAAC Image, Tagged, Truevision Targa, Utah Run length encoded, Wavefront raster, X Window System window dump, X11 bitmap, X11 pixmap, ZSoft IBM PC Paintbrush, binary or ASCII PBM Portable Bit Map, binary or ASCII PBM Portable Gray Map, binary or ASCII PBM Portable Pixel Map, binary or ASCII PBM Portable aNy Map. This module contains a superset of the formats in the READ_ANY_IMAGE module, with several new formats added and improved support for several of the original file types. Please note that this module calls several executable files, including imconv, convert, and gzip. Platform specific binaries for several of these programs are available from the IAC ftp site with this module. See the .txt file of this module or the file multiio.h for further details on the sources for this module's tools. The tools are stored at the IAC in one large compressed tar file, which is 11MB and is stored at avs.ncsc.org in /misc_tools/multiio_tools/multiio_to ols.tar.Z.

object_opera

Author :	**Kathleen Dyer**
	Lawrence Livermore National Laboratory
Number :	1123
Language :	C
Submitted :	01/22/92
Version :	1.000
Last Updated :	01/22/92
Ported To :	DEC IBM Sun HP

Recreates the functionalities of some of the transformation options available in the geometry viewer. While this may seem redundant at first glance, it has proven to be a very useful module. The module is mainly used in networks where the Layout Editor is used to customize the user interface. The locations and appearances of this module's control widgets can be altered, while their corresponding widgets in the Geometry Viewer cannot. The module has no input or output ports. To include it in a network, simply drag it into the workspace.

object_propertie

Author :	**John Stephen**
	Tessella Support Services
Number :	1674
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

This module creates a menu structure for all or part of the geometry viewer scene, and allows alteration of the object properties by simple color or transparency selection without going to the geometry viewer controls, for a simpler user interface.

object_visibilit

Author :	**John Stephen**
	Tessella Support Services
Number :	1673
Language :	C
Submitted :	09/29/93
Version :	1.000
Last Updated :	09/29/93
Ported To :	HP IBM Sun

This module takes input strings for the name of the object, the name of the view, and has a boolean toggle control to control the visible/invisible, or hide/show on that object in the view. It uses the CLI to perform the control, hence it has no output port.

output_8bit

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1073
Language :	C
Submitted :	10/10/91
Version :	2.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Takes as input an AVS image field and creates a postscript file suitable for printing on a monochrome postscript printer. The postscript file contains a representation of the input color image using the “image” postscript command, using 8bits of gray. The color image is squashed down to 8 bits of gray by weighting each of the color channels in an attempt to approximate luminance. The postscript image is scaled so as to be centered on the page and to occupy as much area on the page as possible. The nice thing about the file produced is that it will work and produce similar results on, for example, a 200dpi device (like an Apple Laserwriter) or on a 600dpi device (like a Verityper). After executing, the module resets the filename parameter to NULL. This has the effect of requiring the user to enter a new filename for each subsequent invocation of this module before executing.

output_a60

Author :	**Wade Smith** Convex Computer
Number :	1190
Language :	C
Submitted :	05/05/92
Version :	1.000
Last Updated :	05/05/92
Ported To :	DEC HP IBM

The output a60 module takes an AVS image data structure as input, converts it into Abekas YUV format and sends it to an Abekas a60 digital disk recorder. The user can specify the hostname of the a60 digital disk recorder on the network, the insert point and the number of times to repeat each frame. The insert point may be entered in any of four time formats (hh mm ss ff, mm ss ff, ss ff, or ff).

output_color

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1072
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun

Takes as input an AVS image field and creates a postscript file suitable for printing on a color postscript printer. After executing, the module resets the filename parameter to NULL. This has the effect of requiring the user to enter a new filename for each subsequent invocation of this module before executing.

point_sample

Author :	**Anonymous** Stardent Computer
Number :	1066
Language :	FORTRAN
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota IBM HP

Extracts a single scalar value out of the input 2D scalar floating point field. The X and Y position parameters set the index point for the sample to be taken from the field. The numerical field value is then sent to stdout, usually into the window where avs was invoked. The one line message consist of the current X and Y sample index position, and the floating point value of the field at that point. The position values are clamped to the field dimensions internally, without regard to the dial min-max. This module does virtually no work, so "immediate" mode is recommended on the dials. The author has used this for debugging other networks and filters, by linking it in with a "t" junction to other more complex networks.

print_iv_pick

Author :	**Ian Curington** Advanced Visual Systems
Number :	1324
Language :	C
Submitted :	08/31/92
Version :	1.000
Last Updated :	08/31/92
Ported To :	DEC HP

Print the contents of the "iv_pick" output port of the AVS4 image viewer module. This shows the results from picking on images with the left mouse button. It is a basic module showing the possibilities, and used for debugging of other image pick application modules. This module relies on features in AVS4, and will not function with earlier releases. This module will be useful for those building interactive image processing systems.

print_real

Author :	**Lars M. Bishop** National Center for Comp. Elect., Univ. of Illinois
Number :	1575
Language :	C
Submitted :	02/11/93
Version :	1.000
Last Updated :	02/11/93
Ported To :	DEC IBM

Print real creates a widget on the control panel that will contain the value of the module's input port. This module is part of the the Real Math module group, which is a single file, process-sharing set of modules that allow basic operations on and between integer and real AVS data types. They have been most useful in dealing with linkage of several AVS parameters. AVS has no simple way of performing math operations on reals, even though it does allow math operations on fields.

print_verts

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1196
Language :	C
Submitted :	05/06/92
Version :	1.000
Last Updated :	05/06/92
Ported To :	DEC IBM Sun Kubota HP

This module inputs any geometry, and outputs to the avs console window all of the vertices in the geometry objects present in the geometry edit list. This module initially can serve as a debugging tool for modules that output geometries. It also serves as an example module on how to process geometries on a module's input port. It most useful purpose however, is probably to extract quantitative information regarding surfaces, from visualization displays.

print_xfrm

Author :	**Ian Currington**
	Advanced Visual Systems
Number :	1069
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Is a diagnostic tool, used to show the contents of "upstream transform" data types, sent upstream for interactive control in AVS3. It uses "user defined data types".

print_xgeom

Author :	**Ian Currington**
	Advanced Visual Systems
Number :	1070
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Is a diagnostic tool, used to show the contents of "upstream geom" data types, sent upstream for interactive control in AVS3. It uses "user defined data types".

psfilter

Author :	**Wes Bethel**
	Lawrence Berkeley Laboratory
Number :	1177
Language :	C
Submitted :	10/10/91
Version :	2.000
Last Updated :	05/19/92
Ported To :	DEC IBM Sun Kubota HP

The "output 8bit ps" module takes as input a scalar (byte, single channel) field and creates a postscript file suitable for printing on a monochrom postscript printer. The postscript file contains a representation of the input color image using the "image" postscript command, using 8bits of gray. The postscript image is scaled so as to be centered on the page and to occupy as much area on the page as possible. The nice thing about the file produced is that it will work and produce similar results on, for example, a 200dpi device (like an Apple Laserwriter) or on a 600dpi device (like a Verityper).

remote_lit_volum

Author :	**Howard Stroyan** Hewlett-Packard Advanced Systems Division
Number :	1825
Language :	C
Submitted :	07/15/94
Version :	1.000
Last Updated :	07/15/94
Ported To :	HP

This AVS module is designed to provide access to the HP specific back-to-front Volume Rendering graphics functionality of the Starbase graphics API. This module can only be successfully built and executed on an HP700 workstation. This module prototypes an ability to make use an HP graphics devices volume rendering capability remotely. The idea is to run AVS on the HP with the X display routed to a networked server, but make use of the HP workstation and one of the above listed display devices to render volume dataset on the HP then send the results (as AVS images) to the remote X display. To use this module you'll need to setup the environment variable COMPUTE_SERVER to point at the X server on the HP server that will be performing the rendering. This includes the same gradient lighting model described for the disp_lit_vol module.

remote_volume

Author :	**Howard Stroyan** Hewlett-Packard Advanced Systems Division
Number :	1826
Language :	C
Submitted :	07/15/94
Version :	1.000
Last Updated :	07/15/94
Ported To :	HP

This AVS module is designed to provide access to the HP specific back-to-front Volume Rendering graphics functionality of the Starbase graphics API. This module can only be successfully built and executed on an HP700 workstation. This module prototypes an ability to make use an HP graphics devices volume rendering capability remotely. The idea is to run AVS on the HP with the X display routed to a networked server, but make use of the HP workstation and one of the above listed display devices to render volume dataset on the HP then send the results (as AVS images) to the remote X display. To use this module you'll need to setup the environment variable COMPUTE_SERVER to point at the X server on the HP server that will be performing the rendering.

statistics

Author :	**Anonymous**
	Stardent Computer
Number :	1082
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota HP

Is a field diagnostic and analysis module, similar to VOLSTAT, used to obtain global statistical information about field data. This module scans the data in the volume, and produces a small table, sent to stdout, usually into the window where avs was invoked, unless redirected. The table includes the following information about a field- Dimensions, Min/Max, Mean, Median, Standard Deviation, Skewness, and Kurtosis.

stereo_lit_volum

Author :	**Howard Stroyan**
	Hewlett-Packard Advanced Systems Division
Number :	1827
Language :	C
Submitted :	07/15/94
Version :	1.000
Last Updated :	07/15/94
Ported To :	HP

This AVS module is designed to provide access to the HP specific back-to-front Volume Rendering graphics functionality of the Starbase graphics API. This module can only be successfully built and executed on an HP700 workstation. This module assumes that a stereo ready monitor and viewing system (i.e. StereoGraphics' Crystal Eyes) are available. It will render a stereoscopic version of the disp_lit_vol module. This module make use of a similar environment variable as the COMPUTE_SERVER variable used by the remote_volume module. This module uses the environment variable STEREO_SERVER. Set this variable to point at the stereo-ready device. [In order to use this module well it is required that a second X display of some kind is available.] Run AVS with the DISPLAY set to the non-stereo screen or server. This module will open a full screen window on the display pointed to by STEREO_SERVER and will place the display in stereo mode (via the Starbase STEREO gescape). Stereo mode can be toggled off and on with the "Stereo" control button. (Turn off the stereo mode before deleting this module or exiting AVS, or your stereo display will be left in stereo mode.)

string_to_file

Author :	**Ian Curington**
	Advanced Visual Systems
Number :	1561
Language :	C
Submitted :	11/16/92
Version :	2.000
Last Updated :	09/30/93
Ported To :	IBM HP

Sends the contents of the string input port to an output file, opened with a file browser. The string can replace the current contents of the file, or be appended to the file since opened during the session, as a log of strings in the network, controlled by a mode toggle. This is designed to be used to send command strings via a named pipe (see mknod pipe.file p) to another application program, where command input has been redirected to the same pipe. This allows an AVS user interface modules to control an external command-line driven application.

ucd_statistics

Author :	**anonymous**
Number :	1895
Language :	C
Submitted :	09/07/95
Version :	1.000
Last Updated :	09/07/95
Ported To :	DEC_ultrix DEC_alpha

Ucd statistics is a diagnostic and analysis module, similar to the field module statistics, used to obtain statistical information about field data. This module scans the data in the volume, and produces a small table, sent to a scrollable text window in the module control panel. The table contains the statistics for each data component calculated separately. Then if there was more than one component, the table contains the statistics for all of the data components grouped together. It will create statistics for both node and cell data. It will also generate overall statistics for node data and cell data separately.

ucd_to_wave

Author :	**Ion Barosan** Eindhoven University of Technology, The Netherlands
Number :	1597
Language :	C
Submitted :	03/04/93
Version :	1.000
Last Updated :	03/04/93
Ported To :	DEC IBM

The ucd to wave module writes a ucd structure to disk, in wavefront format, which is supported by Data Visualizer. Ucd to wave works only with node data, therefore if you have cell data, you have to transform the cell data to node data using the cell to node module. Because Data Visulizer doesn't use mid-edge nodes, at this version, the ucd to wave module generally doesn't use the mid-edge nodes. One exception has been made - if the ucd cell is triangle with all mid-edge nodes present, ucd to wave module will split the triangle in 4 triangles. The user can change the module to support another ucd type which use mid-edge nodes. The wavefront file that will be written to disk is an ASCII file.

unix_command

Author :	**A. Jacobson** UCLA Dept. Pharmacology / Crump Institute
Number :	1831
Language :	C
Submitted :	07/16/94
Version :	1.000
Last Updated :	07/16/94
Ported To :	DEC HP IBM Kubota Sun

Often there will be a need to convert a string in AVS to a command or argument in unix to be performed at the command line. This module allows the user to output strings and integers to the command line for execution by the shell in the directory that AVS was run from.

volume_stati

Author :	**Ian Curington**
	Stardent Computer
Number :	1091
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota IBM HP

Is a diagnostic and analysis module, used to obtain global statistical information about scalar volume data. Often the result of numerical analysis and visualization leads to an interest in just a few numbers such as the average value in the field. This "volume statistics" module scans the data in the volume, and produces a small table, sent to stdout, usually into the window where avs was invoked, unless redirected. The table shows the dimensional size of the field, total number of data points, the minimum and maximum scalar values, and the mean. This is a very simple module, and could be extended to measure other parameters such as standard deviation, median, etc. This module does virtually no computationally intensive work, has no parameters. The author has used this for debugging other networks and filters, by linking it in with a "t" junction to other more complex networks. In particular, when using "interpolate", it is nice to know the resulting volume dimensions.

write_A60_yu

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1106
Language :	C
Submitted :	11/27/91
Version :	3.000
Last Updated :	02/09/94
Ported To :	DEC IBM Sun HP

Intended to be used with the AVS Animator application. The module hooks right up to the Animator module or the read_frame_seq module. This module accepts images, one at a time. The image is then converted into the Abekas A60 yuv format - with the same dimensions of the Abekas A60 yuv format - and output to a file in the Frame directory with a name corresponding to the frame number. (i.e. Frame Number 20 – Filename - 20) If the image is not the same size as the A60 image standard, then the AVS image is centered within the view of the Abekas. This works well when you consider the degradation of the image borders on its path to NCSC. This module was adapted from Ian Currington's write_abekas module and Abekas Inc.'s sample to_yuv program in their manual. It writes out a temporary file in between conversions. The module works fine - but it isn't pretty. Version 3 update provided by the UCLA School of Medicine's Andy Jacobson. The update includes bug fixes, and features wherein the user can name the file as with the animate_filename module, but with a fixed .yuv siffix. The user also has the choice of compressing the (huge.) output files. In some nets it may be desireable to allow or prevent overwriting of files, so a switch is now provided for this. Also you can supply your favorite compression program to this module, if desired.

write_Dore_i

Author :	**Anonymous** Kubota Pacific Computer Inc
Number :	1154
Language :	C
Submitted :	03/12/92
Version :	1.000
Last Updated :	03/12/92
Ported To :	Kubota DEC IBM Sun HP

Interprets AVS image data into Dore format and writes it out to a new file.

write_KSWAD

Author :	**Anonymous** Kubota Pacific Computer Inc
Number :	1157
Language :	C
Submitted :	03/12/92
Version :	1.000
Last Updated :	03/12/92
Ported To :	Kubota HP

Interprets AVS image data into KSWAD format and writes it out to a new file.

write_MooV

Author :	**Charles J Williams III** Naval Research Lab
Number :	1544
Language :	C++
Submitted :	10/14/92
Version :	1.000
Last Updated :	10/14/92
Ported To :	Sun

This module is a first shot at incorporating QuickTime(tm) into AVS as a method for producing convenient, compact, and high quality video segments. QuickTime(tm) is rather convenient since there is likely to be more support on other systems, and there is a growing base of QuickTime(tm) cheaply available for the Macintosh. This code currently requires, the QuickTime Movie Exchange Toolkit, which is available from Apple, Inc, and a C++ compiler (you also need the C++ compiler to build the QuickTime(tm) libraries. Please note - since the IAC doesn't have the Exchange Toolkit, we weren't able to test this module out here.

write_abekas

Author :	**Ian Currington** Stardent Computer
Number :	1076
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

This module provides I/O capability to images in the native machine format used by the abekas A60 digital video disk recording system. The image file specification is for the PAL (50 Hz, 625 line) version, but can easily be modified to support the NTSC version, or both. The abekas A60 holds 750 frames, or 30 seconds of live broadcast standard video imagery in the digital storage format dictated by the CCIR-601 specification. Although the disk format is Y-uv, the disk has firmware processing to encode/decode to 3-byte Red-Green-Blue images on-the-fly during tcp/ip file transfers.

write_compressed

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1162
Language :	C
Submitted :	03/15/92
Version :	2.000
Last Updated :	05/19/92
Ported To :	Kubota IBM Sun DEC HP

The “write compressed image” module provides the same functionality as the AVS-supplied “write image” module, with the added twist that the image is in a “compressed” format. An AVS image may be compressed using the UNIX(tm) compress utility, or by using the sibling module “write compressed image.“ Compression ratios of 1 to 10 are easily achieved using this compression scheme.

write_eps_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1239
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the eps format from an AVS Network. Encapsulated PostScript color, grayscale, and black-and- white image files are used by a variety of PostScript tools in order to include PostScript diagrams and images within other documents. Encapsulated PostScript files are recognized by the follow- ing filename suffixes .eps, .epi, .epsi, and .epsf.

write_g_2

Author :	**John Langner**
	Stardent Computer
Number :	1294
Language :	C
Submitted :	07/11/92
Version :	1.000
Last Updated :	07/11/92
Ported To :	IBM Sun HP

This is yet another module to write a Compuserve GIF image file.

write_geom

Author :	**Georg Thallinger**
	Joanneum Research Graz, Austria
Number :	1698
Language :	C
Submitted :	09/30/93
Version :	1.000
Last Updated :	09/30/93
Ported To :	IBM

The write_geom module saves the geometry in a file. Currently there is a choice of three formats - avs-text, avs-binary, dxf (drawing exchange format by AUTODESK). The dxf-routines supply a rudimentary header, every object is placed on a single layer (PRISMS converts every layer in an object). The dxf-output is at last readable by PRISMS, ALIAS and of course AutoCAD.

write_gif_image

Author :	**Terry Myerson**
	International AVS Center (NCSC)
Number :	1240
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the gif format from an AVS Network. GIF (Graphics Interchange Format) is CompuServe's standard for generalized color raster images. This standard is a mechanism to exchange and display high-quality, high- resolution graphics images. CompuServe gif files are recognized by the filename suf- fixes .gif and .giff.

write_hdf_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1241
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the hdf format from an AVS Network. hdf is a generic, tagged Hierarchical Data File format developed by the National Center for Supercomputing Applica- tions (NCSA). hdf files may contain images, scientific data sets, and miscellaneous data items. Such files may be created by several NCSA tools. See the NCSA documentation for details on how to use these tools. hdf format handling within the SDSC image library is limited to images of certain depths and storage methods. hdf files are recognized by these filename suffixes .hdf, .df, and .ncsa.

write_icon_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1242
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the icon format from an AVS Network. icon image files are used by Sun Microsystem's SunView win- dow system, NeWS window system, OpenWindows NeWS tool set, and X11 XView tool set for the storage of icons, cursors, fill patterns, and pieces of widgets (like button check- marks). Sun icon files can be most easily generated using Sun's iconedit(1) icon and cursor editor. The Sun operating sys- tem release includes a directory of standard icons, cursors, background patterns, and widget pieces in icon format in the directory /usr/include/images. See the Sun documentation set for details on how to use the tools dealing with Sun icon files. Sun icon files are recognized by these filename suffixes .icon, .cursor, and .pr.

write_iff_image

Author :	**Terry Myerson**
	International AVS Center
	(NCSC)
Number :	1264
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the iff format from an AVS Network. iff image files are generated by Sun Microsystems TAAC software such as voxvu(1) and cloudvu(1). See the TAAC-1 Application Accelerator Software Reference Manual for information on how to use these programs. Note that image files compressed with the Sun-TAAC utility make_movie(1) cannot be read by the SDSC image library. iff files are recognized by the following filename suffix .iff.

write_irreg

Author :	**Wes Bethel**
	Lawrence Berkeley
	Laboratory
Number :	1092
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Writes an ASCII representation of an input irregular field to a file. The file format is the “LBL” irregular field file format, and is readable by the “read irreg” input module. This module is useful for looking at irregular fields, for example, after some processing step in a data flow network. A browser is supplied to let the user specify the filename to which the ascii representation will be written.

write_jpeg

Author :	**John Langner**
	Stardent Computer
Number :	1102
Language :	C
Submitted :	11/22/91
Version :	1.000
Last Updated :	11/22/91
Ported To :	IBM Sun HP

Compresses an image with the JPEG compression standard and writes to a file.The best and most readily available introduction to the JPEG compression algorithm is Wallace's article in the April '91 CACM- Wallace, Gregory K. "The JPEG Still Picture Compression Standard", Communications of the ACM, April 1991 (v. 34 no. 4), pp. 30-44. (Adjacent articles in that issue discuss MPEG motion picture compression, applications of JPEG, and related topics.) I highly recommend reading that article before looking at any of the JPEG code. For more detail about the JPEG standard you pretty much have to go to the draft standard, which is not nearly as intelligible as Wallace's article. The current version is ISO/IEC Committee Draft CD 10918-1 dated 1991-03-15.

write_mpnt_image

Author :	**Terry Myerson**
	International AVS Center (NCSC)
Number :	1243
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the mpnt format from an AVS Network. The MacPaint mpnt file is the standard Apple Macintosh mono- chrome bitmap image file format. It can be read by many Macintosh graphics applications, and many Macintosh applica- tions that can export bitmap graphics do so in the MacPaint mpnt file format. mpnt files are recognized by these filename suffixes .mpnt, .macp, and .pntg.

write_p_2

Author :	**John Langner**
	Stardent Computer
Number :	1293
Language :	C
Submitted :	07/11/92
Version :	1.000
Last Updated :	07/11/92
Ported To :	IBM Sun HP

Writes a PC Paintbrush PCX image file. Note that a PCX file can contain a maximum of 256 colors. For maximum flexibility, a separate module is used to pick those colors.

write_pbm_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1244
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the pbm format from an AVS Network. pbm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. pbm files are recognized by the filename suffix .pbm.

write_pcx_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1245
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the pcx format from an AVS Network. The pcx image file format was invented by ZSoft for use in its IBM PC PC Paintbrush tool series. It has become a defacto standard in the IBM PC world and is regularly used for the storage of monochrome and color pixel information by paint-type tools. See the documentation for each of the IBM PC tools for details on how to use them. ZSoft pcx files are recognized by these filename suffixes .pcx and .pcc.

write_pgm_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1246
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the pgm format from an AVS Network. pgm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. pgm files are recognized by the filename suffix .pgm.

write_pic_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1247
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the pic format from an AVS Network. pic image files are generated by PIXAR programming software, such as ChapVolumes and ChapReyes, the PIXAR Image Runtime Library called Pirl, and the PIXAR rendering tool RenderMan. See the PIXAR documentation set for details on how to use these applications and tools. PIXAR's pic file format is sometimes referred to as PICIO in PIXAR documentation. pic and PICIO mean the same thing. PIXAR's xpic is not the same as pic and is not supported by the SDSC image library. Note PIXAR's RenderMan always saves its image files with .pic filename suffixes. However, depending upon output defaults, RenderMan can generate .pic files with PICIO (same as pic) data or TIFF data. .pic files with TIFF data will confuse the SDSC image library. For you to avoid generating this type of file, we recommend that you configure RenderMan defaults to generate .pic files with PICIO data. PIXAR pic files are recognized by the following filename SDSC Last change March 4, 1991 1 IMPIC(3IM) SDSC IMAGE LIBRARY IMPIC(3IM) suffixes .pic, .picio, and .pixar.

write_pict_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1248
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the pict format from an AVS Network. The PICT file is the standard Apple Macintosh image file format. It can be read by almost any Macintosh graphics application. Most Macintosh applications that can export graphics do so in the PICT file format. pict files are recognized by these filename suffixes .pict and .pict2.

write_pict_seque

Author : **Michael Rangitsch and Terry Myerson**
Dow Chemical and International AVS Center
Number : 1577
Language : C
Submitted : 02/11/93
Version : 1.000
Last Updated : 02/11/93
Ported To : HP IBM Sun

This module writes a series of images in the pict format from an AVS Network. The file names are constructed from the base file name and the frame number as such ¡base file name¿.¡frame number¿.pict The frame number is padded to the left with "0" to allow proper sequencing of frames. The intent of this is to allow conversion of a sequence of images into the pict format for video production.

write_pix_image

Author : **Terry Myerson**
International AVS Center (NCSC)
Number : 1249
Language : C
Submitted : 05/21/92
Version : 1.000
Last Updated : 05/21/92
Ported To : Sun DEC IBM

This module writes an image in the pix format from an AVS Network. pix image files are generated by the rendering and painting tools of Alias Research, Inc., such as renderer, raytracer, and paint. See the Alias documentation set for details on how to use these tools. _ΦN_Φo_Φt_Φe Alias quickpaint, available on Silicon Graphics, Inc., IRIS workstations, uses Silicon Graphic's rgb image file format rather than the Alias pix image file format. See the imrgb(3IM) man page for details on the rgb format. Alias pix files are recognized by these following filename suffixes .alias and .pix.

write_pnm_image

Author : **Terry Myerson**
International AVS Center (NCSC)
Number : 1250
Language : C
Submitted : 05/21/92
Version : 1.000
Last Updated : 05/21/92
Ported To : Sun DEC IBM

This module writes an image in the pnm format from an AVS Network. pnm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. pnm files are recognized by the filename suffix .pnm.

write_ps_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1251
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the ps format from an AVS Network. PostScript color, grayscale, and black-and-white image files are used by a variety of PostScript laser printers and windowing system tools. PostScript files are recognized by the following filename suffixes .ps and .postscript.

write_ras_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1252
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the ras format from an AVS Network. ras image files are used by various Sun Microsystems Inc. tools, such as screendump(1) and screenload(1). See the Sun documentation set for details on how to use these tools. Sun ras files are recognized by any of the following filename suffixes .ras, .scr, .sr, and .sun.

write_rectilinea

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1296
Language :	C
Submitted :	08/04/92
Version :	1.000
Last Updated :	08/04/92
Ported To :	DEC IBM Sun HP

This module write data in ascii format from a "rectilinear field" data structure. The disk file is structured to contain information about the field at the beginning of the file, followed by the data.

write_reg

Author :	**Wes Bethel** Lawrence Berkeley Laboratory
Number :	1093
Language :	C
Submitted :	10/10/91
Version :	1.000
Last Updated :	10/10/91
Ported To :	Kubota DEC IBM Sun HP

Writes an ASCII representation of an input uniform field to a file. The file format is the "LBL" uniform field file format, and is readable by the "read uniform" input module. This module is useful for looking at uniform fields, for example, after some processing step in a data flow network. The uniform input field can be any type, size, and dimensionality. A browser is supplied to let the user specify the filename to which the ascii representation will be written.

write_rgb_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1253
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the rgb format from an AVS Network. rgb image files are generated by Silicon Graphics, Inc., software such as icut(1) and snapshot(1). See the Silicon Graphics documentation for information on how to use these and other Silicon Graphics programs. Silicon Graphics rgb files are recognized by the following filename suffixes .rgb, .iris, and .sgi.

write_rgb_sequen

Author :	**Michael Rangitsch and Terry Myerson** Dow Chemical and International AVS Center
Number :	1576
Language :	C
Submitted :	02/11/93
Version :	1.000
Last Updated :	02/11/93
Ported To :	HP IBM Sun

This module writes a series of images in the rgb format from an AVS Network. The file names are constructed from the base file name and the frame number as such ¡base file name¿.¡frame number¿.rgb The frame number is padded to the left with "0" to allow proper sequencing of frames. The intent of this is to allow conversion of a sequence of images into the rgb format for video production.

write_rla_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1254
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the rla format from an AVS Network. rla image files are generated by Wavefront Technologies, Inc., software such as image and tdv. See the Wavefront documentation for information on how to use these and other Wavefront programs. rla is a subset of the newer Wavefront rlb specification. Programs that can read and write rlb files can also read and write rla files. The SDSC image library reader/writer is written to accommodate both types of files. Wavefront rla files are recognized by the following filename suffixes .rla and .rlb.

write_rle_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1255
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the rle format from an AVS Network. rle image files are generated by the tools of Utah's Raster Toolkit. See the Utah documentation set for details on how to use these tools. Utah rle files are only recognized by the following filename suffix .rle

write_rpbm_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1256
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the rpbm format from an AVS Network. rpbm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. rpbm files are recognized by the filename suffix .rpbm.

write_rpgm_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1257
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the rpgm format from an AVS Network. rpgm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. rpgm files are recognized by the filename suffix .rpgm.

write_rpnm_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1258
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the rpnm format from an AVS Network. rpnm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. rpnm files are recognized by the filename suffix .rpnm.

write_rppm_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1259
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the rppm format from an AVS Network. rppm image files are used by various tools in Jef Poskanzer's PBM+ tool suite. See the PBM documentation set for details on how to use these tools. rppm files are recognized by the filename suffix .rppm.

write_synu_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1260
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the synu format from an AVS Network. synu is the image file format output by SDSC's synu (Syn- thetic Universe) portable renderer. synu files are recognized only by the following filename suffix

write_tiff_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1261
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the tiff format from an AVS Network. tiff is a generic Tagged Image File Format developed by Aldus and Microsoft in conjunction with leading scanner and printer manufacturers. tiff files may contain images and miscellaneous other image-related items. Such files may be created and manipulated by a variety of Tagged Image File Format tools. See the Tagged Image File Format documentation for details on how to use these tools. tiff support within the SDSC image library is limited to images of certain depths and storage methods. tiff files are recognized by the following filename suf- fixes .tiff and .tif.

write_xbm_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1262
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the xbm format from an AVS Network. xbm bitmap image files are generated by MIT's X Window Sys- tem, version 11 (hereafter referred to as X11). xbm files are generated by the X11 bitmap(1) bitmap editor and used by most X11 tools to define cursors, icons, and other monochrome glyphs. See the X11 documentation set for details on how to use tools and subroutines dealing with X11 bitmaps. X11 xbm files are recognized by the following filename suf- fixes .xbm and .bm.

write_xwd_image

Author :	**Terry Myerson** International AVS Center (NCSC)
Number :	1263
Language :	C
Submitted :	05/21/92
Version :	1.000
Last Updated :	05/21/92
Ported To :	Sun DEC IBM

This module writes an image in the xwd format from an AVS Network. xwd window dump image files are used by the xwd(1) and xwud(1) tools of MIT's X Window System, version 11 (hereafter referred to as X11). See the X11 documentation set for details on how to use these tools. X11 xwd files are recognized by the following filename suf- fixes .xwd and .x11.

Catalog

Data Input Modules 0

3D_axis	ACE_to_UCD	AFS_extract	AMS_extract	ANIMATE_DUAL
ARC_MAP	Application_Menu	BGF_Protein_Disp	BP_Coroutine	ConstImage
DMOL_to_field	Dials_Contro	Dials_Control	Dipole	Edit_File
Edit_String	Extrusion	FIND_DATA_AND_MA	FIND_MAP	FITS_read
GET_MODEL_MAP	Generate_label	Get_Terrain_Data	Graph_slider	Image_Sequen
LBF_extract	Label_Axis	Life	Life_WT	Load_Simulation
MOSS_LINES	MOSS_Markers	MOSS_POLYGONS	Menger_sponge	Mouse_Trap
Mouse_Trap_IV	QUAL_field	RADM_Credits	READ_ANY_IMAGE	READ_TIMESTAMP
READ_TIN	READ_WIND_TIMESE	RdUcdAnim	ReadLat	ReadRadiance
ReadSeqAnim	Read_2	Read_CFD	Read_DXF	Read_EJ65
Read_Hologram	Read_SDRC	Read_SEG_Y	Read_Unversal_FI	Sin_Wav
Sonify	Splash	Tick_Marks	Vector_Label	Vernier_Dial
WriteSeqAnim	Zoomer	animate_file	animate_floa	animate_inte
animated_boolean	animated_str	animated_track	application_mana	awais
cabinet_maker	ccyl_pipe	choice_2	choice_alias	choice_selec
color_cube	color_palette	component_menu	compress_file	coord2geom
crystal	curdle	cylinders	draw_arrow	draw_polygon
endif	extruder	fast_animate	field2_from_Math	field3_from_Math
file_path_and_na	file_to_field	flight_path	fractal_3d_f	fractal_field
gaussian_field	generate_contine	generate_gri	genloadmd	geom_from_Math
geometry_sphere	graduate32	grid_generator	harwell_colo	if_2
image_snap	int_positiv	integer_list	integer_step	jigsaw
lathe	lines_to_geom	load_md	load_mdn	macro_menu
mandelbrot	mask_generator	menu_example	microscope	mk_vel2
mk_vel	multiformat_read	named_toggle	newfilla	noise_source
pdb_to_geom	phoenics_int	pick_center	place_camera	plate
plato_super	plot_xyz	plot_xyz_col	r_tif_ra	r_tifvol_ra
readFLOW3D	read_16_bit	read_ACR_NEMA	read_AIFF	read_Dore_im
read_F3D	read_FLOW3D	read_GOCAD	read_HDF_SDS	read_Leica_confo
read_abekas	read_blokjes	read_compressed	read_dem	read_dyna3d
read_eps_image	read_gif2	read_gif	read_gif_image	read_grd_to_fiel
read_hdf_image	read_icon_image	read_iff_image	read_irregul	read_jpeg
read_mpnt_image	read_netcdf	read_netcdf_example	read_p_2	read_pbm_image
read_pcx_image	read_pgm_image	read_pic_image	read_pict_image	read_pix_image
read_plot3d	read_pnm_image	read_points	read_ras_image	read_rectilinear
read_rgb_image	read_rla_image	read_rle_image	read_rpbm_image	read_rpgm_image
read_rpnm_image	read_rppm_image	read_semper	read_shak	read_sunras
read_synu_image	read_tiff_image	read_tiff_sectio	read_tiff_vol	read_ucd_points
read_uniform	read_xbm_image	read_xwd_image	river_bed	shaker
sololoadmd	spaceball_master	spaceball_slave	sphere_to_geom	string_example
string_list	string_macros	strip_chart	taperer	teapot
track_cursor	transform_xyz	tree	trigger_int	trigger_int_2
ucd_color_ra	ucd_comp	uncompress_file	uv_rect	uv_sector
waves_demo				

Filter Modules 0

4D_to_3D_Sli	ColorToBW	Color_Legend	Cylinder	Delta_Int
Downsize_2	E01SAF	Extract_label	Field_Concat	Field_Formula
Field_Math	GRID_MATH	GeomXYZtoFld	Geom_Dup_n_Mirro	Geom_Duplicate
Geom_Unique_Name	Geos_Image	IDL	IMAGE_WINDOW	Input_Formula
Int_to_string	Intervals_Field	Intervals_UCD	Iterate	LatLon_UTM
MINMAX_2_3	MOVING_AVG	MathAVS	MinMax_2	Nvect_to_2d
Quantize_Image	Reset_Legend	SIDE_BY_SIDE	Shadow	SlowDown
Sobel_operat	Sphere	Stepper	Toggle_Black_Whi	Topo2irreg
UcdColorizeGeom	UcdXYZtoFld	Unjaggy	WINDOW_MATH	WIND_MINMAX
WIND_SPEED	X_Crop	X_ray	Z_slicer	aCNN
add_2_ints	add_coords_2	add_coords_3	add_cps	add_extents
adjust_rgb	alpha_mix	and_or_merge	arith_binary	arith_logical
arith_unary	arithmetic_2	backgroundRGB	byte_image_t	classify
closing	collage	collapse	collapse_2_num	colorizer_2d
colour_cones	combine	combine_vecto_2	combine_vectors	complex_math
compute_shade	concat_str	cone	cone_2	contour_perc
convert_color	convert_data	crop2	cylinder_2	delay
demultiplex	dilate	downsize_points	draw_pixel	dual_control
erode	excavate	exponential	extend	extrude_field
feature	fft2d	field_2D_to_3D	field_averaging	field_control
field_conver	field_label	field_mux	field_scatte	field_to_float_parm
field_to_glo_2	field_to_globe	field_to_int_parm	field_to_irr	field_to_rec
field_to_string_parm	field_to_uni	fill	filter_1Dfreq	filter_1Dtime
filter_2Dfreq	filter_2Dspatl	filter_morph	fix_field_edge	float_image
float_parm_to_field	float_to_str	fourier_filter	fourier_transfor	gamma_correct
geo_control	geomanip	geometrySearch	glow_white	gradual_mix
grey_scales	grid_scale	histogram	hit_lumin_mi	hsv_to_rgb
image_combine	image_flip	image_rotate	image_stack	image_stats
image_tile	info	input_create1D	input_create2D	int_2_st
int_parm_to_field	integer_control	integer_to_real	interp_cmap	irreg_2_reg
last_cone	linearop1D	local_area_stats	log_field	lumin_mix
luminence	matrix_algebra	matte_3x3	matte_sample	min_max_fiel
mix2	modify_seq1D	my_mirror	natural_log	no_format
opening	output_dither	output_print	panel_pan	pencil_sketc
persp_field	real_control	real_to_integer	reconstruct_slic	rect2pol
remap_field	remote_gis	rgb_to_cmyk	rgb_to_hsv	river_fill
scat_to_tri	scat_znmo	scatter_downsize	scatter_to_spars	segment
set_alpha	set_extents	shutter	size_fit	slave_slicer
smooth	specify_slice	spectest1D	standard	stereogram
stick	stream_to_scatte	string_parm_to_field	strip_cfd	subregion
super_sample	surface	switch_2	switchfire	sysModuleName
table_editor	thresholdclamp	transform1D	transform2D	transform_po
transpose_coords	tubeness	ucdTri2prism	ucd_extent_range	ucd_load_node_da
ucd_minmax	ucd_particle	ucd_polar	ucd_select	ucd_split
ucd_to_field	ucd_tri_to_prism	un_jag	unif2irreg	unif2zscat
uniform_to_i	vec2_to_vec3	vec_mag_2d_v	vect_div_scal_2	vect_div_scal_5
vect_div_scalar	vector	vector_cross	vector_cross_2	vector_cross_3
vector_cross_4	vector_cross_5	vector_dot	vector_dot_3	vector_dot_6
vector_field	warp	xform_field	y_join	

Mapper Modules 0

2D_bounds	3D_contour_map	Analog_Clock	CFD_object	CFD_object_conto
CFD_objectline	ChairPlot_2D_Txt	Colored_bounds	Contour_Lines	Contour_Mesh
Draw_Text	Field_To_Lines	Field_To_Points	Field_To_Polygon	Field_grid
Field_to_Isosurf	GRAPH2D	Get_Local_IP_Num	GraphSound	Grid_2_TxtMesh
Isoband_Slice	Isoband_UCD	Isoline_Slice	Isoline_UCD	Keyframe_Ani
LABELS_UNITS	LBL_bezier_v	MASK_TILE	MINMAX_REPORT	Marker_Draw
Marker_Gen	Marker_Merge	Marker_Pick	Marker_Select	NEW_CROP
NEW_LEGEND	NEW_PRINTFIELD	NEW_SLICE	NEW_TILE	Ortho_Texture
Polygon_Volume	Polyline	QUERY_TILE	RESET_2	Reset_Geom
Scatter_Rect	TEXT_TO_GEOM	TIME_SPAN_TEXT	Title	UCD_Node_Contour
UCD_grid	VIEW_SHARE	Vector_Arrow	WIND_PLOT	Wide_Arrow
add_polar_2D	area	bin_field	bivar	carpet
city_scape	color_legend	color_tile	cone_bounding	confocal_slicer
contour_field	contour_to_g	contour_to_posts	crop_and_slice	draw_tracks
field_lines	field_o_matic	field_params	field_size	field_stats
field_to_con	field_to_map	glue	grid_mesh	hog_2D
interpolating_sl	irreg2map	labeler	line_1D	line_probe
local_coord	loop_objects	mesh_axis	mesh_slicer	mesh_to_ucd
mesh_with_base	new_abuffer	new_arbitrar	new_bubbleviz	new_ortho_sl
non_ortho	ov_slicer	plumb_line	probe_path	real_math
rgbtogray	rubberbonds	scat_2d	scat_3d	scat_base_la
scat_bub	scatter_arrows	scatter_bounds	scatter_cubes	scatter_rib_2
scatter_ribbon	show_flow	str_to_geom	strip_charts	surface_mesh
surface_web	symmetrize	texture_mesh	tracer_bounds	transform_tool
trivar	ucd_cell_arrows	ucd_cell_cones	ucd_cell_geometr	ucd_line
ucd_line_2	ucd_planes	ucd_planes_2	ucd_plumbline	ucd_profile
ucd_streak	ucd_to_prim	volume_mesh	waffle	wire_bender
zone_fence	zone_picker			

Data Output Modules 0

Bit_Movie	CREATE_MINI_DATA	CREATE_USER_MINI	Create_MPEG	Draw_Map
GEOM2IV	GenMask	Geom_to_Wavefron	HdB1	Image_to_Postscr
Record_Anima	Spline_Animator	Trace_Render	WRITE_ANY_IMAGE	WRITE_TIMESTAMP
WrUcdAnim	avs2vrml	avs2vrml	comput_vol	compute_volume
continuous_s	create_unix_pipe	deleteGeom	delete_object	dira2avs
display_ARGB_vol	display_lit_volu	display_tracker	display_volume	dump_icc
dump_ps	dump_sunras	field2_to_Math	field_info	field_to_EXCEL
fix_points	geom_parent	gui_page	image_2_icc	image_to_PPM
image_to_YUV	image_to_can	multi_geom_view	multiformat_writ	object_opera
object_propertie	object_visibilit	output_8bit	output_a60	output_color
point_sample	print_iv_pick	print_real	print_verts	print_xfrm
print_xgeom	psfilter	remote_lit_volum	remote_volume	statistics
stereo_lit_volum	string_to_file	ucd_statistics	ucd_to_wave	unix_command
volume_stati	write_A60_yu	write_Dore_i	write_KSWAD	write_MooV
write_abekas	write_compressed	write_eps_image	write_g_2	write_geom
write_gif_image	write_hdf_image	write_icon_image	write_iff_image	write_irreg
write_jpeg	write_mpnt_image	write_p_2	write_pbm_image	write_pcx_image
write_pgm_image	write_pic_image	write_pict_image	write_pict_seque	write_pix_image
write_pnm_image	write_ps_image	write_ras_image	write_rectilinea	write_reg
write_rgb_image	write_rgb_sequen	write_rla_image	write_rle_image	write_rpbm_image
write_rpgm_image	write_rpnm_image	write_rppm_image	write_synu_image	write_tiff_image
write_xbm_image	write_xwd_image			