IDL Tutorial

Volume Rendering

The IDL Intelligent Tools (iTools)

The IDL Intelligent Tools (**iTools**) are a set of interactive utilities that combine data analysis and visualization with the ability to produce presentation quality graphics. The iTools allow users to continue to benefit from the control of a programming language, while enjoying the convenience of a point-and-click environment. There are 7 primary iTool utilities built into the IDL software package. Each of these seven tools is designed around a specific data or visualization type :

- Two and three dimensional plots (line, scatter, polar, and histogram style)
- Surface representations
- Contours
- Image displays
- Mapping
- Two dimensional vector flow fields
- Volume visualizations

The iTools system is built upon an **object-oriented** component framework architecture that is actually comprised of only a single tool, which adapts to handle the data that the user passes to it. The pre-built *iPlot*, *iSurface*, *iContour*, *iMap*, *iImage*, *iVector* and *iVolume* procedures are simply shortcut configurations that facilitate ad hoc data analysis and visualization. Each pre-built tool encapsulates the functionality (data operations, display manipulations, visualization types, etc.) required to handle its specific data type. However, users are not constrained to work with a single data or visualization type within any given tool. Instead, using the iTools system a user can combine multiple dataset visualization types into a single tool creating a hybrid that can provide complex, composite visualizations.

Displaying 3-Dimensional Volumetric Data

A volumetric dataset consists of a 3-Dimensional array of numbers that represent a certain measurement made at each element location within the 3-D space. Each data element location within the 3-D array is called a **voxel**, which is analogous to a three-dimensional pixel. One manner in which volumetric datasets can be created is by stacking numerous 2-D image slices together into a three-dimensional array.

In the following exercise, volumetric data of numerous image slices from the Visible Human project stacked together will be visualized using the *iVolume* utility. This volume consists of seventy 128 x 128 image slices acquired through the torso of a human body. This example data is stored in a file named "*torso.dat*" that is located in the "*data*" subfolder.

The "*torso.dat*" file does not have any particular file format, and the data is stored within this file in a **flat binary** fashion. Consequently, the user will be required to provide IDL with the information it needs in order to successfully read the data from the file on disk. Use the following steps to load this dataset into the *iVolume* utility :

1. IDL> iVolume
2. Select "*File > Open…*" from the *IDL iVolume* window.
3. Within the *Open* dialog, change the "Files of type:" droplist to "*All files (*)*".
4. Select the "*torso.dat*" file and hit "*Open*".

The dialog for the *Binary Template* wizard will appear. This wizard helps walk the user through the steps of providing IDL with information on the structure of the binary file so the data can be read into the *iVolume* utility. The user must provide the following basic information in order for the data to be successfully input :

- Number of dimensions
- Size of each dimension
- Data type
- Offset (size of header)
- Byte order (Little Endian or Big Endian)

Without this information, there is no way for the software to know how to read-in the binary data from the file on disk.

5. Start by pressing the "*New Field…*" button within the *Binary Template* wizard.

A field is basically a data segment that will be read-in from the file on the harddrive. In this case, the "*torso.dat*" file only contains the stacked image data, so the entire volume will be input from the file as one field.

6. Within the *New Field* dialog, set the "Field name:" parameter to the string "*torso*" [Fig. 1].
7. The data type for this volumetric dataset is unsigned 8-bits (Byte), so the "Type:" droplist can be left as its default setting [Fig. 1].
8. The selected data file does not contain a header, so the "Offset:" parameter can be left as its default "*>0*" setting [Fig. 1].
9. The volumetric dataset has three dimensions, so set the "Number of dimensions:" droplist to "*3*" [Fig. 1]. Once this is accomplished, the "Size:" text boxes for the 1st, 2nd, and 3rd dimensions will become active.
10. The seventy images are stored within this volume using an interleave known as *band-interleaved by line* (BIL). Consequently, the dimensions of the 3-D array are 128 x 70 x 128 [Fig. 1].
11. Once these settings have been completed, press "*OK*" to dismiss the *New Field* dialog.

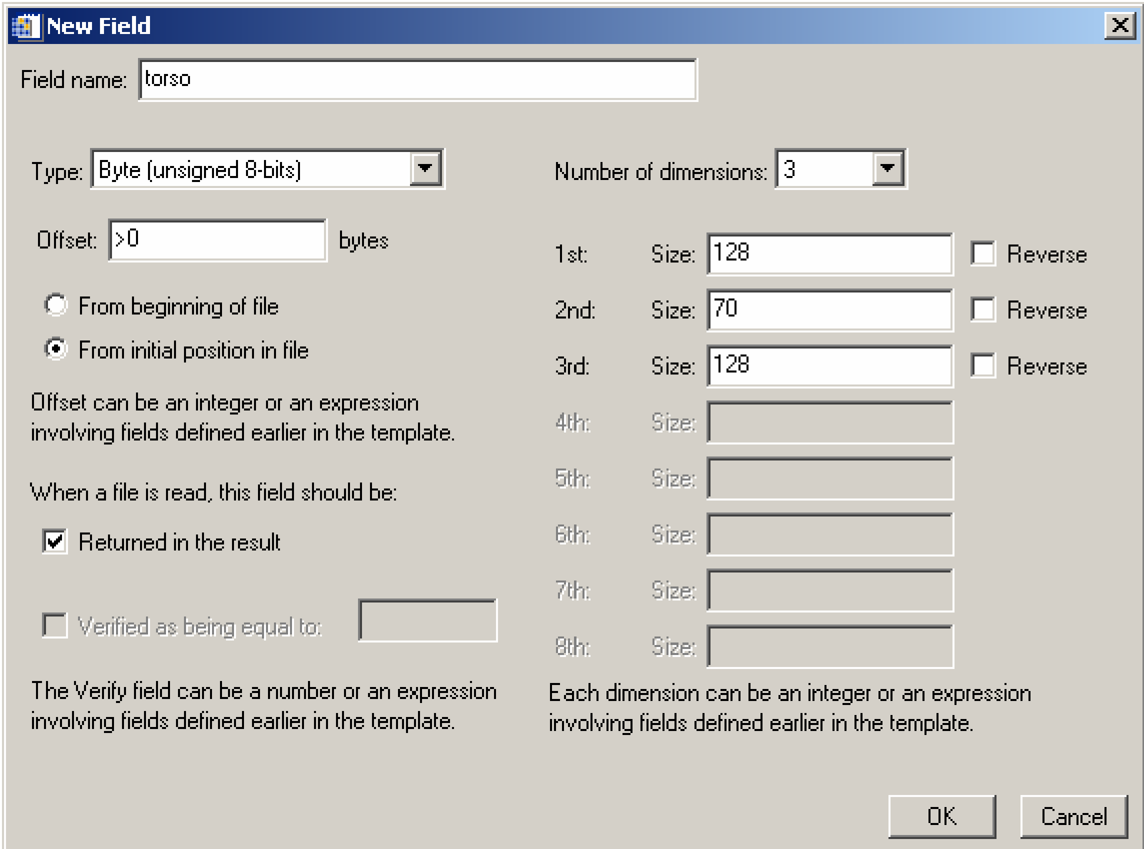

Figure 1: The New Field dialog within the Binary Template wizard

The user will be returned to the *Binary Template* wizard, which will show the parameters for the "*torso*" field definition that was just created [Fig. 2].

12. Since the data type for the selected file is only single bytes, the user does not need to be concerned with the byte order, and the "File's byte ordering:" droplist can be left as its default setting of "*Native*" [Fig. 2].
13. Press "*OK*" to complete the *Binary Template* wizard.

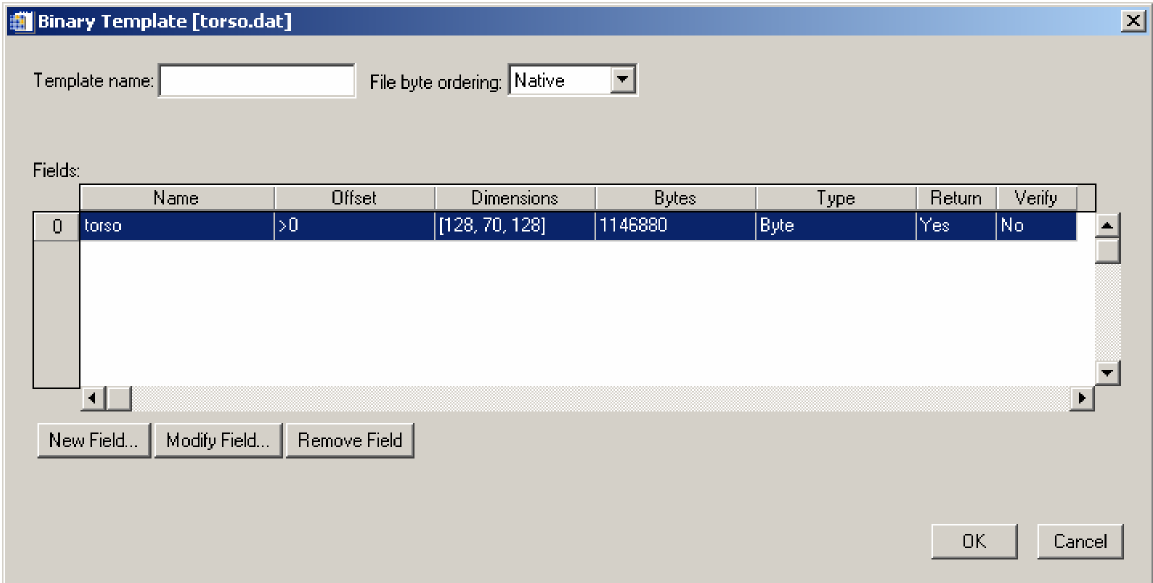

Figure 2: The completed Binary Template wizard

Once the *Binary Template* wizard is finished a 3-Dimensional volume object will be automatically inserted into the *IDL iVolume* utility. By default, the data space and axes will be shown, but the volume itself will not be rendered.

There are a number of ways to visualize volumetric data within the *iVolume* utility. Of course, the individual **image planes** that were stacked up to create the volume can be extracted and viewed in each of the three (X-Y, Y-Z, and X-Z) orthogonal planes. A technique known as **isosurfacing** can be used to contour the data, which creates a surface that shows the voxels in the 3-D data space that all have a specified data value (the isovalue). In addition, an **interval volume** can be extracted which is a tetrahedral mesh that spans the 3-D data space between two isosurfaces created at different data values. Finally, **true volume rendering** can be performed, which maps the voxel data elements to colors and opacity values through a set of lookup tables, then projects the 3-D volume to a two-dimensional graphical visualization. By default, the volume object within the iVolume utility is setup to perform volume rendering.

14. While the data space is selected, click on the "*Render*" button within the *Volume* panel on the right-hand side of the *iVolume* utility.

A volume rendering of the 3-Dimensional array is created using the default color palette (grayscale) and opacity settings. The reason the *iVolume* utility forces the user to press the "*Render*" button in order to visualize the object is because the volume rendering process of very large datasets can be very computationally intensive. In most cases, it is not feasible to constantly render the volume object when any of the standard iTools manipulators (Translate, Rotate, Zoom, etc.) are used to modify the visualization, especially when performing a click-and-drag operation.

15. Using the Select/Translate arrow button, click on the data space and move it slightly in one direction within the viewplane. The volume rendering of the object disappears.

Since the current volumetric dataset is relatively small, the Auto-Render option can be turned on so the iVolume utility will automatically update the visualization after every action.

16. Click on the small white checkbox to the left of the Auto-Render option to enable the auto-rendering capability.

Now the volume will be constantly rendered while object manipulations are being performed.

Data Space Scaling

Currently the data space is displayed with its default geometry, which scales the volume object in each of the three dimensions so it fits within a perfect cube. However, the size of this volume is actually 128 x 70 x 128, so it is not equal in all three dimensions. Consequently, the volume object is being exaggerated in the Y dimension (70) to fit the same 3-Dimensional space as the other X and Z dimensions (128). In order to display the volume with its true geometry, the scaling of the data space must be set to **isotropic**.

1. From the IDL iVolume menu system, select "Window > Visualization Browser…".
2. On the left-hand side of the Visualization Browser window, select the "Data Space" object [Fig. 3].
3. In the Data Space property sheet, click on the box to the right of the "Isotropic scaling" property and set the droplist to "Isotropic" [Fig. 3]. The volume object within the iVolume utility will automatically update and display the data space in an isotropic fashion.
4. Close the Visualization Browser window.

The data space is now displayed in an isotropic fashion and the length of each of the three X-Y-Z axes is proportionate to their size.

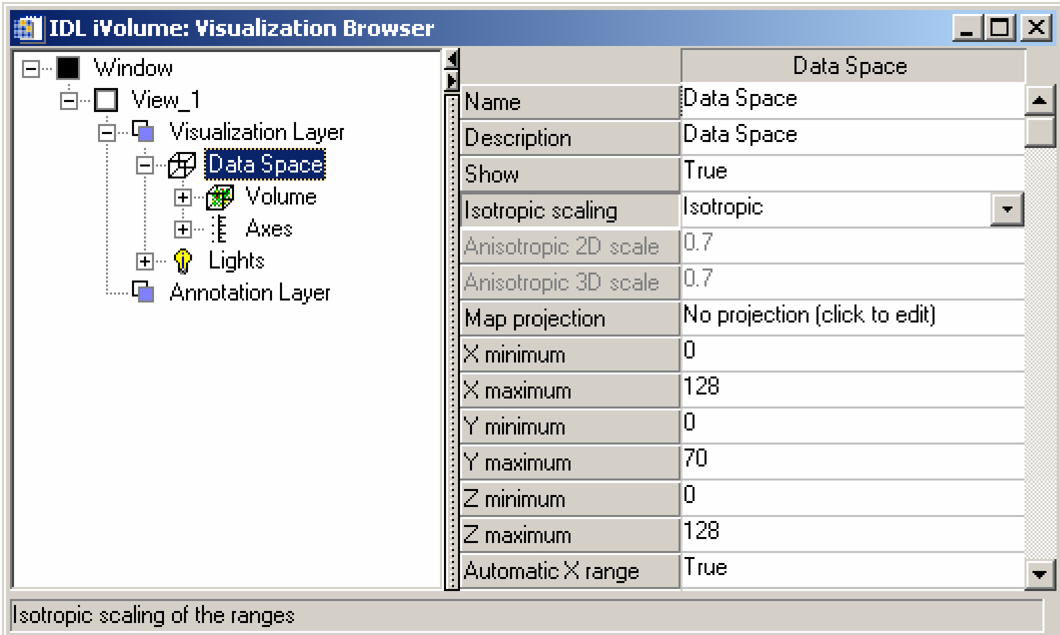

Figure 3: Changing the scaling of the data space object to isotropic

Volume Object Properties

Volume objects have numerous properties that control the manner in which they are rendered. These properties define the rendering attributes for the visualization. The first properties of the volume rendering that can be easily modified by the user are present in the **Quality** and **Boundary** droplists on the *Volume* panel along the right-hand side of the *iVolume* utility. By default, the *iVolume* utility displays the volume with a low rendering quality and solid walls boundary.

There are two rendering qualities available for the volume object visualization :

- **Low (textures)**: Rendering is performed with a stack of 2-D texture-mapped semi-transparent polygons. The polygons are oriented so that the flat sides face the viewer as directly as possible. On most computers, low quality mode renders the volume faster but with less accuracy.
- **High (volume)**: Rendering is performed with a ray-casting volume renderer. This quality is more CPU-intensive and will usually take much longer to complete than low quality mode, but creates a much more accurate visualization.

The Boundary option displays an internal translucent solid cube within the data space, which is useful when the volume is not automatically rendered so that the user can locate and select the volume when no graphic is visible. Since the Auto-Render option is currently enabled, there is no need to display the boundary.

1. Change the Quality droplist to "*High (volume)*" and the Boundary droplist to "*None*". Notice the effect these changes have on the volume rendering visualization [Fig. 4].

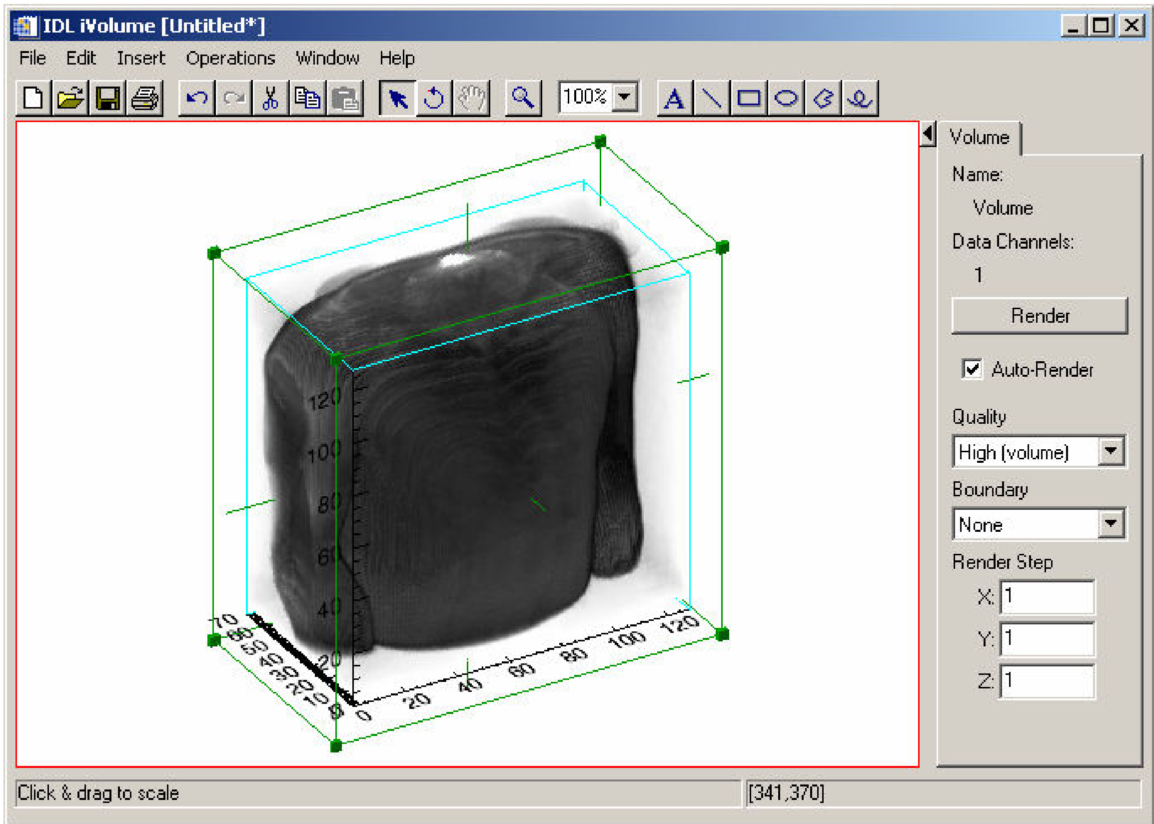

Figure 4: Volume rendering with high quality and no internal extents

Once the rendering quality is set to high, the Render Step "X:", "Y:", and "Z:" fields will become sensitive, which allow the user to specify the stepping factor (in screen dimensions) through the voxel matrix [Fig. 4]. By default, these fields are set to "*1*" so that each and every voxel is considered when the volume is rendered. Changing this setting to "*2*" would render only half as many voxels in the specified screen dimension.

It is much quicker and easier to modify the other properties of the volume visualization when the Auto-Render functionality is turned off. Once the desired changes have been made, the "*Render*" button can be pressed in order to display the visualization of the volume dataset.

2. Un-check the Auto-Render checkbox.

The **opacity** lookup table controls the transparency of any given voxel, while the current **color palette** defines what color is applied based on the voxel's data value. Manipulation of the color palette and opacity table is critical to controlling and improving the appearance of a volume rendering.

3. While the volume object is selected, right-click on it and select "*Properties…*".
4. Within the *Visualization Browser* window, click on the "Edit color/opacity table" field to the right of the "Color & opacity table 0" property and select "*Edit…*". This will bring-up the *Palette Editor* dialog.

5. Within the "*Load Predefined…*" droplist, scroll down and select the "*Rainbow + white*" color table.

Next, the opacity table can be modified in order to control the transparency effect that is applied during the volume rendering. The *Palette Editor* dialog contains a number of items [Fig. 5] :

- **Reference colorbar** along the top illustrating data value range increasing from left-to-right using a grayscale ramp.
- **Current palette colorbar** that displays the currently loaded color table being applied to the visualization.
- **Channel display window**, which contains line plots for the individual red, green, and blue color palette channel vectors, along with a purple line for the Alpha channel (opacity lookup table).
- **Cursor location/value window**.
- **View and edit tools**, including zoom options, color space, editing operations, and channel selection.

The user can use the mouse cursor within the channel display window to modify the selected channels by interactively drawing the desired lookup table curve. The 3 color channel vector line plots are displayed with increasing brightness from bottom-to-top, and the combination of these 3 vectors creates the "*Rainbow + white*" color table that was just loaded. Since the R, G, B color channels have already been set to the desired color palette in the previous step, there is no need to modify or visualize these channels.

6. At the bottom of the *Palette Editor* window, un-check each of the R, G, B color channels in both the "Display" and "Modify" rows [Fig. 5].

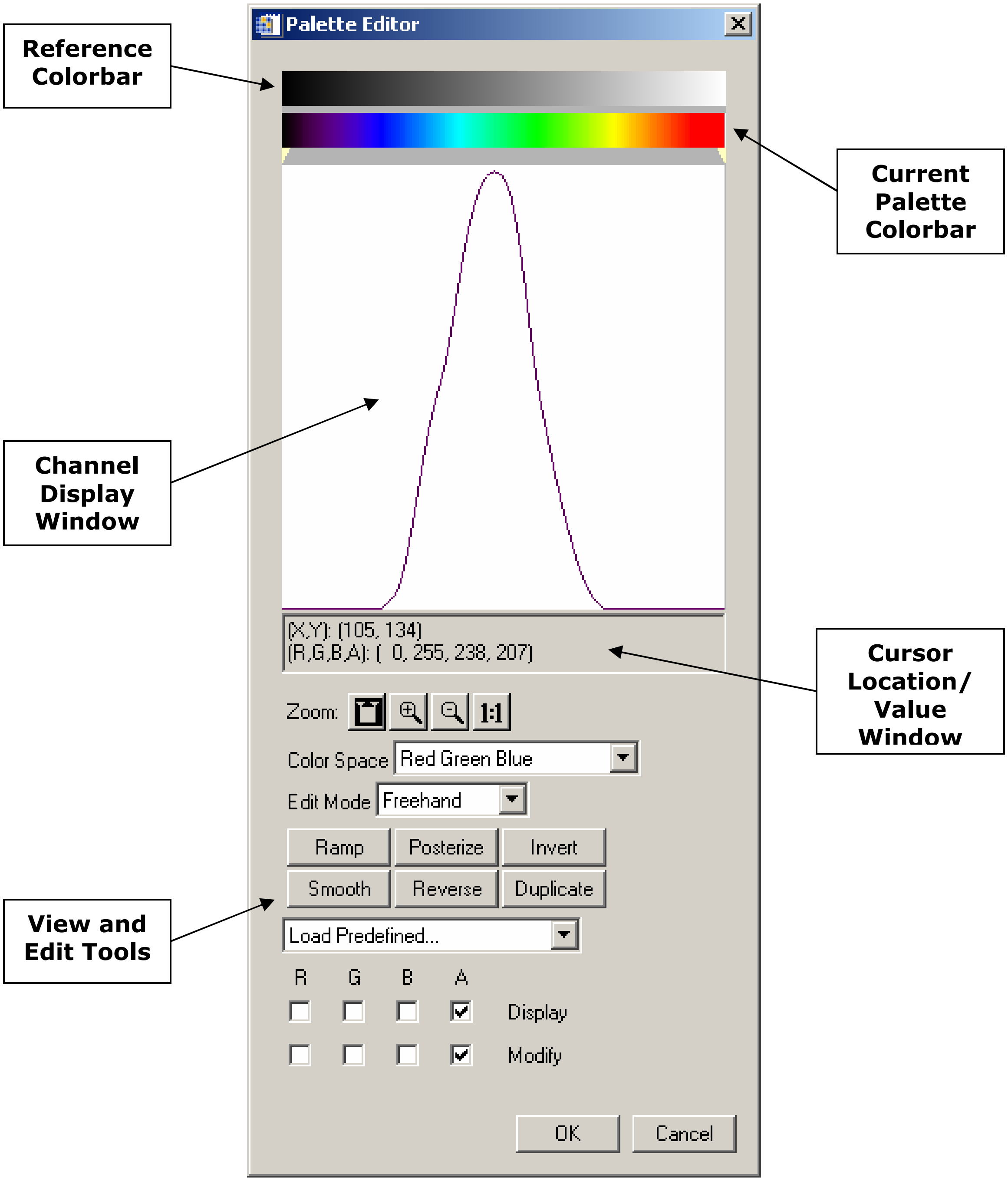

Figure 5: The Palette Editor dialog with a gaussian distribution curve for the alpha (opacity) channel

Now that all of the boxes under the R, G, B color channels are un-checked, only the **Alpha** channel (A) is displayed with the purple line plot. This is a plot of the lookup table for the current opacity effect, ranging from 0% opaque (100% transparent) at the bottom of the channel display window to 100% opaque (0% transparent) at the top. By default, the current opacity lookup table is a straight ramp from minimum data value (left-hand side) to maximum data value (right-hand side). This straight line has a slope of 1, meaning that the volume is rendered in a progressively more opaque fashion as the voxel data values get larger. Since the "Modify" field under the A channel is still checked, the user can click with the mouse cursor within the channel display window and draw a new opacity lookup table line in an arbitrary fashion.

7. Using the mouse cursor, click-and-drag within the channel display window and draw a standard gaussian shaped distribution curve [Fig. 5].
8. Click on the "*Smooth*" button a few times to give the line a smoother appearance if necessary.
9. Once the desired curve has been drawn, click "*OK*" to dismiss the *Palette Editor* dialog.

Creating a gaussian shaped curve for the opacity lookup table has the effect of highlighting (i.e. making more opaque) those voxels within the volume that have a medium data value around the middle of the overall data range.

10. Press the "*Render*" button in order to display the volume visualization.

The resulting visualization should look similar to Fig. 6.

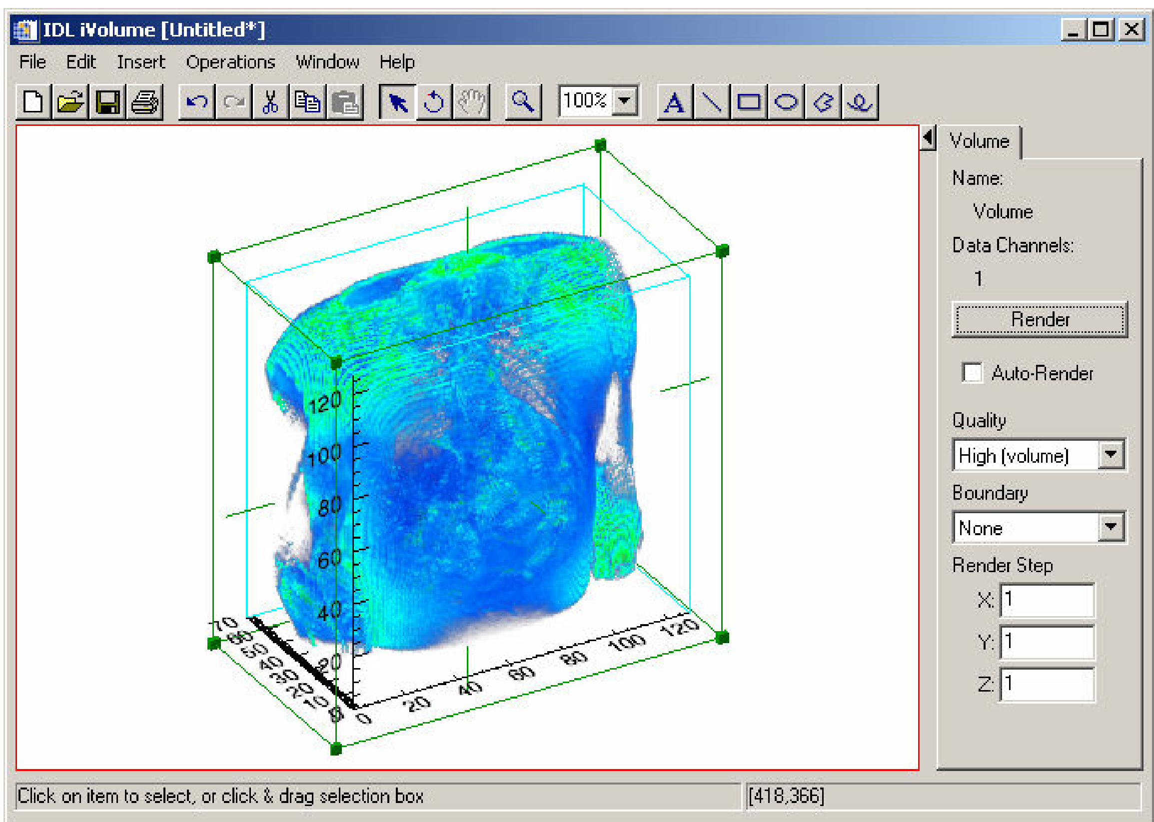

Figure 6: The volume rendering using a gaussian curve for the opacity

There are a number of other properties for the volume object that can be modified in order to control the appearance and quality of the visualization. By default, when rendering a volume object to the screen the voxel values that are selected to compute the visualization properties are selected using nearest neighbor selection process. If higher quality rendering is desired, the "**Interpolation**" property can be changed so this task if performed using a trilinear interpolation.

11. If the *Visualization Browser* window is not still visible, select "*Window > Visualization Browser…*". Within the *Visualization Browser* window, scroll down and change the "Interpolation" property for the Volume object to "*Trilinear*".
12. Press the "*Render*" button in order to display the volume visualization.

The rendering of the volume object should visually improve. When a volume is rendered, gradients within the volume are used to approximate a surface normal for each voxel, and the lighting sources in the current visualization are then applied to illuminate the object. **Gradient shading** can be enabled by modifying the "Use lighting" property. Furthermore, both sides of the voxels can be lighted by changing the "Voxel gradient" property.

13. Change the "Use lighting" property to "*True*" and the "Voxel gradient" property to "*Two-sided*".
14. Press the "*Render*" button in order to display the volume visualization.

The resulting visualization should look similar to Fig. 7.

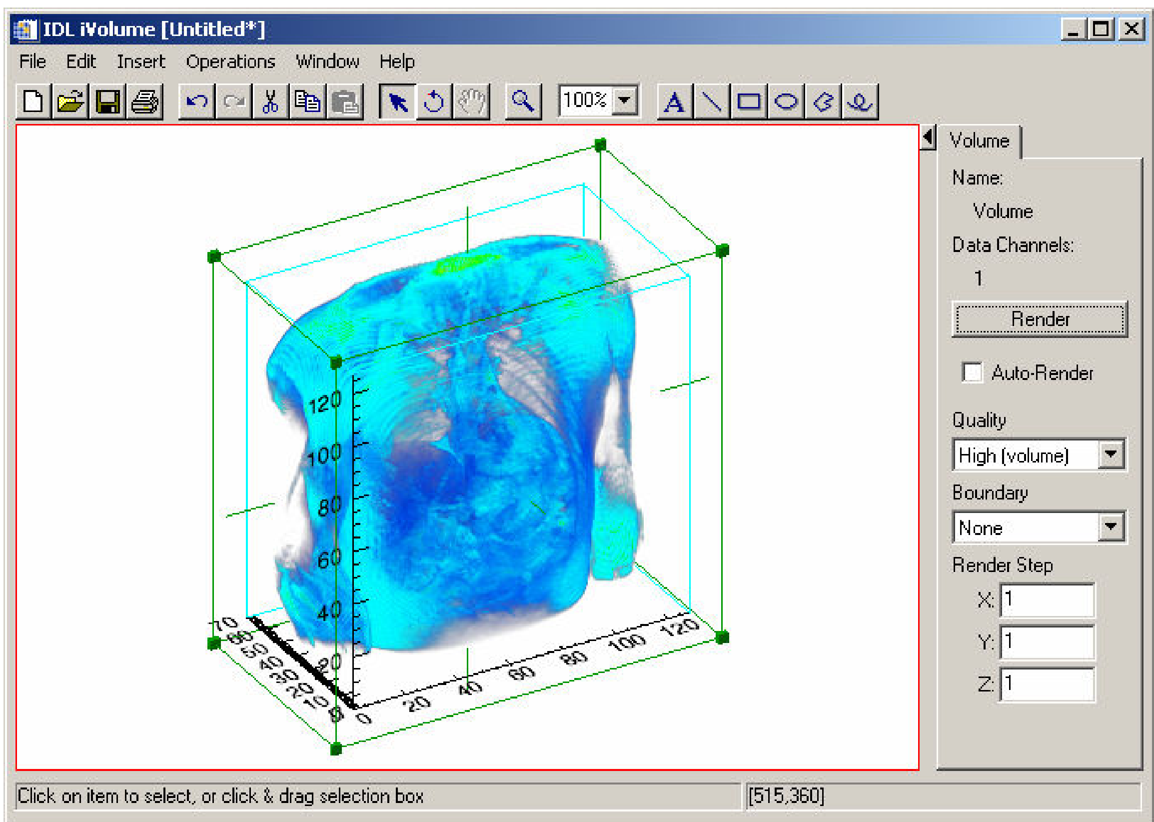

Figure 7: The rendering of the volume object with modified properties

By default, the volume object is rendered using the color palette and opacity table in a composite function technique known as **Alpha blending**. In Alpha blending, each voxel occludes other voxels behind it according to the opacity of the voxel in front, thereby allowing the viewer to see features within the 3-D volume. However, the *iVolume* utility also supports a number of methods for blending the projected voxels together to form an image. One common technique for projecting a volume into an image is the **Maximum Intensity Projection** (MIP).

15. Within the *Visualization Browser* window, change the volume object's "Composite function" property to "*Maximum intensity projection*".
16. Press the "*Render*" button in order to display the volume visualization.

The volume object rendering will change to display a MIP of the volume based on the current orientation of the 3-D data space. The resulting visualization should look similar to Fig. 8.

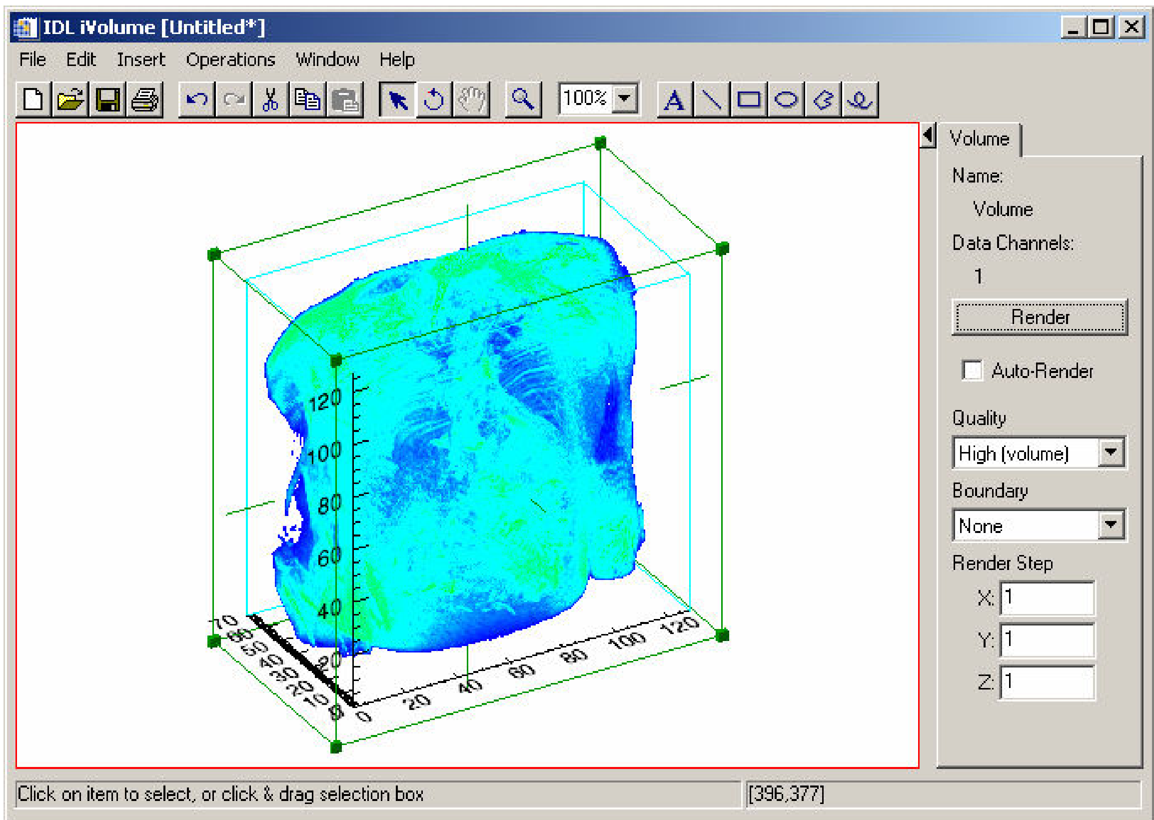

Figure 8: Maximum Intensity Projection (MIP) of the volume object

17. Once finished viewing the MIP visualization, revert the setting for the "Composite function" property back to the default "*Alpha blending*" and close the *Visualization Browser* window.

Image Planes, Subvolumes, and Isosurfaces

In addition to the true volume rendering capabilities, the *iVolume* utility also offers a wide variety of other tools that allow the user to visualize volumetric datasets using additional graphical objects. For example, the user has the ability to extract and analyze individual image slices from any of the 3 orthogonal planes.

1. While the volume object is selected, use "*Operations > Volume > Image Plane*" to insert a new image object into the existing data space.

A cyan color rectangular box outlining the default image plane oriented in the Y-Z direction and located in the exact center of the volume is inserted. The user can click on this cyan box to move the location of the image plane within the current orientation, and the center voxel location for the image plane is displayed in the lower-right hand corner of the *IDL iVolume* window. For more advanced modifications to the image plane, the *Visualization Browser* must be utilized.

2. Select "*Window > Visualization Browser…*" from the menu system.
3. Click on the new "*Image Plane*" object so its property sheet is displayed [Fig. 9].
4. Change the "Opacity control" property to "*Opaque*" so the image is readily visible [Fig. 9].
5. Change the "Orientation" property to "*Y*" so the image plane is oriented in the X-Z direction [Fig. 9].

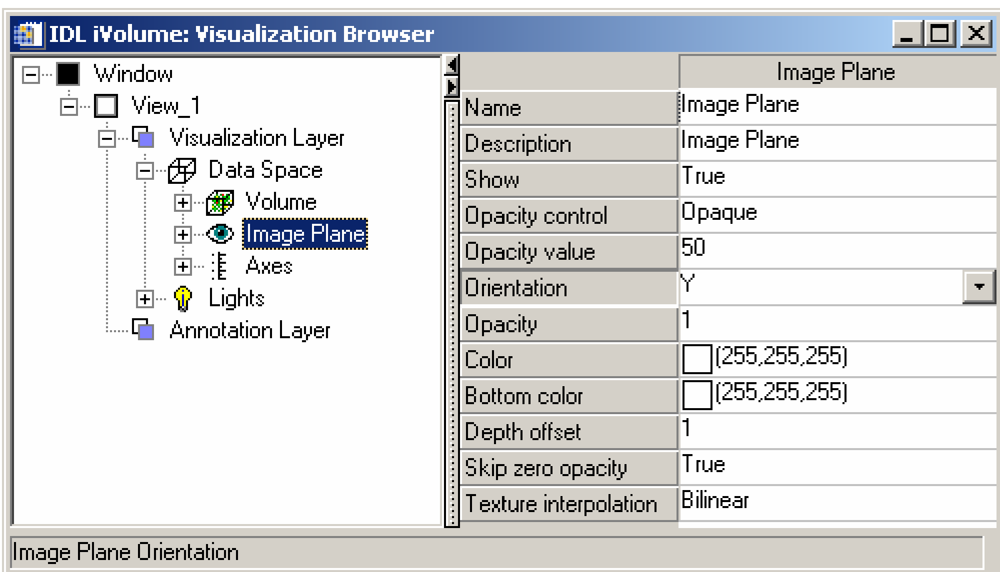

Figure 9: Modifying the image plane properties

The *iVolume* utility also has the ability to display a **subvolume** within the current data space. This allows the user to specify a subset of the 3-Dimensional array to use when rendering the volume object.

6. Within the *Visualization Browser* window, select the "*Volume*" object.
7. Click on the "Edit Subvolume extents" item next to the "Subvolume" property and select "*Edit…*".
8. Within the *SubVolume Extents Selector* dialog box, use the keyboard to enter a specific voxel range for a subvolume to display [Fig. 10]. This will extract a 32 x 32 x 32 voxel subvolume located in the center of the volumetric dataset.

- Volume X Extents : 48 → 79
- Volume Y Extents : 20 → 51
- Volume Z Extents : 48 → 79

9. Click "*OK*" to dismiss the *SubVolume Extents Selector* dialog and close the *Visualization Browser* window.
10. Press the "*Render*" button in order to display the volume visualization.

The subvolume extent will be rendered instead of the full 3-Dimensional dataset.

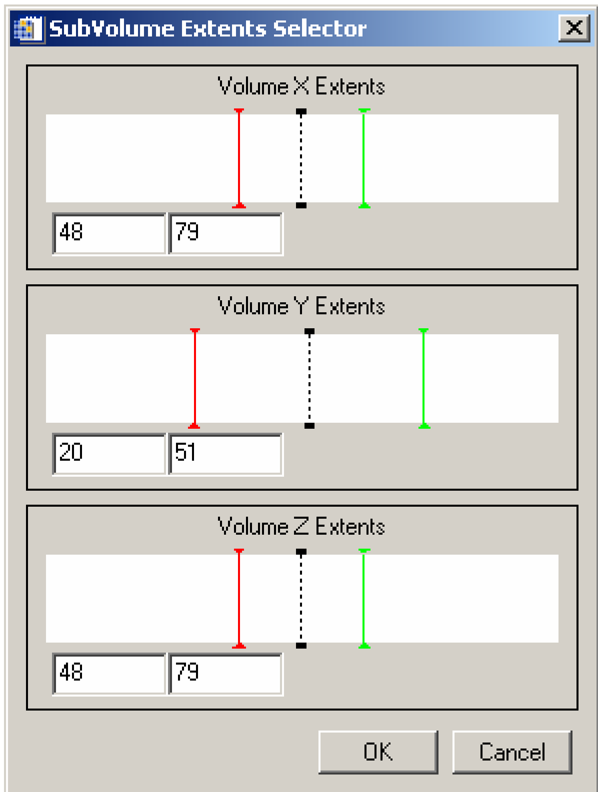

Figure 10: Specifying a subvolume to render

Finally, the volumetric dataset can be contoured in order to highlight all of the voxels that have the same specific data value. These voxels are highlighted with a planar object called an **isosurface** that is automatically loaded into the existing data space.

11. While the subvolume object is selected, use the "*Operations > Volume > Isosurface*" menu item to launch the *Isosurface Value Selector* tool [Fig. 11].
12. Use the mouse to drag the red data value selector line until the number "*95*" appears in the text box.
13. Press the "*OK*" button to dismiss the *Isosurface Value Selector* tool.

The isosurface is computed for the volume and a new graphical object is inserted into the *iVolume* utility. The resulting visualization should look similar to Fig. 12.

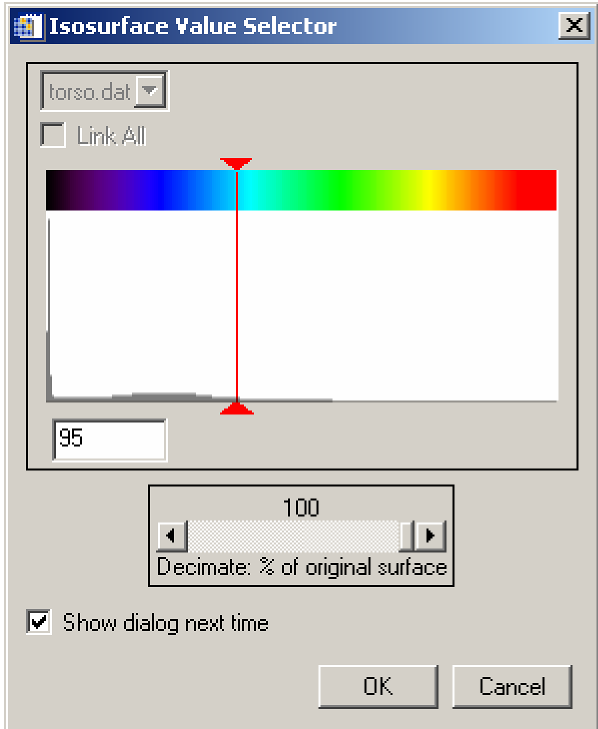

Figure 11: Selecting an isosurface data value

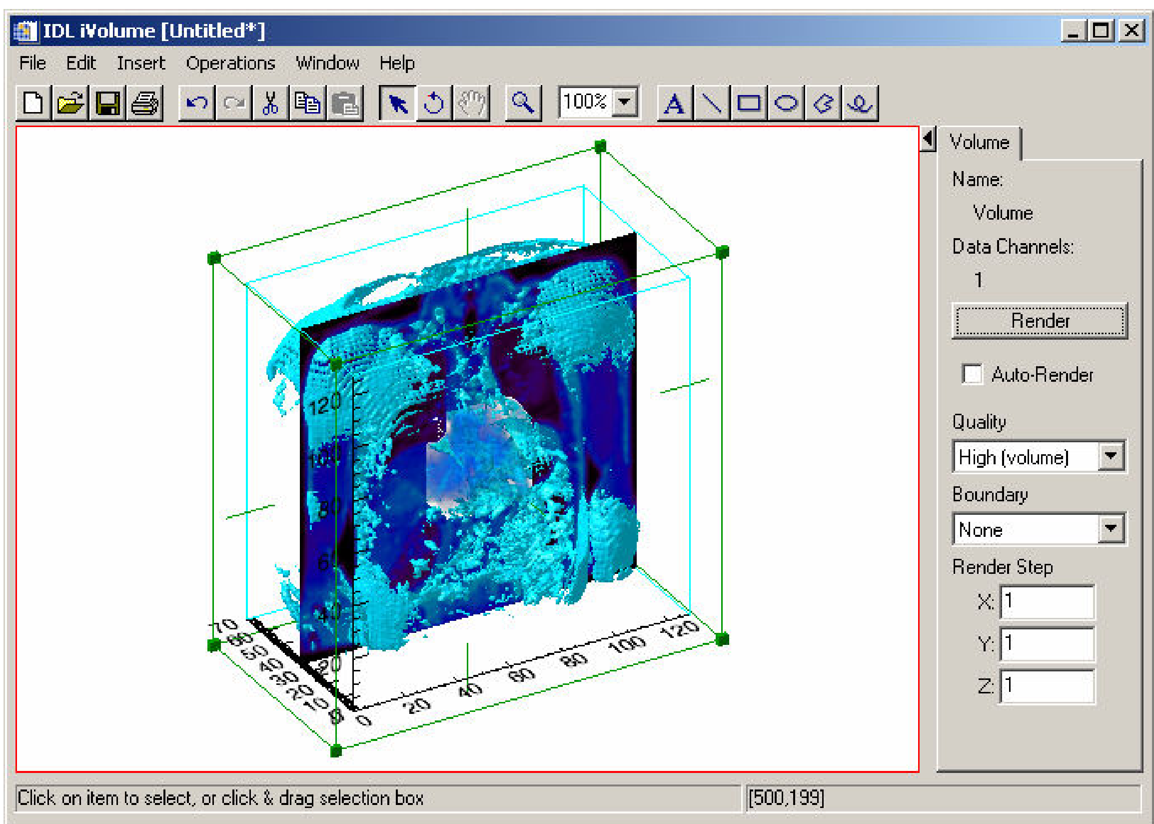

Figure 12: Composite visualization of the volumetric dataset

14. Once finished viewing the volume visualization, close the *IDL iVolume* window.