THE UNIVERSITY
of MANCHESTER

Volume Visualization for Medical Application

University of Manchester, Manchester

United Kingdom

Joanna Leng, Andrew Grant & Steve Larkin

Manchester Visualization Centre
Manchester Computing

Edition 2.0 November1998

Practical Exercises

1. Introduction

The data used in the following examples all come from medical sources and include:

- a 2D slice from 3D volume of MRI scan of a human head (mri.fld);
- a 3D volume of part of the human spine (spine.fld);
- a 3D volume of a CT scan of the human head (head.fld)

These are all in the same directory, a directory off the field directory. The path may vary depending on where AVS/Express is installed. In the path below XP_ROOT is the path inwhich AVS/Express is installed.

- /XP_ROOT/data/field/med

The objectives of these exercises is to look at techniques used to visualize medical data and to investigate the issues which arise when studying medical data. The exercises start with 2D techniques more commonly used in hospitals and then go onto 3D techniques which are not so widely used. It demonstrates pre-processing, image analysis and the difference between volume and surface rendering.

All the examples will use the AVS/Express network editor and will show how to use these techniques from within AVS/Express. All of the modules used will be available in the **main module** library.

2. Techniques

The techniques covered by this tutorial are not extensive but are aimed at giving the user a good grounding. There are a number of exercises and at the start of each one is a brief description of the aim of the exercise.

- Exercise 1 **Overview of AVS/Express**
- Exercise 2 **Orthogonal Slicing**
- Exercise 3 **Animated Orthogonal Slicing**
- Exercise 4 **Threshold Slicing**
- Exercise 5 **Probing The Data**
- Exercise 6 **Image Analysis**
- Exercise 7 **Generating Isosurfaces**
- Exercise 8 **Excavating The Data**
- Exercise 9 **Volume Rendering**

Exercise 1. — Overview of AVS/Express

Aim

If you have worked through the Introduction to AVS/Express workbook you can skip this exercise but it may be useful to refer to some of the examples and diagrams in this exercise.

Here we look at the way to use AVS/Express; the Network Editor (building networks) and the DataViewer (manipulating the visualized image). The DataViewer has special functions of its own as well as the control over the modules used in the network.

To understand AVS/Express we will first build a module to display 2D data. With this we can look at the use of the controls on the DataViewer including rotating and moving the image, scaling, moving lights and altering the Data Map which maps colours onto the image. The DataViewer also offers us control of the modules in the network so we can alter the parameters in Downsize. This alters the resolution of the image so that it can be manipulated more quickly.

Starting AVS/Express

To start the AVS/Express Visualization Edition, type the following command:

vxp

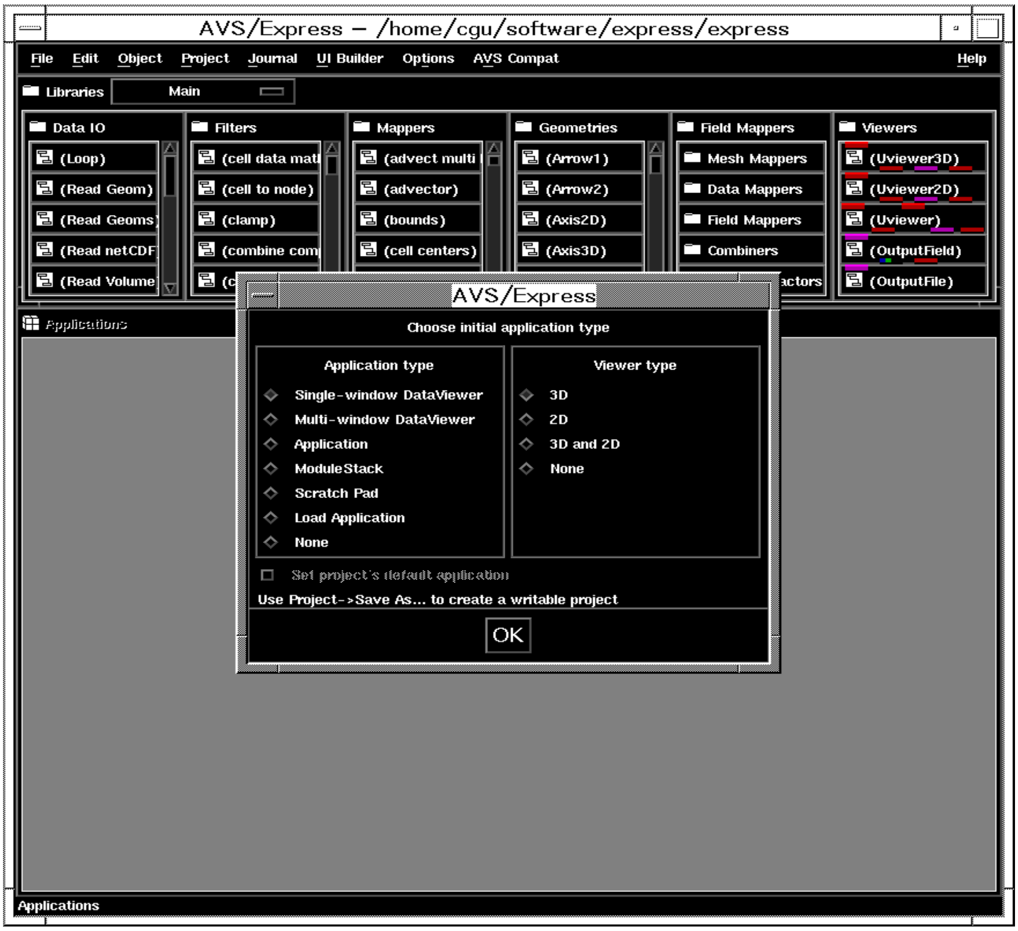


Figure 1: Click on OK and select the 3D DataViewer

When the dialogue box appears click on **OK**. This the dialogue box is giving you the choice of DataViewers, it's default setting is for a **single-window DataViewer with a 3D viewer type,** this is often the best for simple applications. Although in this exercise we will look at a 2D image it is easiest to use the 3D viewer so you can look at its functionality.

The Network Editor

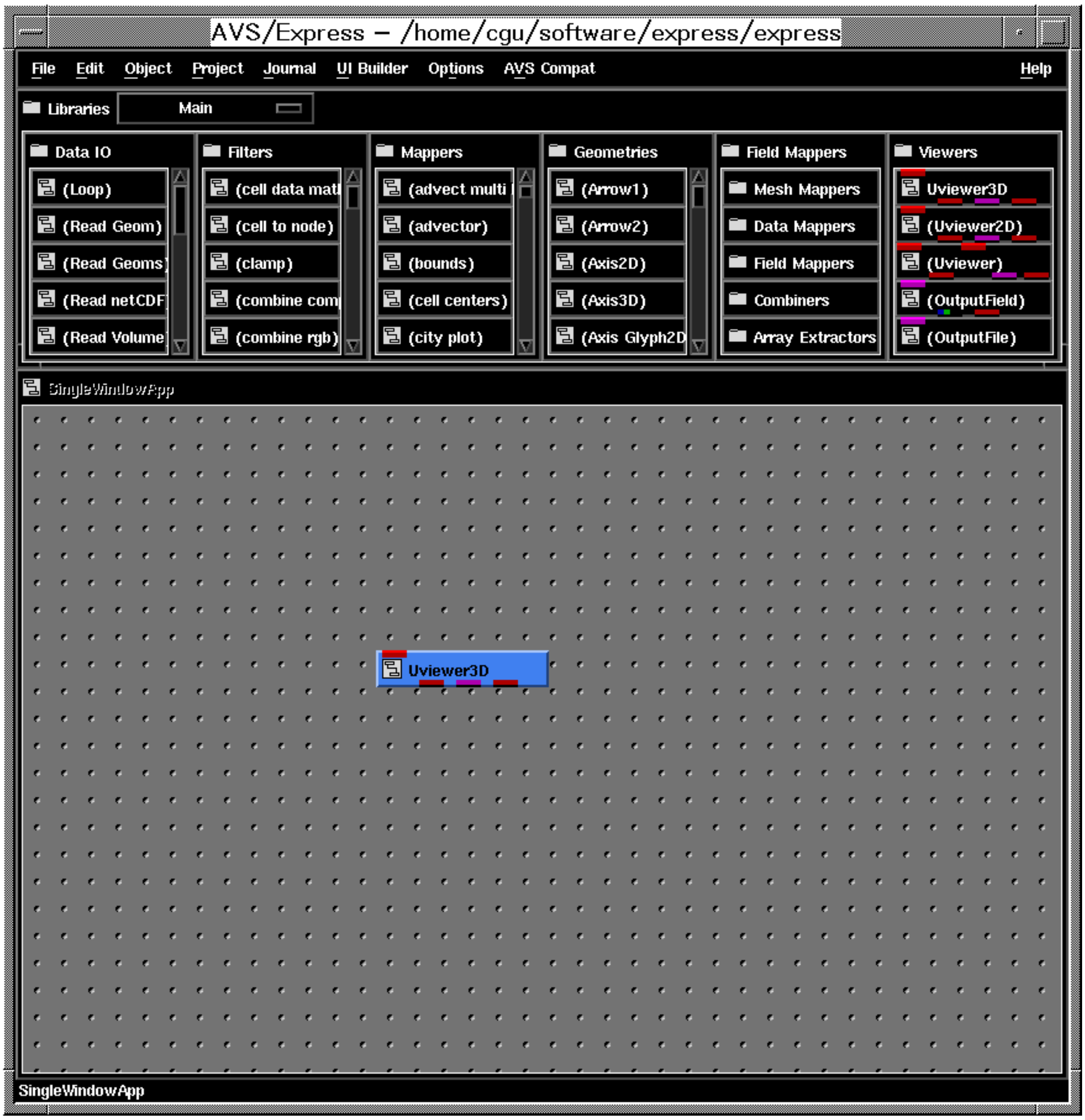


Figure 2: The network editor.

The diagram shows the layout of the of the network editor. The library of modules on view, above the application work area, is called **main.**Within each library area there are all the major modules for building a network. The modules are divided into sub-libraries:

- **Data IO** (this is for reading in data)
- **Filters** (these primarily alter the fields data points)
- **Geometries** (these create simple geometric objects like lines, planes, arrows, and crosshairs)

• **Field mappers** (theses primarily modify the field's mesh)

• **Viewers** (This displays an image of the data)

The sub-libraries correspond to the components of the Haber and Mc Nabb Visualization Pathway. Different libraries can be accessed from the popup menu which appears when main is selected.

To find modules scroll down each sub-library or if it is not obvious which sub-library to look in use the **Object Finder** which is in the **Objects** menu. The wildcard * can be used to increase the flexibility of your search.

To **select a module** we will give an example of the selecting the Read_Volume module needed later:

• Find the Read_Volume module in the **Data IO** library.

• Point to the module using the **left mouse button** and keep this depress as you drag the module into the application workspace.

• Release the left mouse button to instance the module into the workspace.

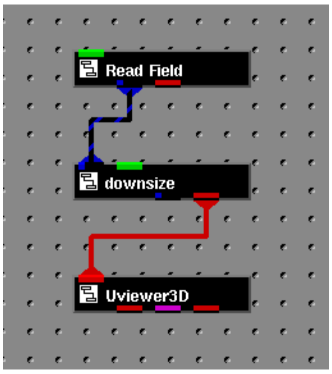
To **delete a module**:

• Select the module by pointing at it and holding down the **right mouse button.**

• A popup menu appears.

• Choose the **delete** option and when the mouse button is released the module will be deleted.

To **build a network:**



• Select and instance the modules given below:

- Read_field (DataIO): reads a AVS field file and converts it into an AVS/Express field
- downsize (Filters): resamples a filed using a scaling factor. Usually the factor is greater than 1 and the size of the field is reduced to save processing time and memory.

- Uviewer3D (Viewers): creates a 3D image in the DataViewer

• **Connecting the modules:**

- To connect module ports together move onto one of the ports that you wish to connect with the mouse.
- Now press the left hand mouse button - all the possible connections are shown.
- While keeping the button pressed move the mouse toward the other module port you wish to make the connection to. The connection should now become highlighted.
- Release the mouse button and the connection is made.

• **Disconnecting the modules**:

- To disconnect the module connection move onto the module which is connected via its output port and "remake" the connection i.e. press the left mouse button while positioned on this port, move toward the other port and when the connection between the two ports becomes highlighted, release the button.
- The connection will be removed.

This covers how a network is built from modules in the Network Editor. Next we need to use the DataViewer to read the data files and manipulate them. First we have an aside to show how the Network Editor is used to save and read networks which you may want to refer to latter but now it can be skipped.

To **save a network**:

- Select the popup menu **File** and select the option **Save Application**.
- The Save Application file browser will appear, move to the right directory-your home directory.
- Type in the network filename that you want in the top text reader. keep the ".v "extension at the end of the file name so network files can be identified.

To **load a network:**

- Select the popup munu **File** and select the option **LoadApplication**.
- The Load Application file browser will appear, move to the right directory where the network is.
- The networks will appear in the file browser and select the correct one.

To **delete a network:**

- Select the popup munu **File** and select the option **Delete Application.**

To **start a new network:**

- Select the popup menu **File** and select the option **New Application.**

The DataViewer

There are three sets of controls we will use in the **DataViewer**:

- Controls of the modules in the Network Editor.
- Controls of the editors that interact with the objects in the DataViewer.
- The Toolbar above the DataViewer that allows quick access to the most commonly used editor's interactions.

Each module that is istanced in the Network Editor has its controls displayed in the editor panel of the Data Viewer Pad.

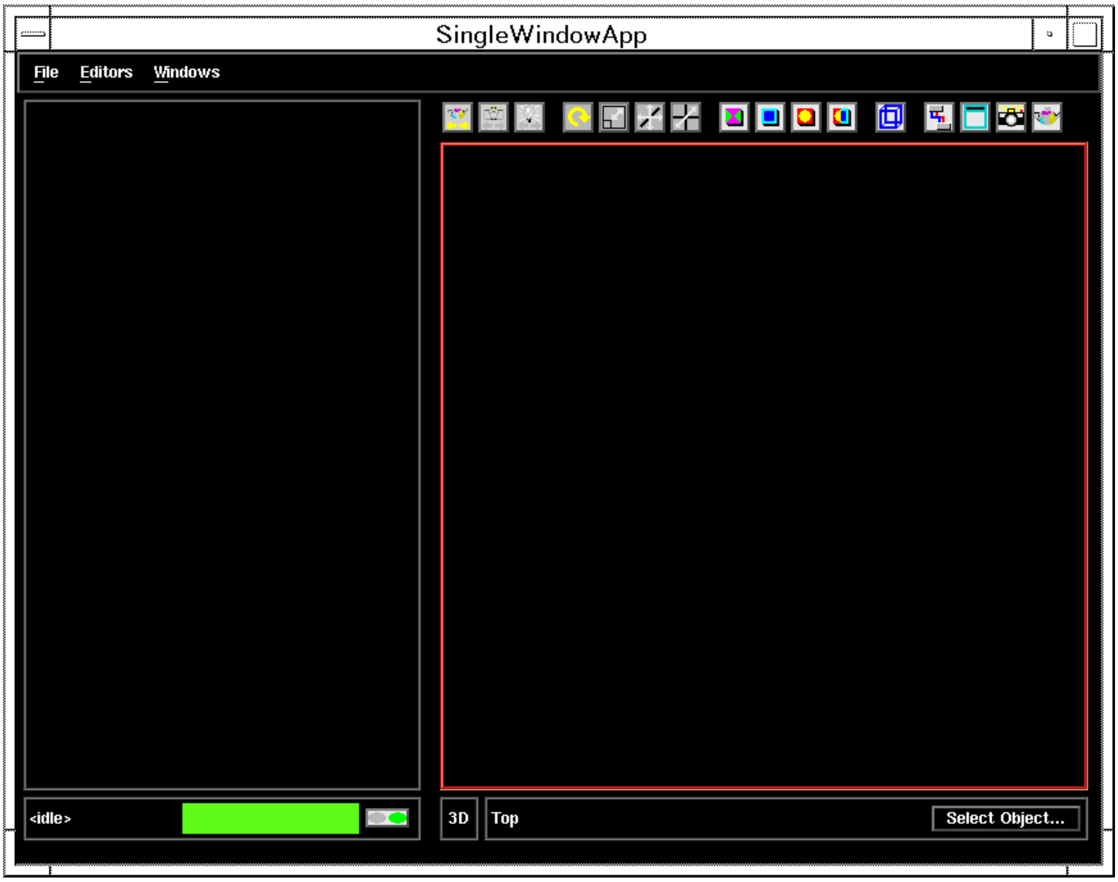


Figure 3: The DataViewer

1. From the **Editors** menu at the top left of the DataViewer select **Modules**. A popup menu will appear with an option for the control panels of the modules in the network. Select **Read field** and use the **file browser** to pick the 2D mri.fld file of the head.

2. An image should appear in the DataViewer. We will use the **Tool Bar menu** to alter the image.

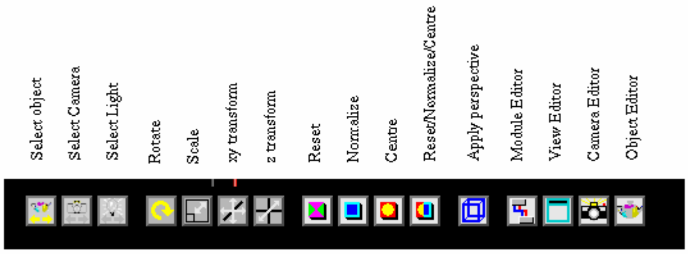


Figure 4: The Tool Bar on the DataViewer.

Using the left mouse key click on the rotate button and then place the mouse over the image press the left mouse button down and drag an arc across the DataViewer. This should start the object spinning, stop it by clicking on the object with the left mouse. All the other functions in the Tool Bar are given in the figure above, try some of these as well.

Before continuing with the rest of the exercise press the Reset/Normalize and Centre button on the Tool Bar.

3. Select the **Modules** option from the Editors menu and then select the **downsize** option. Alter the values on the sliders and see how it affects the resolution in the resulting image.

4. Select **Datamap** from the **Editors** menu. The datamap has several functions not all will be used here but we will return to it. The main purpose of the datamap is to map colours to ranges of data values for any object in the DataViewer. Objects are selected by the using the **Select Object** at the bottom right corner of the DataViewer. Now from the window select the object of interest, in this case **DownSize**.

The current **colour range map** can be seen at the top of the control panel, this can be stored as either of the two major colour models.

- **HSV** (**H**ue **S**aturation **V**alue)
- **RGB** (**R**ed **G**reen **B**lue)

The colour range maps can be either **linear** (interpolated between the max and min values of data) or **stepped** (constant colour for each step). The mapping is set by **control points**.The datamap range and the data range can be altered to give a more precise representation of the data.

We will alter the colour range map by reading in some pre-defined ones.

- In the **Options** popup menu select the **input/output** option.
- In the **Action** popup menu select **Read Datamap**.
- A file browser appears try applying some different colour map ranges.

To get a greyscale image (doctors prefer medical images in greyscale): binary black and white image:

- Read the Greyscale datamap in from the Read Datamap window.

To convert this to a binary black and white image:

- From the options menu select **Edit Colour**.
- Change the **Colour mapping range** from **linear** to **constant**. A binary map should be obtained as

shown above.

- Select **Edit Range/Data** from the **Options** menu.
- Select **sub-range values** and alter the values of max and min to alter the position of the black/white division.

Exercise 2. Orthogonal Slicing

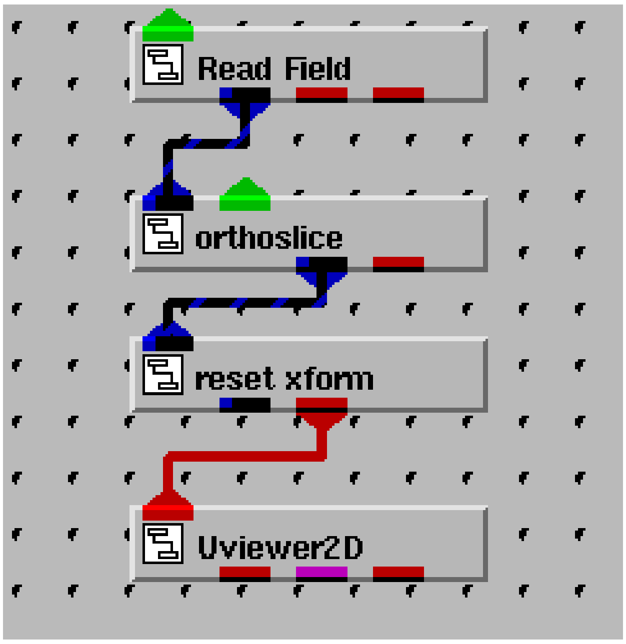
Aims

Medical data often comes in the form of regular arrays of 3D scalar values. Examples of such data sets include CT scan data and MRI scan data. In trying to interpret such data it is often useful to be able to look at a 2D slice through the data volume. The orthogonal slice module in AVS allows the user to generate a slice through the data which is orthogonal to the X, Y or Z axis.

Currently orthoslices are one of the most common methods that medical images are viewed, although they may be tiled out. Orthoslices are slices through the data that are orthogonal to the principal axis.

Orthogonal Slicing

Build or read the following network (**/usr/networks/ex2/ortho2d.v**):



The modules used are:

- Read_Field (Data IO): reads an AVS field file and converts it into an AVS/Express field.
- Orthoslice (Mappers): used to view 3D uniform data. Shows all the 2D slices of data for each of the principal axes
- reset_xform (Filters): reset the transform on a field to an identity.
- Uviewer2D (Viewers): produces a 2D image in the DataViewer.

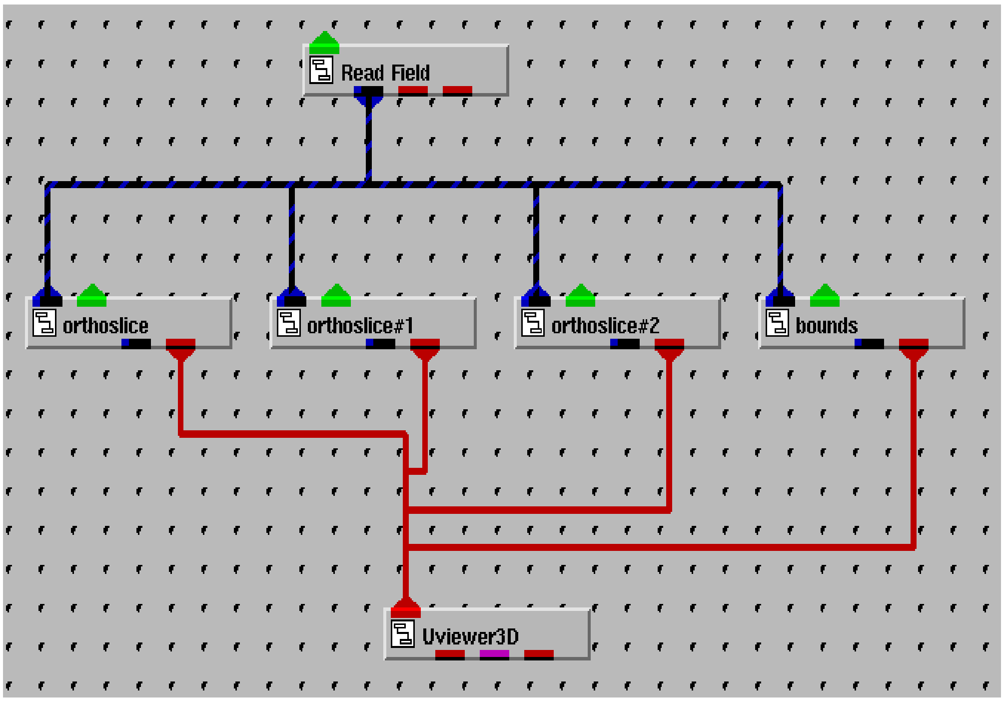
1. Execute the network.

Select the module's **read_field** control panel in the DataViewer and read in the field file `head.fld`. Select the **orthoslice** control panel and alter the **axis slider** and the **plane slider**, you will see that you are able to scroll through the volume so you can examine any feature in the volume.

You can change the values on sliders by dragging the slider through the data by holding down the left hand mouse button and dragging it or you can go through the slices one at a time by pointing the mouse at the arrows at the ends of the slider and clicking the left mouse button.

This is a 2d technique and it can be difficult to know where the slices are in the volume. The technique can give 3d information by having 3 orthoslices, one for each axis viewed in the volume at the same time.

Build or read the following network (**/usr/networks/ex2/ortho3d.v**):



The modules used are:

- Read_Field (Data IO): reads an AVS field file and converts it into an AVS/Express field.
- Orthoslice (Mappers): used to view 3D uniform data. Shows all the 2D slices of data for each of the principal axes
- Bounds (Mappers): generates lines or surfaces that show the bounding box of a 3D structured field.
- Uviewer2D (Viewers): produces a 2D image in the DataViewer.

1. Execute the network.

Again select the module's **read_field** control panel in the DataViewer and read in the field file `head.fld`. Select the **orthoslice** control panel and alter the **axis slider** to 1 then select the **orthoslice#1** control panel and alter the **axis slider** to 2. Now manipulate the view alter the plane values to see how these 2D images are related.

You can manipulate the view by selecting options from the **Tool Bar** menu. Try the different options on the

bounds control panel.

Now press the **Reset/Normalize/Centre** icon from the Tool Bar. You can view the data along its other principal axes by selecting the **Editors** popup menu and selecting **Transforms**. The **Transforms** control panel will appear and by selecting the **Interactor Behavior** button at the top of the control panel a popup menu appears, select the other option **Transform**. This control panel lets you specify exact transforms on the object in the DataViewer. We only need the three buttons **Front, Top** and **Side** which run across the top of the panel. Find out which axis in the orthoslicer corresponds to the Front, Top and Side buttons.

Exercise 3. Animated Orthogonal Slicing

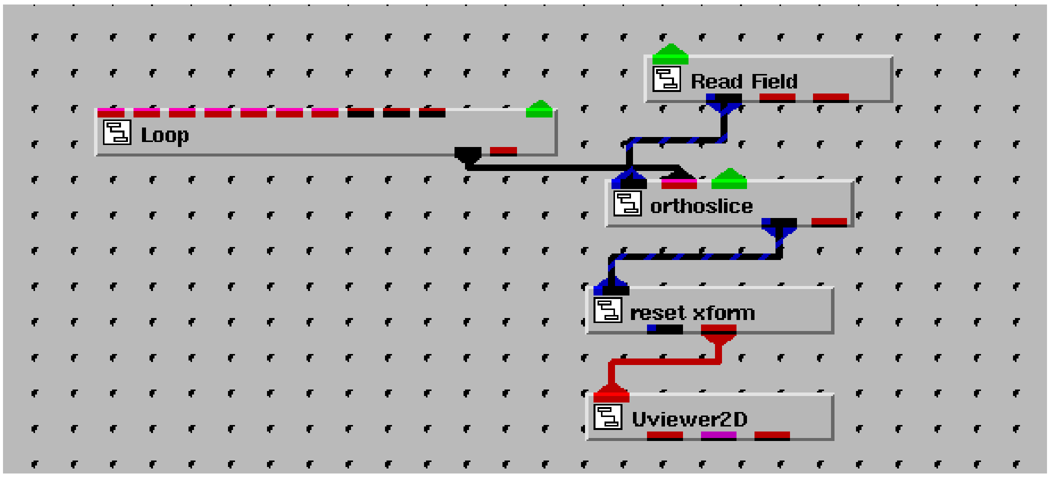
Aim

Often one slice is not enough to determine the features of the data and it is required to play through a sequence of slices to look at changes in the data in a particular dimension.

This exercise will automate the process of stepping through the slices in the volume.

Animated Orthogonal Slicing

Build the network and follow the steps below. Alternatively read the network from **/usr/networks/ex3/animated.v**:



The modules used are:

- Read Field (Data IO): reads an AVS field file and converts it into an AVS/Express field.
- orthoslice (Mappers): used to view 3D uniform data. Shows all the 2D slices of data for each of the principal axes
- reset_xform(Filters): reset the transform on a field to an identity.
- Loop (Data IO): It provides a way to iteratively execute an object.
- Uviewer2D (Viewers): produces a 2D image in the DataViewer.

1. Re-create the initial network used in exercise 2 and bring down the **loop** module from the data input modules menu.There is no external port to connect the **Loop** module to the **orthoslice** module we must make the **planes** parameter external to the **orthoslice** module.

2. In the network editor double click with the left mouse button on the **orthoslice** module, this opens the module. Move the mouse over the left hand side port on the **plane sub-object** and press down the right hand mouse button. A **pop up menu** appears select **Export Port**. Now double click with the left mouse on the **plane sub-object** to open. Delete the text and close the plane sub-object by double clicking on it with the left mouse button again. Close the **orthoslice** module in the same way.

3. Connect the output port from **Loop** into the new input port in **orthoslice.**

4. Read in one of the 3D data sets and then set the control panel for the **orthoslicer** so that the **axis slider** is set to **2** and a image can be seen in the viewer.

5. To animate the orthoslice module select the control panel for the **Loop** module on the DataViewer. Values must be typed into the parameter value boxes. These values should be floats so to get the correct number of decimal places it is best to **press the return key** after putting a value in each box:

- **Start Value 0.00**
- **End Value 54.00**
- **Increment 1.0**

To start the animation **click on the dimple Run**.

6. Try altering the axis along which the animation i done and then try exchanging the **orthoslice** with the **slice plane** and **slice** modules if you have time

Aside: The animation can be speeded up in a variety of ways to make it more like a cine animation. Details of these methods are in the white paper produced by AVS. Some of the simpler methods include:

- Stop the modules flashing. Select the **Option menu** form the **main menu bar** on the network editor. From this menu select **Module Flashing**.
- Disable the status bar. The status bar is the green bar on the bottom left hand side of a single window application. It has two buttons connected to it the bottom one stops the bar from show which module is currently firing.
- Switch off the caching. From the **Editors menu** on the Single Window Application select the **Object** menu click on the **toggle named Cached**.

Exercise 4. Threshold Slicing

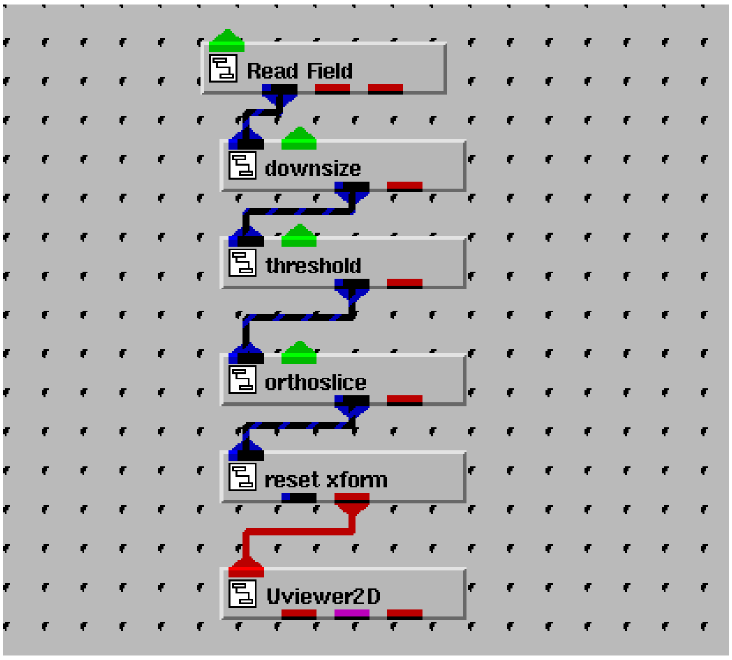
Aim

This allows the user to restrict the data values which are displayed in a slice of data to those within a certain range. This is often useful when trying to classify data into various substances. For instance, it may be required to restrict the data in a CT scan volume to values which represent bone.

Thresholding is one of the simplest forms of segmentation.

Threshold Slicing

Create or read in the network shown below (**/usr/networks/ex4/threshold.v**)



The modules used are:

- Read_Field (Data IO): reads an AVS field file and converts it into an AVS/Express field.
- downsize (Filters): resamples a field using a scaling factor. Usually the factor is greater than 1 and the size of the field is reduced to save processing time and memory.
- orthoslice (Mappers): used to view 3D uniform data. Shows all the 2D slices of data for each of the principal axes
- threshold (Filters): it “removes” all the values in a field above a max_value, and/or below a min_value by setting them to NULL.
- Bounds (Mappers): generates lines or surfaces that show the bound ing box of a 3D structured field.
- Uviewer3D (Viewers): produces a 3D image in the DataViewer.

This network combines the functionality of the **orthoslice** with that of the **threshold**. This allows you to isolate a range of data values in the slice displayed and then go through the rest of the slices to check that the thresholding is accurate for the entire dataset. This is particularly useful when trying to isolate features in the data.

1. Read in one of the 3D datasets.

2. We have put a **downsize** module in the network to reduce the resolution and speed up the system. Use this module to reduce the resolution at the beginning and as the threshold is more accurate increase the resolution of the data.

3. With the two sliders **min value** and **max value** on the **threshold** control panel try to select a grey scale range to isolate the bone from the rest of the image. You may need to alter the settings on the buttons **Below min value** and **Above min value** to do this. Now see if you can also get a range out for brain tissue.

Exercise 5. Probing The Data

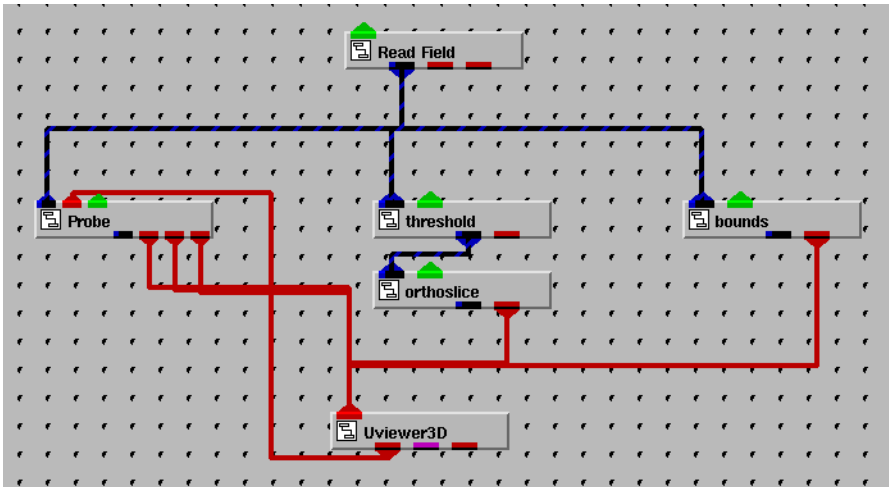
Aim

When viewing medical data, it is often necessary to relate the image generated back to the original data to answer questions such as ‘what data values cause this feature?’. Probing is a technique which can be used to report back the original data values Probes are commonly used in medical imaging but can be used as an image analysis tool.

Probe

Add the Probe module (N.B. there are two probe modules we want the starting with a capital P) to the network as shown below or read the network from **/usr/networks/ex5/probe.v**. The orthoslice and threshold modules will not act in a consistent way if they are put in the network the wrong way round. A more consistent result is obtain with the network given below.

The Probe module may produce an error message because initially it does not have a selected data point so dose not have an output, ignore this.



The extra modules are:

- Probe (Mappers): this displays a data value present at the probe object’s location in the mesh. When the probe is not exactly on a node the value is interpolated.

The **Probe** module allows individual points to be investigated. Use the **Probe** to look at the data slice that you have generated; values should appear on the view with the view. Make a note of the value range for bone. What are the problems with it?

2. To probe the data press the **<Ctrl>** button on the key board and then press **left mouse button** on the correct point in the data.

3. The size of the probe crosshairs is proportional to the data value. Press the **Probe Normalize** button and increase the value on the **probe scale** slider to get a consistently visible cross hair. Remember from the data-map that the dark blue colour on the image has a zero data value

4. Use the probe to confirm the ranges you determined in exercise 4.

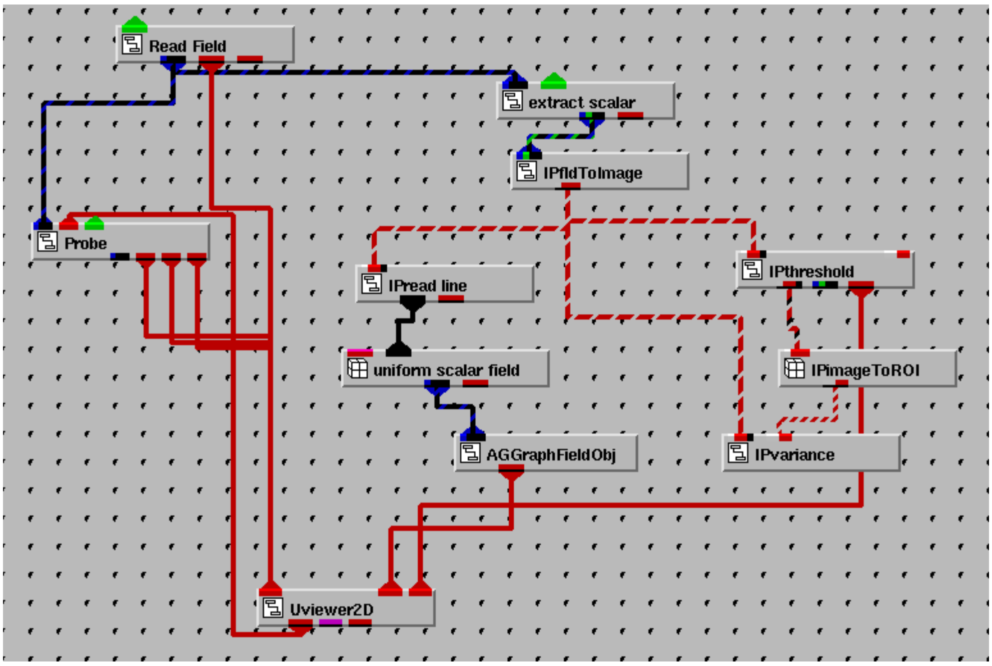
Exercise 6. Image Analysis

Aim

It is useful to analyse images so that details can be better understood. There are many methods of image analysis but a few are used here so that we can build up a classification table. The probe is used to find data values from images. Image variance is used to show the characteristics of a region of interest and image measure is used to give the length of a line and a histogram of values across that line. This application is designed to allow you to examine a 2D image by a variety of methods so that you can compare the results.

Image Analysis

1. Create the network shown below and read in the 2D file mri.fld. To build this network start by producing a **2D MultiWindowApp**, produce the extra input ports on the Uviewer2D by Selecting the **Editors** menu on the Viewer Control Panel, select the **View** option and select **Scene** form the top pull down menu, select a **2D New Camera Type** and then click on **Create**. Do this before instancing any modules in the network area. This network is also in **/usr/networks/ex6/im_process.v**



The modules used are:

- Read_Field (Data IO): reads an AVS field file and converts it into an AVS/Express field.
- Probe (Mappers): this displays a data value present at the probe object's location in the mesh. When the probe is not exactly on a node the value is interpolated.
- extract_scalar (Filters): this extract a scalar data element from a field's vector component.
- IPfldToImage (Accessories. Utility_Modules):
- IPthreshold (Accessories.Image_Processing): it "removes" all the values in an image above a max_value, and/or below a min_value by

setting them to NULL.

- IPimageToROI (Accessories. Utility_Modules): converts a ip_image to a Roi image.
- IPvariance (Accessories.Image_Processing): finds the mean, variance, and number of pixels in an ip_Image.
- IPreadline (Accessories.Image_Processing): reads image data along a specified line.
- uniform_scalar_field (Field_Mappers.Field_Mappers): generates a field of uniform scalars.
- AGGraphFieldObj (Accessories.Graphics.Graphing): produces the outline and axes for a graph.
- Uviewer2D (Viewers): produces a 2D image in the DataViewer.

2. The Probe module is used to let you select a point in the image a find out the corresponding data value. Try finding the values for blood vessels and bone.

3. Threshold lets you select a data value range and display the selected pixels as a binary image.

4. IPvariance takes the data values selected from threshold and displays the number of pixels, mean and variance on the module's control panel. If the variance value is low then threshold values are likely to be valid for one tissue.

5. The IPreadline module allows you two type in the location of two data points and produces an array of data values for the data points that lie on the line. These are used to produce a histogram of results. Use the Probe module to find the location of two data points on either side of a tissue of interest put these values in the IPreadline module and look at the histogram (With AVS/Express Developers Edition the location points of the line can be fed in from a module that draws a line on the data).

6. How do these results compare and can they be used to aid classification?

Exercise 7. Generating Isosurfaces

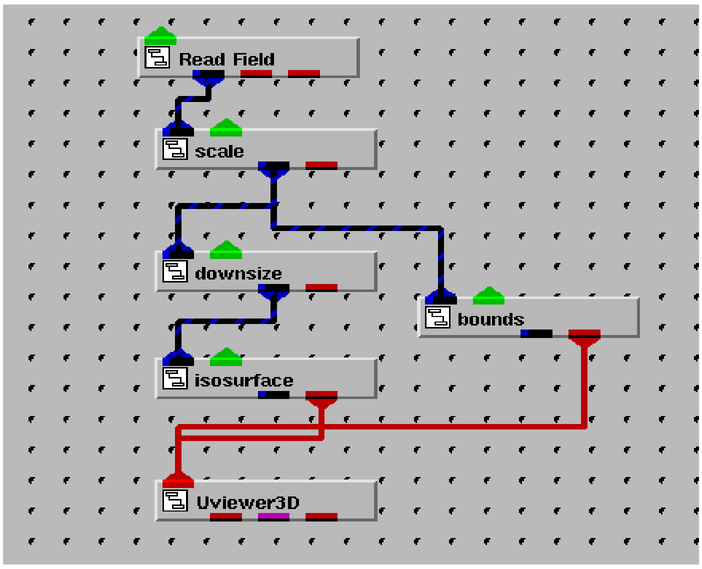
Aim

Volume data can be divided into a number of objects by segmentation/classification. Once objects are isolated they can be rendered as solid objects. Isosurface puts a surface around data points of the value so that the data is seen as a number of solid objects.

Isosurfacing is a widely used technique for the visualization of volume data. With Isosurfacing a geometric surface is generated for the data values which lie within a range specified by the user. AVS has a module for isosurfacing which uses the marching cubes algorithm.

Isosurface

Create the network shown below. Reduce the resolution of the image by altering the **downsize** values before reading the data in.You can also read the network from **/usr/networks/ex7/iso.v**



The modules used are:

- Read_Field (Data IO): reads an AVS field file and converts it into an AVS/Express field.
- downsize (Filters): resamples a field using a scaling factor. Usually the factor is greater than 1 and the size of the field is reduced to save processing time and memory.
- isosurface (Mappers): creates a contour surface that passes through all cells containing a specified level value. The surface may occur between node locations and so the contour is interpolated for these values.
- scale (Filters): scale 3D mesh in X, Y and/or Z.
- Bounds (Mappers): generates lines or surfaces that show the bound ing box of a 3D structured field.
- Uviewer3D (Viewers): produces a 3D image in the DataViewer.

2. Read in the **smlhead.fld** data set. Producing the appropriate isosurface may take some time so disable execution if you wish to alter several controls. This is the top button to the right of the status bar at the bottom of the View Control Panel.

3. The **isosurface** module will create a surface by generating a triangle mesh between points of similar values. The value of the chosen **isosurface** value can be altered using the level parameter in the **isosurface** control panel

4. Create an **isosurface** using the value for bone or blood vessel values which you determined earlier.

5. The scale module lets you alter the interslice distance in the Z axis so the data looks more realistic.

6. Change the rendering style on the isosurface from the **Editors** menu**.** Try these firstly with low resolution or the results will be very slow**.** From the Editor menu select **Object** and then alter the top button labelled **Object** to **Modes** from the popup menu. Alter the values in the **Surface Render** to see the effects of different shading styles.

7. Try altering more properties of the surface rendering by changing the **Object** popup menu to **Properties** and then changing the **Type** popup menu to **Surface**. These will alter many of the surface properties making it glossy or transparent.

What are the problems with viewing medical data using isosurfaces generated using these methods?

Exercise 8. Excavating The Data

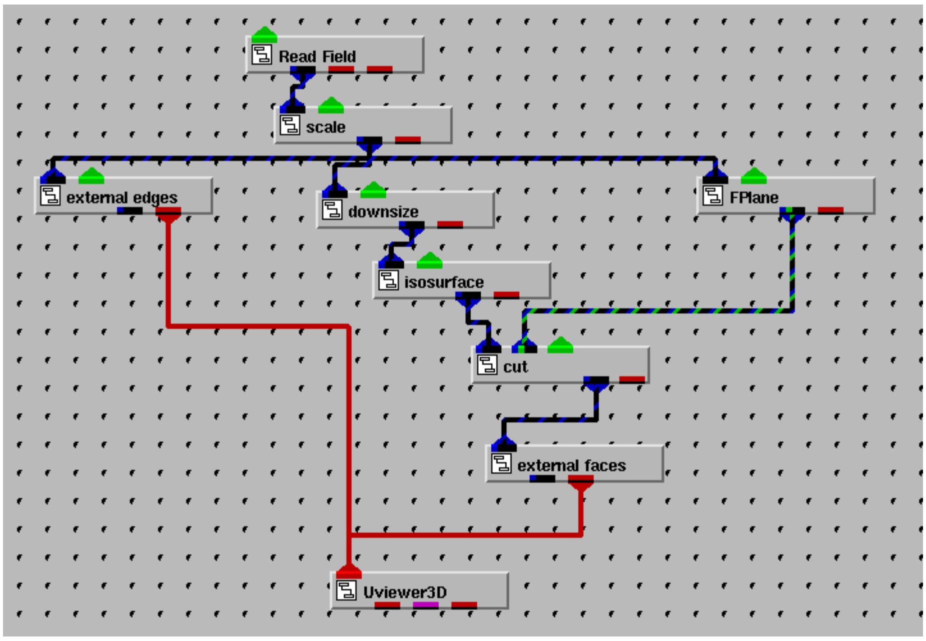
Aim

Once a volume of data has been rendered it may be useful to be able to look at the interior structure of the volume. One method of doing this is to use cut planes to excavate the volume. Cut planes allow the user to zero out portions of the outer structure of the data so that the interior can be viewed.

Once you have a solid model (an isosurface) you can only see the outside surface of it unless you cut into it.

Excavate

Build the excavate network using the **cut** module so that it looks like this.You can also read this network from **/usr/networks/ex8/excavate.v**.



The modules used are:

- Read_Field (Data IO): reads an AVS field file and converts it into an AVS/Express field.
- scale (Filters): scale 3D mesh in X, Y and/or Z.
- downsize (Filters): resamples a field using a scaling factor. Usually the factor is greater than 1 and the size of the field is reduced to save processing time and memory.
- isosurface (Mappers): creates a contour surface that passes through all cells containing a specified level value. The surface may occur between node locations and so the contour is interpolated for these values.
- cut (Mappers): cuts a portion of a field on one side of the cutting plane.

- FPlane (Geometries): create a plane mesh scaled to an input field.
- external faces (Mappers): extract faces of a field for faster rendering.
- external edges (Mappers): extracts the edges of a field to reveal inside objects.
- Uviewer3D (Viewers): produces a 3D image in the DataViewer.

Read in the **smlhead.**fld file. Do not forget to disable the execution if there are several parameters you need to change and it is slow.

2. Use the plane distance slider on the **cut** control panel and the X, Y and Z rotation sliders on the Transformation Editor on the **FPlane** control panel to carve away portions of the data volume. These controls let you cut away blocks of your data and reveals what is inside it.

3. Use the **isosurface** controls to put a surface around the bone and brain data ranges you obtained earlier and try cutting through the surface.

4. It is not always possible to recognize the features being displayed so try to add an **orthoslice** or another **isourface** module to the network.

Exercise 8. Volume Rendering

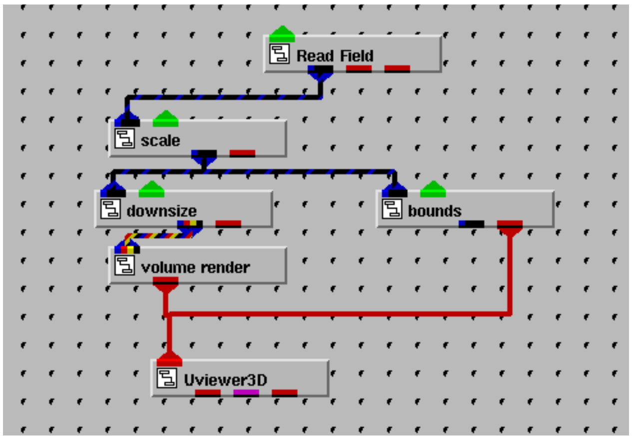
Aim

Although the Isosurfacing techniques and use of cut planes do allow some investigation of the interior of volumes they are limited in a number of ways. Firstly the methods used for isosurfacing are dependent on a binary classification where data is either inside or outside a surface. For some data sets this can cause important features to be missed. Secondly the use of cut planes can cause the loss of context information where the relative position of various features is required. A more accurate, though computationally much more expensive method for viewing volumes is the use of direct volume ray-casting. This technique allows small features to be represented and transparency to be incorporated into the images, so that interior features can be viewed in the context of their surroundings.

Volume rendering does not put a surface around features in the data it is a projection through the data and by altering the colour and transparency mapping internal details can be seen in the volume. MIPS (maximum intensity projection) is the type of volume rendering used most with medical data, the result is a grey scale view which looks like an X-ray image.

Volume Render

1. Create the network or read the network shown below.You can also read this network from **/usr/networks/ex9/vol_rend.v**



The modules used are:

- Read_Field (Data IO): reads an AVS field file and converts it into an AVS/Express field.
- downsize (Filters): resamples a filed using a scaling factor. Usually the factor is greater than 1 and the size of the field is reduced to save processing time and memory.
- volume render (Mappers): this directly renders a uniform, 3D, 3-space, scalar, byte volume using BTF or Ray-tracer rendering modes. Both must be operated with the software render. It provides mixed rendering of volumes and geometries.
- Bounds (Mappers): generates lines or surfaces that show the bound ing box of a 3D structured field.
- Uviewer3D (Viewers): produces a 3D image in the DataViewer.

2. This is a soft ware render and the default must be set to this option. In the **DataViewer** select the **Editors** menu and from this select the **View** option. On the View control panel select the popup **Render** menu and select **software**.

3.Read in the `head.fld` file

4.Keep the resolution from the downsize module as high as possible, it is easier to see the effects of the volume render on a high resolution data but low resolution will make the rendering quicker.

5. There are many parameters that can be altered in the **volume render** control panel.Try altering some of these to see their effect, particularly try:

- **Surface**
- **Interpolation**
- **Ray Algorithm**
- **Distance Normalize**
- **Fat Ray**

Try **editing the alpha ranges** at the bottom of the control panel to see how you can cut away volumes with a known range of data values. Set the alpha ranges so that they correspond to the classification ranges that you determined for bone and soft tissue.

- There are only two possible data ranges, you can switch between editing the ranges by altering the **Current Range slider**.
- The **Range Position slider** alters the midpoint of the ranges to be altered.
- The **Minimum Alpha** and **Maximum Alpha** sliders allow you to alter the minimum and maximum alpha values in the current data range.
- The **Alpha Range Model** menu controls the way the alpha values are calculated across the alpha range.

6. Try adding the cut module or orthoslice module to the network to cut away portions of the dataset to reveal the inside detail.

7. How do these techniques compare with isosurfacing?