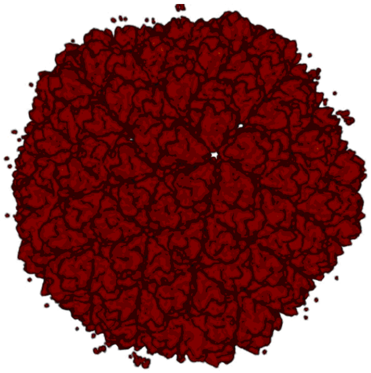
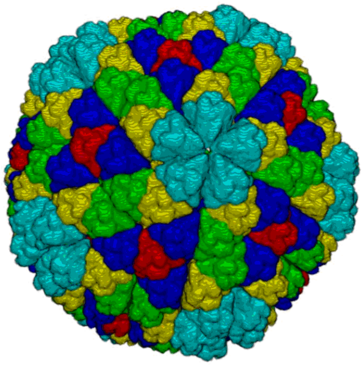
Volume Rover$_{1.1.2}$





September 17, 2007

Computational Visualization Center

Institute for Computational Engineering and Sciences & Department of Computer Sciences

The University of Texas at Austin

http://www.ices.utexas.edu/cvc

Contents

1 What is Volume Rover?

The Volume Rover is an interactive visualization client that uses modern programmable graphics hardware to provide combined geometry and volume rendering displays. Normal rendering software can't display large datasets because of memory and processor limitations.

The client uses a multi-resolution zoom feature that allows users to view arbitrarily large datasets, but visualizing subvolumes from the dataset.

The client runs in two modes: stand-alone, and as a front end to our parallel rendering servers.

2 Requirements

The following are the requirements for installing the Volume Rover:

- GCC 3.x+
- Qt 3.3.x (free for download for Linux from http://www.trolltech.com)
- An NVIDIA GeForce 3 card, ATI Radeon 9700 card, or greater
- CORBA libraries, if you plan to interface with the parallel rendering servers.

3 VolRover distribution

The following programs are distributed with VolRover:

- CompServ - Remote computation server
- RawIVEditor - Header editor for RawIV format volume files
- Vol2Raw - Extracts isosurface from volume and outputs triangulation
- VolumeRover - the main Volume Rover program binary

4 Compilation

This section is only in case you have the source code. To obtain the source code contact Dr. Bajaj at bajaj@ices.utexas.edu. Volume Rover is developed using the Qt toolkit provided by Trolltech, and relies on qmake to generate makefiles for target build platforms. It is made to work with Qt 3.3.x but should work with any version 3.x. If you want to use CORBA servers and functionality, create an environment variable named OOCDIR and have it point to the directory where CORBA is located. If you do not want CORBA functionality, do not create the OOCDIR variable. The version of CORBA we use is ORBacus 4.0.5.

To build Volume Rover and it's associated binaries, in the root directory of the Volume Rover source code, type: **qmake; make** On Windows, Volume Rover can be built using Visual C++ 6.0. To do this, open the file NewVolume.dsw in the Volume directory. Set the active project to NewVolume and build. As with the qmake build process, you can create an environment variable OOCDIR that points to the directory where CORBA is installed.

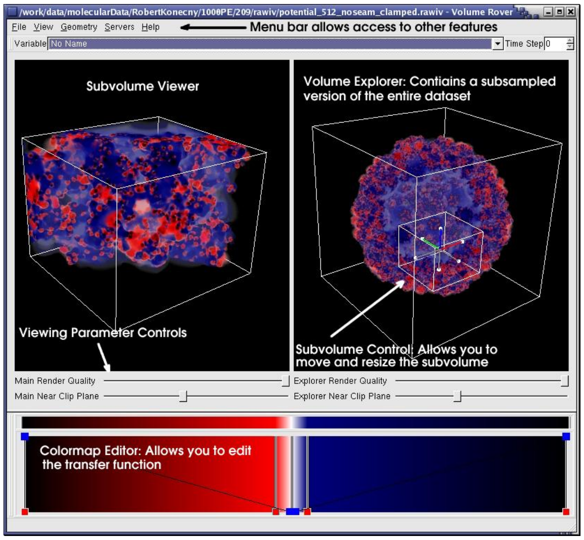
5 Interface Description

This is the interface of Volume Rover when it's first loaded up. It describes the user interface components. We will refer to these components by the labeled names as shown in Figure 1.

6 Options

There are two sliders below each render subwindow. The top one controls the render quality. Moving it all the way to the left selects the lowest render quality while moving it all the way to the right selects the highest render quality. The lowest render quality is sometimes useful for making a dataset more transparent. It is also useful for speeding up drawing when the view of the volume or the transfer function is being manipulated. The bottom slider controls the position of the near clipping plane. If you move it all the way to the left, none of the volume is clipped. Move it all the way to the right and all of the volume is clipped.

Drawing of the wireframe volume bounding box can be enabled or disabled from the **Show Wire Cube** option in the **View** menu.



/work/data/molecularData/RobertKonecny/1000PE/209/rawiv/potential_512_noseam_clamped.rawiv - Volume Rover
File View Geometry Servers Help
Main Render Quality
Main Near Clip Plane
Explorer Render Quality
Explorer Near Clip Plane

Variable No Name

Time Step 0

Menu bar allows access to other features

Subvolume Viewer

Volume Explorer: Contiains a subsampled version of the entire dataset

Viewing Parameter Controls

Subvolume Control: Allows you to move and resize the subvolume

Colormap Editor: Allows you to edit the transfer function

Figure 1: Volume Rover interface

Wireframe rendering of loaded geometry and isosurfaces can be enabled or disabled from the **Wireframe Rendering** option in the **Geometry** menu.

General settings for Volume Rover can be accessed by selecting **Options** from the **File** menu. There are four types of options:

- The **Update Method** option controls how the subvolume is updated as the subvolume control is manipulated. Interactive means that the subvolume is updated as it changes. **Delayed** means that the subvolume is updated after changes have been made. **Manual** means that the subvolume is only updated when **Update** is selected from the **View** menu.

- The **Isosurfacing** option has two non-mutually-exclusive options. They are **Show Thumbnail Isosurface** and **Show Magnified Isosurface**. These options control whether or not isosurfaces are extracted and rendered in the left (magnified) or right (thumbnail) subwindows.

- The **Render Style** option determines how the volume is rendered. **Single Variable** means that only one variable is visualized at a time and that the transfer function will be used to map that variable's densities to colors and opacities. **RGBA Combined** means that four variables will be combined into one volume where each variable represents a color component red, green, blue, or alpha. In this mode, the mappings of variables to red, green, blue or alpha can be modified via a tool bar that appears below the menu bar. Note that this option is only meaningful for multi-variable datasets.

- The **General** options control where Volume Rover writes its cache files and what the background color of the render subwindows is. The **Cache Directory** option is a path to a directory in which another directory named ”VolumeCache” will be created to hold caches of datasets. The current dataset is unloaded whenever the value of this option changes. The **Background Color** option is straightforward. Click the button to bring up a color selector. The color you select will become the background color for the two render subwindows.

7 Loading Data

To load a volume data file into the Volume Rover, use the menu bar and click on File¿Open. You'll see a standard file selection dialog box. Select the dataset file desired and click on the Open button. The first time you load a data file, the Volume Rover goes through a pre-processing step. During this pre-processing step, the Volume Rover creates a series of ”mip-maps” of the original data and places them in the cache directory selected from the options menu. The series of mip-maps create a valuable cache of the data that allows for subsequent interactive manipulation of the dataset. As long as the original data file is not modified, the cache created in the pre-processing step will be used.

The time taken for pre-processing scales linearly with the size of the dataset. Also, the cache size is proportional to the initial data size (and can easily be several hundreds of megabytes) Please make sure that you have enough free space available in the current working directory.

To load a geometry file into the Volume Rover, use the menu bar and click on **Geometry→Load Geometry**. You'll be presented with a file selection dialog where you can choose a file to open.

Occasionally the geometry will not display after it is loaded. This is usually due to a difference in the scale of the vertex coordinates in the geometry file and the scale of the current view transformation. It can be fixed by opening a volume dataset with a scale comparable to that of the geometry.

Unlike volume data, geometry data can be unloaded from the Volume Rover. Just click on **Geometry***rightarrow***Clear Geometry** on the menu bar.

8 Exploring Volumes

A subsampled version of the whole dataset is shown in the Volume Explorer (the subwindow on the right). The subvolume control encloses the portion of the data that is shown in higher resolution in the subvolume viewer (the subwindow on the left). The subvolume control can be resized by clicking and dragging the axis endpoints (circles at the ends of each axis). It can be translated along the direction of one axis by dragging the desired axis toward the chosen direction. see Figure 2

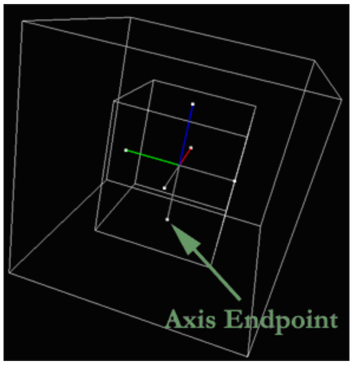


Figure 2: Volume Explorer

9 Manipulating the Transfer Function

The volume rendering transfer function assigns colors and opacities to different densities in the dataset. Your visualization is only as good as your transfer function. Becoming proficient with the Volume Rover's Colormap Editor will serve you well.

Figure 3 highlights the user interface components. To change the opacity function, move the Alpha Nodes (blue squares) around. To move an Alpha Node, left click on the node and drag. To add more Alpha Nodes to your

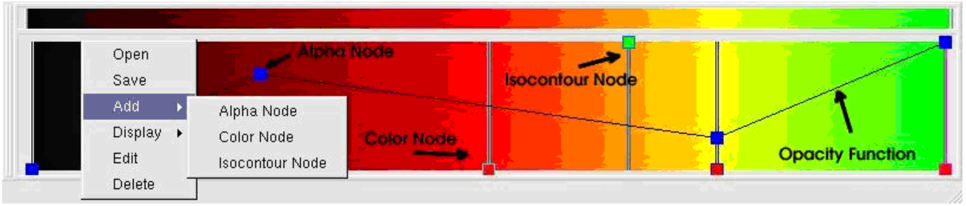


Figure 3: Transfer Function

opacity function, right click anywhere on the colormap editor to bring up the small menu shown in the figure of the Colormap Editor. Then click on **Add→Alpha Node**. Adding more Alpha Nodes gives you greater control over the shape of the opacity function.

To change the color spectrum of the Colormap Editor, one must modify the color nodes (red squares). Changing the color of an existing color node involves right clicking on the color node to display the Colormap Editor Menu, then clicking **Edit** to bring up a standard color selection dialog. To add a color node, bring up the Colormap Editor Menu and select **Add→Color** Node. You can change the color of the node by following the procedure for editing. Color nodes can be moved left and right along the Colormap Editor by left clicking and dragging.

Visualize isocontours by adding, editing and moving isocontour nodes (green squares) in the Colormap Editor [12]. To add an isocontour node, bring up the Colormap Editor Menu and select **Add→Isocontour Node**. To change the color of an existing isocontour node, right click on the node to display the Colormap Editor Menu and click **Edit** to bring up a standard color selection dialog. Isocontour nodes can be moved left and right along the Colormap Editor by left clicking and dragging in the direction desired.

For a particular dataset, arriving at a good transfer function is a trial and error process. It is the most time consuming part of using the Volume Rover. Therefore, the Colormap Editor's settings can be saved and loaded up later. Right click on the Colormap Editor to bring up the menu and click on **Save**. The suffix for transfer function files is *.vinay. Transfer function files must be called *.vinay. A contour spectrum can be computed and displayed for any dataset or any part of a dataset when working with rawv files [13]. To achieve this, right click on the Colormap Editor and select **Display→Contour Spectrum**.

Similarly, a contour tree can be computed and displayed for any dataset or part of a dataset [14]. To do so, right click on the Colormap Editor and select **Display→Contour Tree**.

10 Saving Files

You can create a new dataset in Volume Rover by saving the current subvolume. This writes the volume visible in the left subwindow to a new rawiv, rawv, or MRC file. To do this, click on the File menu and select **Save Subvolume**.

At any time, an image of the left or right subwindow can be saved. The formats available for saving are dependent on what the Qt library supports, so check your local installation for more details. To access this feature, select **Save Image**... from the **File** menu. You will be asked to select which subwindow to save the image from as well as which format to write the image in. After doing those two things and clicking OK, you will be presented with a standard file save dialog.

Isosurfaces that have been extracted from a dataset can be written to one of the four types of raw geometry files. To do this, select **Export Thumbnail Isosurface** or **Export Subvolume Isosurface** from the **Geometry** menu. The thumbnail isosurface is the one on the right side, and the subvolume isosurface is the one on the left side. You will be asked to choose a file type for the new file. If you are viewing an RGBA dataset and you wish to preserve the isosurface's colors, then select either rawc or rawnc. In most other cases, raw or rawn will suffice.

11 Animation

Volume Rover provides a basic interface for animating camera paths as well as a couple of other rendering parameters. Please note that all animation actions take place in the Volume Explorer (right sub-window). Volume Rover's animation abilities can be accessed from the Animation Menu, and consist of the following:

- **Start Recording** begins the recording process. All changes to the camera rotation, zoom, and position will be recorded. The position of the near clipping plane and whether or not isosurfaces are drawn as wireframes are also recorded.

- **Stop Recording** stops the recording process.
- **Play Animation** plays back the current animation.
- **Stop Animation** stops playback of an animation.
- **Save Animation** saves the camera path and other animation keys to a text file. It does not create a movie file that can be viewed in a media player.
- **Load Animation** loads a camera path and animation keys from a previously saved text file.
- **Save Frame Sequence** saves the rendered frames of the animation as a sequence of PPM files.You supply it with a root filename and it will add a unique number and filename extension to each frame that it writes. For convenience, you should probably create a folder to contain the animation. Animations are rendered at "30 frames per second". This means that the sequences of frames should be played back at 30 frames per second. Actual rendering is done as fast as possible.

12 Using Render Servers

Volume Rover can operate as a client to remote render servers. In this mode, Volume Rover assists in creating the transfer function and positioning the camera for a higher resolution rendering.

Once a render server is running, there should be a ref file that tells CORBA how to connect to the server. This ref file should be copied to the directory that Volume Rover was launched from (its current working directory). After this has been done, select **Connect...** from the **Servers** menu. This brings up a dialog where you must choose what type of render server you are connecting to. When you make your selection and click OK, another dialog will come up. This is the **Server Settings** dialog.

The **Server Settings** dialog is different for each type of render server. Because the settings are specific to the particular server, you should consult the documentation for the server for more details.

Once a connection has been established and settings have been specified, the server is ready to render images. To render an image, select **Render Frame** from the **Servers** menu. Volume Rover will appear to lock up for the duration of the render so do not be alarmed. When the render is complete a window containing the final rendered image will open.

To end a session with a render server, select **Disconnect** from the Servers menu. This will close the CORBA connection to the server.

13 Supported Functionalities

- Contrast
- Filtering - Bilateral and Anisotropic
- Segmentation - Asymmetric, Symmetric, and monomer
- Tiling

14 Contrast Enhancement

Our method of Adaptive Contrast Enhancement assigns a new intensity to each pixel according to an adaptive transfer function that is designed on the basis of the local statistics (local minimum/maximum as well as local average intensity) [1]. Contrast Enhancement has Resistor parameter. It is a number from [0-1]. A smaller resistor value allows finer details to be enhanced. Figure 4 shows the result of contrast enhancement on an example virus map using the default resistor value.

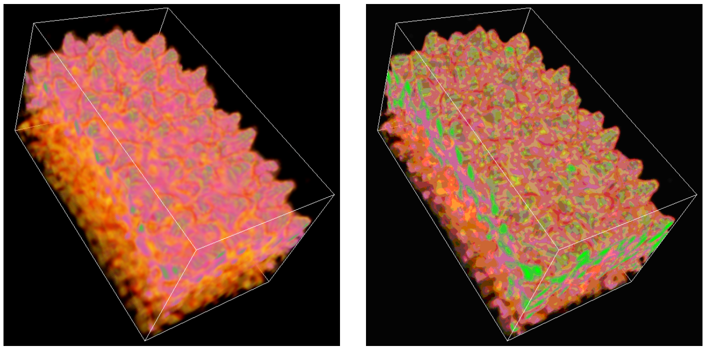


Figure 4: pre-Contrast Enhancement shown on the left and post-Contrast Enhancement shown on the right

15 Anisotropic Diffusion

Anisotropic diffusion is used in image processing for its efficiency of smoothing noise while preserving sharp edges [6]. The only parameter for anisotropic diffusion is currently the number of iterations over the dataset. The default of 20 is good for many situations. See Figure 5.

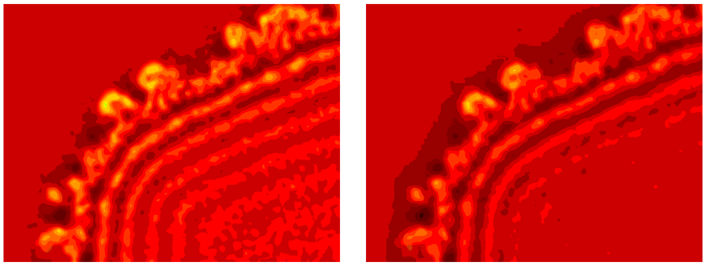


Figure 5: pre-Anisotropic Diffusion shown at left and post-Anisotropic Diffusion shown at right

16 Bilateral Filter

Bilateral Filtering is a simple non-iterative scheme for edge-preserving smoothing [2]. You can apply a bilateral filter to the subvolume's data by selecting **Bilateral Filter** from the **Tools** menu. The filter is applied to the subvolume displayed in the left window. Any change to this subvolume from the subvolume control in the right window will loose the filter.

The three parameters needed to run Bilateral Filtering are Radiometric Sigma, Spacial Sigma, and Filter Radius. The Radiometric Sigma controls the discrimination power between true features and noises with the assumption that larger pixel intensity value variations are mainly from true features and smaller pixel intensity value variations are contributed by noise. The Spacial Sigma controls the extent of the normal spacial low pass filtering in pixels (where a larger value causes severe smoothing). The Filter Radius contros the number of slices of the volume to filter at once. See Figure 6

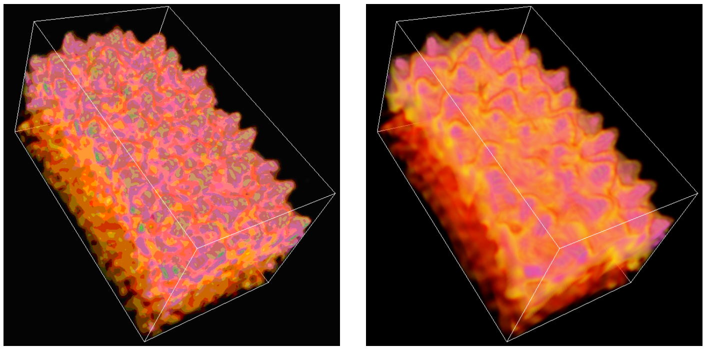


Figure 6: pre-Bilateral Diffusion shown at left and post-Bilateral Diffusion (using default parameters) shown at right

17 Segmentation

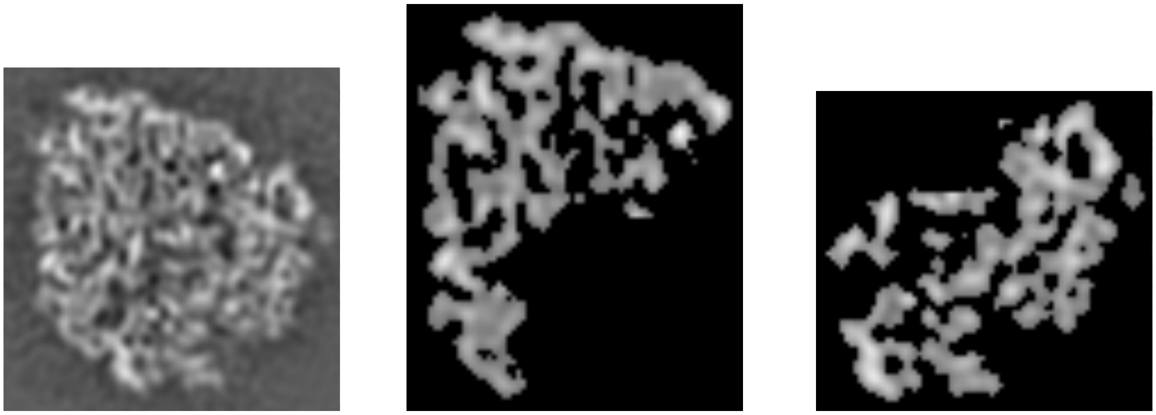
There are two methods of segmentation implemented in VolRover. The first method is a general segmentation that takes a set of seed points from the user as input. The second method is a more automated algorithm that assumes the data exhibits icosahedral symmetry, and is specifically designed to segment icosahedral viruses [4].

Segmentation can be run either locally, or remotely. The benefit of this is that if the local machine does not have enough RAM to run the memory intensive general segmentation routine, it can be run remotely on high end hardware.

17.1 General Segmentation

To run general segmentation, first create 2 or more seed point classes. The procedure for doing this is described in section 19. Every seed point class represents a specific subportion of the volume that you want to segment. For every point class, the program will output a volume. Next choose a low and high segmentation threshold, everything outside these bounds will be thrown away. Then decide if you want to run it remotely or locally. If locally, simply click run. If remotely, first run the program CompServ on the remote machine. Then for hostname and port, use the hostname of the remote machine, and the port that you ran CompServ on. Finally set the path of the remote volume file to load that corresponds to the local volume file, then click run. Refer to the console from which VolumeRover is run for general segmentation output.

Suppose we are trying to segment a file called ”dataFile.rawiv” into two parts, then the output will be two new files called ”dataFile_subunit00.rawiv” and ”dataFile_subunit01.rawiv”. An example of a segmentation of the ribosome is shown in Figure 7.



The combined 70s ribosome The segmented 50s subunit The segmented 30s subunit

Figure 7: An example segmentation of the ribosome. These images are all of the same slice.

17.2 Segment icosahedral data

The Segment Virus Map interface is a front end to 3 separate virus segmentation routines. To properly segment the dataset, segment the map according to the following pipeline: Capsid segmentation → Subunit segmentation → Monomer segmentation (SegSubunit relies on output from SegCapsid, and SegMonomer relies on output from SegSubunit).

17.3 Capsid Segmentation

Capsid segmentation is available from **tools→Segment Virus Map** and has 4 modes of operation depending on the capsid layer type, see Figure 8. There is a check box that allows the user to run anisotropic diffusion on the data before running the segmentation algorithm.

- If there is only 1 capsid layer, and it is distinct from other data, then select ”Single Capsid, distinct” and enter the voxel value of the capsid layer and a point that lies inside the capsid layer.

- If the capsid layer is not distinct, then you must select ”Single Capsid” and enter 2 seed points, one inside the capsid layer, and one inside the genomic structure. The two seeds are usually close to each other where the capsid is most indistinctive from the genomic structure. You may use the tool VolumeGridRover to find the coordinates of the 2 seed points you want to select.

- If the virus structure has a double capsid, it can be segmented by initially using ”Double Capsid, Initial Segmentation.” This process must be run after first running one of the Single Capsid segmentation routines. Provide the estimated small and large capsid radii (in voxels) as a hint to the segmentation routine. You may use the VolumeGridRover tool to determine good values for these radii.

- After running ”Double Capsid, Initial Segmentation,” one may refine that result by running ”Double Capsid, Refined Segmentation.” This routine uses information from ”Double Capsid, Initial Segmentation,” so be sure that you run that first.

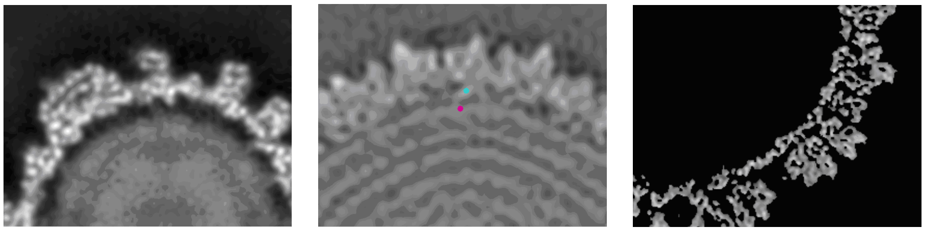


Figure 8: A single and distinct capsid at left, a single but not distinct capsid shown in the middle, and a double capsid shown at right.

17.4 SegSubunit

SegSubunit has 6 different arguments. Subunit segmentation is performed as follows:

1. Decide h and k numbers: These numbers must be visually detected. Adjust the transfer function to clearly display the structure of the capsid layer. Locate the 5-fold symmetry axes (see Figure 9a). There are 12 such axes. Rotate the map to a view as shown in Figure 9b. Locate the 6-fold symmetry axes. Draw two lines that go through the chosen 5-fold axis and its neighboring 6-fold axes as shown in Figure 9b.

   - These two lines will be used to define a coordinate system similar to the standard Euclidian coordinate system. The unit length of this coordinate system is the distance from one 6-fold axis to an adjoining 6-fold axis.
   - We can now decide the coordinate of one of the neighboring 5-fold axes. In this example the coordinate is (7, 7), meaning that h = 7 and k = 7.

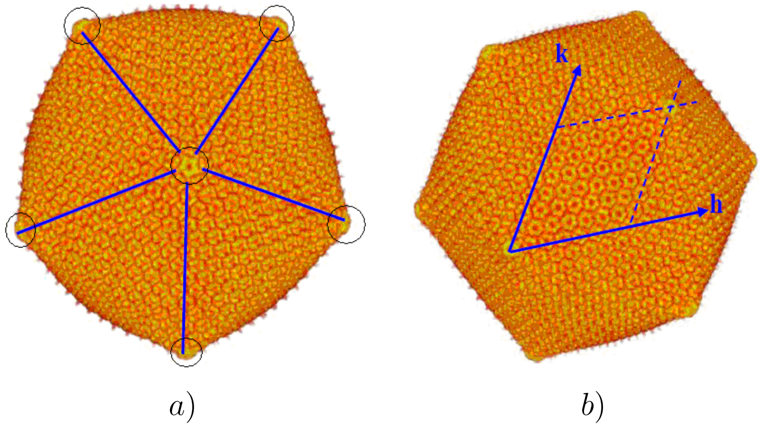
2. Decide the 3-fold number: Are there subunits located at the 3-fold symmetry axes? If yes then 3-fold = 3, if no then 3-fold = 0. In the example shown in Figure 9, the answer is no.

3. Decide the 5-fold number: Are there any 5-fold subunits? If yes, 5-fold = 1. Otherwise, 5-fold = 0.

4. Decide the 6-fold number: Are there subunits located at the 6-fold symmetry axes? If yes then 6-fold = 6, if no then 6-fold = 0. In the example show in Figure 9, the answer is yes.

The output of SegSubunit is as follows:

- test_index.rawiv (a new volume where the densities are assigned a value based on the segmentation index)
- test_seg.rawv (coloring map)
- test_3f_subavg.rawiv or test_6f_subavg.rawiv (averaged subunit)
- test_5f_subunit.rawiv (segmented 5-fold subunit)
- test_matrix.txt (transformation matrices)
- comatrix_new.txt: the similarity/transformation table between subunits



a)

b)

Figure 9: a) The 5 fold axis of symmetry are circled. b) the h and k axes are drawn.

- three_fold_refine.txt/six_fold_refine.txt: detected local symmetry axes.
- SymmetryAxis_refine.raw: the symmetry axes mesh

17.5 SegMonomer

SegMonomer has 1 argument, the fold number of a segmented subunit. For example, if the subunit is a trimer, the fold number is 3. If the subunit is a penton, the fold number is 5.

17.6 P22 example

With our release of VolRover we have included a sample data set of the P22 virus (EMD 1101, resolution 9.5A). Here is a step by step guide to working with an icosahedral virus. See figure 10 for images of P22 results.

1. If you are getting a map file directly from the data base, select **file→open** choose **all file types** to display the map file and then open it. In the right window maximize the subvolume control widget so that the entire volume is displayed in the left window. Then select **file→save subvolume**. This creates a file in .rawiv format that we will use. NOTE: for the P22 data we've included the data is already in .rawiv format, you may begin with the next step.

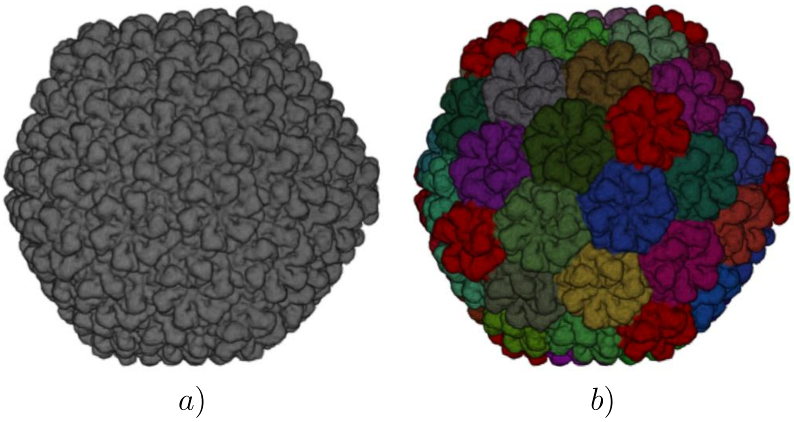


Figure 10: a) P22 capsid segmented. b) P22 subunits segmented.

2. To segment the capsid from the rest of the data, open the .rawiv file and go to the slice viewer. Use the slice viewer to find an appropriate lower bound threshold and seed point(s) as described above. For our P22 data, we use single distinct capsid segmentation with seed point (305, 80, 239) and threshold 130. We encourage you to try your own values and look at the results. It is often easiest to use the slice viewer when the data is displayed in grey scale. There is a checkbox to the right that controls this option.

3. Enter the values you've chosen into the capsid segmentation window and press run. This will take several minutes, possibly up to half an hour for very large data sets.

4. When this is complete, open the capsid output. In this case the output will be called p22_capsid.rawiv. Examine the output visually to see if this is a good segmentation or if you need to go back and find new values for the capsid segmentation.

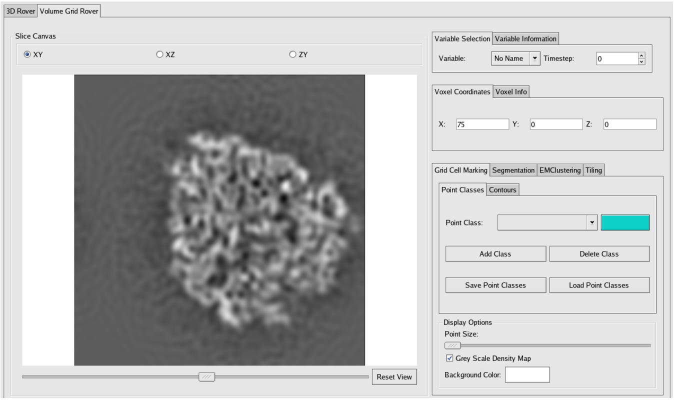
5. Find the values for h, k, 3-fold, 5-fold, 6-fold as described above. For our P22 data, use H=2, K=1, 3-fold=0, 5-fold=1, 6-fold=6, radius=5

6. Enter the values into the segment subunit window and press run. This process shouldn't take more than 10 minutes.

7. If applicable, you may run the segment monomer code. For our P22

data there are both 5-fold and 6-fold subunits. So open the subunit and enter the corresponding folding number into the box and press run.

18 Volume Grid Rover



● XY ○ XZ ○ ZY

Variable: No Name Timestep: 0

X: 75 Y: 0 Z: 0

Point Class:

☑ Grey Scale Density Map

Background Color:

Figure 11: Volume Grid Rover Interface

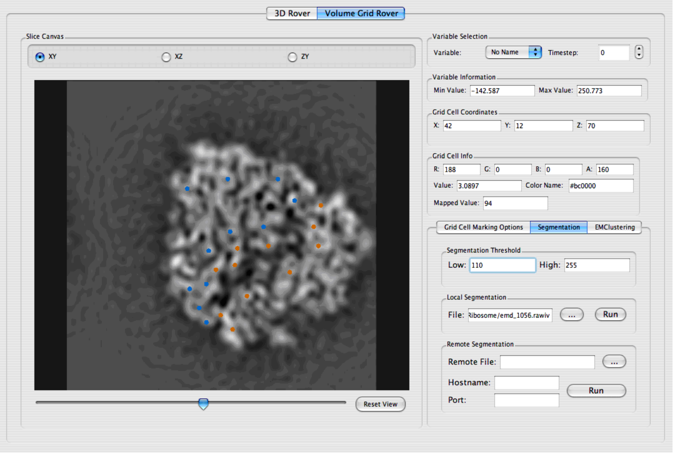
Volume Grid Rover is a 2D volume browser that lets you view volumes slice by slice. The slice canvas is where the volume slices are displayed. At the top of the Slice Canvas, you may select which direction to take slices from. In the initial mode XY, slices are taken in the Z direction, meaning that as you slide the depth slider in this mode, Z values are incremented. Similarly, for XZ and ZY, slices are taken in the Y and X directions respectively.

As you move the mouse over the volume slice in the Slice Canvas, the Grid Cell Coordinates will be updated, showing the current mouse position in the volume. Also, the Grid Cell Info will be updated to show the current voxel value, the voxel value mapped to [0-255] (for indexing the color table), and the color of the voxel (from the color table).

Holding the middle mouse button and moving the mouse up and down will zoom out and in respectively. Holding the right mouse button and moving the mouse in any direction will translate the slice in that direction. Press the Reset View button to re-center the slice on the slice canvas.

Checking the Grey Scale Density Map box will cause volume slices to be drawn using a grey scale color table. It is usually much easier to work with a slice while viewing in grey scale than it is to find an appropriate color transfer function. Darker voxels are closer to the minimum density value, while brighter voxels are closer to the maximum density value.

19 Grid Cell Marking



● XY ○ XZ ○ ZY

Variable: No Name Timestep: 0

Min Value: -142.587 Max Value: 250.773

X: 42 Y: 12 Z: 70

R: 188 G: 0 B: 0 A: 160

Value: 3.0897 Color Name: #bc0000

Mapped Value: 94

Low: 110 High: 255

File: Ribosome/emd_1056.rawiv

Remote File:

Hostname:

Port:

Figure 12: Grid Cell Marking

The **Grid Cell Marking Tab** allow the user to define point classes for segmentation purposes. To create a point class, click **Add Class**. Select a suitable color for that class. To add points, simply double click on the volume slice. To remove points double click on the point to be removed, zooming in will often make this task easier. You may increase the size of points by sliding the **Point Size** slider to the right.

You may save a point set and load it again for later modification. A useful trick using this feature is to run a segmentation, open one of the output files and then load the point set while viewing the output file. This can help you determine what changes need to be made to the point set.

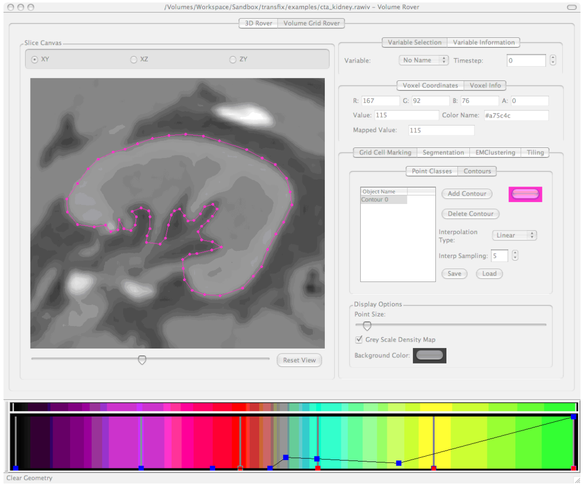
20 EM Clustering

EM Clustering is another segmentation tool that uses the defined point classes to identify the range of each material in voxel values [3]. To use it, first define a point class for each material you want to identify. Then simply go to the EM Clustering tab and click run. Output will be given in the terminal which Volume Rover is run from. Note: In the release version of VolumeRover, this functionality has been disabled, as it is incomplete.

21 Tiling

The main purpose of tiling is to reconstruct surfaces from raw volumetric data. Tiling provides the ability to first define a set of 2D contours on several slices of the loaded volume, and then construct a triangular mesh from those 2D contours [10].

There are many possible shapes for the same input data. So the criterion is a single-sheeted surface. This leads to the following result: If the projection of two contours of the adjacent slices overlaps, and they are of the same level (the number of enclosing contours), then they are constructed as a linked shape. If the projections of two contours separate, they will be reconstructed as disjoint objects. This algorithm works very well in the cases of densely sampled Z. Please see the *Limitation* section of the cited paper to avoid undesirable results.



/Volumes/Workspace/Sandbox/transfix/examples/cta_kidney.rawiv – Volume Rover

◉ XY ○ XZ ○ ZY

Variable: No Name Timestep: 0

R: 167 G: 92 B: 76 A: 0

Value: 115 Color Name: #a75c4c

Mapped Value: 115

Object Name: Contour 0

Interpolation Type: Linear

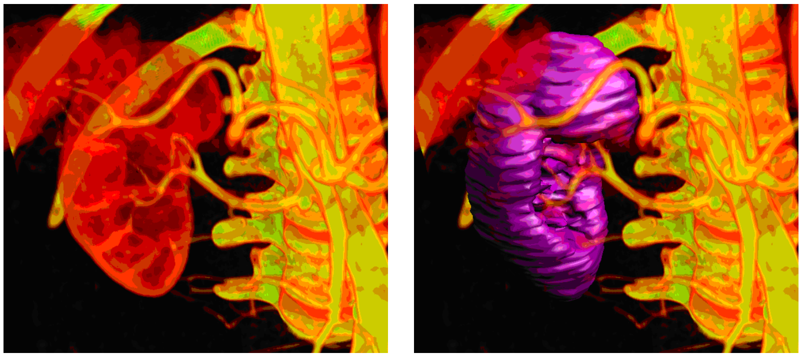
Interp Sampling: 5

☑ Grey Scale Density Map

Figure 13: Tiling Interface

To create 2D contours along each slice, first navigate to the slice you want to draw a contour on using the Depth Slider. Then click Add Contour. Afterward, as you click on the slice, you will notice that a new point will be added and a line segment will connect the new point to the previous point in the contour. To close the loop, click on the very first contour. If you wish to modify a point after it has been created, you must first select it by either clicking on it, or dragging a box around it. To translate the set of selected points, hold the Shift+Alt keys and while holding the left mouse button, drag the mouse in the direction you wish to translate. To delete a point, press the delete key. If you find the points difficult to select or manipulate because they are too small, you may change the point size by adjusting the Point Size Slider. If, after closing a loop, you wish to draw a second loop around another disconnected component of the same object in the volume, you may simply start clicking around the object and another loop will be formed.

To finally run tiling on the set of 2D contours, click on the Tiling tab and click Run. By default, the tiling output will be directly rendered in the 3D rover window. However there are instances where the number of meshes to be output have many more triangles than can be efficiently rendered, so it's best to write them directly to disk. In that case, you may click To Files and enter an output directory path. The filenames for each output mesh will contain the names of each contour.



Before tiling.

After Tiling.

Figure 14: Before and after images of tiling.

22 Supported Data Formats

22.1 RawIV

The rawiv data format is used to represent 3D volumetric data of scalar fields defined on a regular grid. A rawiv file is created by adding the header to the raw format. Everything is in big-endian. Big endian is the byte order on Sun, SGI, IBM architectures. Intel's byte order is little endian. The suffix on the name of a rawiv file is .rawiv

22.1.1 header

Order of information is as follows, concatenated contiguously.
(minX)(minY)(minZ)(maxX)(maxY)(maxZ)(numVerts) (numCells)(dimX) (dimY)(dimZ)(originX)(originY)(originZ) (spanX)(spanY)(spanZ)

- (minX, minY, minZ) are the co-ordinates of the 1st voxel.
  (maxX, maxY, maxZ) are the co-ordinates of the last voxel.
  The mins, and maxs are floats. These define the bounding box of the data in co-ordinate space.

- numVerts is the number of vertices in the grid.
  numVerts = dimX×dimY×dimZ
  numVerts is an unsigned int.

- numCells is the number of cells in the grid.
  numCells = (dimX - 1)×(dimY - 1)×(dimZ - 1)
  numCells is an unsigned int.

- dimX = number of vertices in x direction
  dimY = number of vertices in y direction
  dimZ = number of vertices in z direction
  The dims are unsigned ints.

- originX
  originY
  originZ
  The origins are floats.
  The existence of the origin co-ordinates is somewhat of a mystery. Some developers claim the origin co-ordinates are exactly the same as the co-ordinates of the first voxel.

- The spans are the spacing between one vertex and the next along the given description.
  spanX = (maxX - minX)/(dimX - 1)
  spanY = (maxY - minY)/(dimY - 1)
  spanZ = (maxZ - minY)/(dimZ - 1)
  The spans are all floats.

- The size of a rawiv header is 68 bytes.

- There are a number of fields in the header that are redundant. For example, numVerts = dimX * dimY * dimZ. If while reading the rawiv format, you find that numVerts != dimX * dimY * dimZ , then the appropriate action is to determine that the rawiv file is corrupted.

- A byte is 8 bits. A float is 4 bytes. An unsigned int is 4 bytes. An unsigned short is 2 bytes. A character is a single byte.

22.1.2 Data

The data portion in raw format imidiately follows the header. The raw portion of the rawiv file is in binary big-Endian format used to represent 3D volumetric data of scalar fields defined on a regular grid. It is simply a sequence of values. These values can be floats, unsigned shorts, or unsigned chars. The data is listed with the x co-ordinate varying fastest, and z varying slowest. So, in C++ syntax a reader would contain the following code snippet:

```
for (intz=0; z ¡ dimZ; z++)
    for (int y=0; y ¡ dimY; y++)
        for (int x=0; x ¡ dimX; x++)
        {
            //read data here
        }
```

22.2 RawV

The rawv data format is very similar to rawiv. A rawv file is a binary file consisting of a variable length headera followed by one or more volumetric scalar fields. The rawv file format was created with multi-variable time varying data in mind. All data including the header is big endian.

Offset	**Size**	**Type**	**Description**
0	4	unsigned int	Magic = 0xBAADBEEF
4	4 ×3	unsigned int	XYZ Dimensions
16	4	unsigned int	# of Time steps
20	4	unsigned int	# of Variables
24	4×4	float	min X,Y,Z,T
40	4×4	float	max X,Y,Z,T
56	1	unsigned char	Variable Type 1
57	1×64	char	Variable Name 1
……	…..		
	1	unsigned char	Variable Type n
	1×64	char	Variable Name n
			Volume Data

- Fixed size 121 byte header for single variable datasets
- Variable names are NULL terminated
- Variable types are:
  - 1 for unsigned char (1 byte)
  - 2 for unsigned short (2 bytes)
  - 3 for unsigned int/long (4 bytes)
  - 4 for float (4 bytes)
  - 5 for double (8 bytes)
- RawV is pronounced ”Raw Five”
- Data is stored with the X co-ordinate varying fastest, followed by Y, followed by Z.
- T varies slower than Z
- All of a variable’s time steps are stored contiguously

22.3 Raw Geometry (raw, rawn, rawc, rawnc)

Raw geometry files are simple ASCII files used to represent triangle meshes. They come in four flavors: raw, rawn, rawc, and rawnc. Raw files are just triangles, rawn files are triangles with vertex normals for smooth shading, rawc files are triangles with vertex colors and rawnc files are triangles with vertex normals and vertex colors. See figure 15 for a simple raw file example. The basic file structure is as follows.
<numvertices><numtriangles>
<vertex 0>
⋮
<vertex n>
<triangle 0>
⋮
<triantle m>
EOF

The vertices section differs depending on the file type. They are as follows

- Raw (type: float)
  <vertX><vertY><vertZ>
- Rawn (type: float)
  <vertX><vertY><vertZ><normX><normY><normZ>
- Rawc (type: float)
  <vertX><vertY><vertZ><colorR><colorG><colorB>
  colors are in range [0,1]
- Rawnc (type: float)
  <vertX><vertY><vertZ><normX><normY><normZ><colorR><colorG><colorB>
  colors are in range [0,1]

Each line in the triangles section defines a triangle by references to lines in the above vertices section. Indices may start at 0 or 1. VolRover will load a file marginally faster if the indices start at 0. The values in this section are all of type int.
<vertexIndex><vertexIndex><vertexIndex>

```
3    1
0.0  0.0  0.0
1.0  0.0  0.0
0.0  1.0  0.0
0    1    2
```

Figure 15: An example .raw file.

22.4 MRC

The MRC header has length 1024 bytes.

Table 1: MRC header format

SIZE	DATA	NAME	DESCRIPTION
4	int	NX	Number of columns (fastest changing in map)
4	int	NY	Number of rows
4	int	NZ	Number of sections (slowest changing in map)
4	int	MODE	Types of pixel in image
			0 = Image unsigned bytes
			1 = Images signed short integer (16 bits)
			2 = Image float
			3 = Complex short×2
			4 = Comples float×2
4	int	NXSTART	Number of first COLUMN in map (Default = 0)
4	int	NYSTART	Number of first ROW in map (Default = 0)
4	int	NZSTART	Number of first SECTION in map (Default =0)
4	int	MX	Number of intervals along X
4	int	MY	Number of intervals along Y
4	int	MZ	Number of intervals along z
4	float	XLEN	Cell dimensions (Anstroms)
4	float	YLEN	Cell dimensions (Anstroms)
4	float	ZLEN	Cell dimensions (Anstroms)
4	float	ALPHA	Cell Angles (Degrees)
4	float	BETA	Cell Angles (Degrees)
4	float	GAMMA	Cell Angles (Degrees)
4	int	MAPC	Which axis corresponds to Columns (1,2,3 for X,Y,Z)
4	int	MAPR	Which axis corresponds to Rows (1,2,3 for X,Y,Z)

MRC header format table...continued from previous page.

SIZE	**DATA**	**NAME**	**DESCRIPTION**
4	int	MAPS	Which axis corresponds to Sections (1,2,3 for X,Y,Z)
4	float	AMIN	Minimum density value
4	float	AMAX	Maximum density value
4	float	AMEAN	Mean desnisty value
2	short	ISPG	Space group number (0 for images)
2	short	NSYMBT	Number of bytes used for storing symmetry operators
4	int	NEXT	Number of bytes in extended header
2	short	CREATID	Creator ID
30		EXTRA	Not used. All set to zero by default
SIZE	DATA	NAME	DESCRIPTION
2	short	NINT	Number of integers per section
2	short	NREAL	Number of reals per section
28		EXTRA2	Not used. All set to zero by default
2	short	IDTYPE	0=mono, 1=tilt, 2=tilts, 3=lina, 4=lins
2	short	LENS	
2	short	ND1	
2	short	ND2	
2	short	VD1	
2	short	VD2	
24	float	TILTANGLES	Used to rotated model to match new rotated image
4	float	XORIGIN	Origin of image
4	float	YORIGIN	Origin of image
4	float	ZORIGIN	Origin of image
4	char	CMAP	Contains 'MAP'
4	char	STAMP	Machine stamp
4	float	RMS	Deviation of map from mean density
4	int	NLABL	Number of labels being used
800	char		10 labels of 80 character

References

23 References

23.1 Filtering and Contrast Enhancement

[1] Z. Yu, C. Bajaj **A Fast and Adaptive Algorithm for Image Contrast Enhancement** *Procedings of 2004 IEEE International Conference on Image Processing (ICIP'04), Volume 2, Oct. 24-27 2004, Pages 1001-1004, Singapore.*

[2] W. Jiang, M. Baker, Q. Wu, C. Bajaj, W. Chiu **Applications of Bilateral Denoising Filter in Biological Electron Microscopy** *Journal of Structural Biology, Volume 144, Issues 1-2, October 2003, Pages 114-122.*

23.2 Classification

[3] S. Park, C. Bajaj, G. Gladish **Artery-Vein Separation of Human Vasculature from 3D Thoracic CT Angio Scans** *CompIMAGE. Coimbra, Portugal. October 20-21, 2006.*

23.3 Segmentation

[4] M. Baker, Z. Yu, W. Chiu, C. Bajaj **Automated Segmentation of Molecular Subunits in Electron Cryomicroscopy Density Maps** *Journal of Structural Biology, 2006. Accepted.*

[5] Z. Yu, C. Bajaj **Automatic Ultrastructure Segmentation of Reconstructed CryoEM Maps of Icosahedral Viruses** *IEEE Transactions on Image Processing: SpecialIssue on Molecular and Cellular Bioimaging, 2005 Sep;14(9):1324-37.*

[6] Z. Yu, C. Bajaj **Anisotropic Vector Diffusion in Image Smoothing** *Proceeding of the 9th IEEE International Conference on Image Processing, vol. 1, pp. 828-831, (Published by IEEE CS Press), Rochester, New York, September 22-25, 2002.*

[7] Z. Yu, C. Bajaj **Normalized Gradient Vector Diffusion and Image Segmentation** *Procedings of the 7th European Conference on Computer Vision (LNCS 2352), vol. 3, pp. 517-530, (Published by Springer Press), Copenhagen, Denmark, May 28-31, 2002.*

[8] Z. Yu, C. Bajaj **Computational Approaches for Automatic Structural Analysis of Large Bio-molecular Complexes** *Accepted for publication in IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2007.*

[9] C. Bajaj, Z. Yu **Geometric and Signal Processing of Reconstructed 3D Maps of Molecular Complexes** *Handbook of Computational Molecular Biology, Edited by S. Aluru, Chapman & Hall/CRC Press, Computer and Information Science Series, December 21, 2005, ISBN: 1584884061.*

23.4 Surface Reconstruction

[10] C. Bajaj, E. Coyle, K. Lin **Arbitrary Topology Shape Reconstruction from Planar Cross Sections** *Graphical Models and Image Processing, 58:6, (1996), 524-543.*

23.5 Visualization

[11] Wilson, Van Gelder, Wilhelms **Direct Volume Rendering via 3D Textures** *Technical Report UCSC-CRL-94-19. Univ. of Calif. Santa Cruz. 1994*

[12] Lorensen, Cline **Marching Cubes: A High Resolution 3D Surface Construction Algorithm** *Proceedings of the*

14th annual conference on Computer graphics and interactive techniques

[13] C. Bajaj, V.Pascucci, and D.Schikore **The Contour Spectrum** *Proceedings of the 1997 IEEE Visualization Conference,167-173, October 1997 Phoenix, Arizona*

[14] C. Bajaj, M. van Kreveld, R. van Oostrum, V. Pascucci, and D. Schikore **Contour Trees and Small Seed Sets for Isosurface Generation** *Topological Data Structures for Surfaces, Chap5, pg 71 - 86, 2004 ed. by S. Rana, John Wiley & Sons, Ltd, 2004*

[15] C. Bajaj, Z. Yu, M. Auer **Volumetric Feature Extraction and Visualization of Tomographic Molecular Imaging** *Journal of Structural Biology, Volume 144, Issues 1-2, October 2003, Pages 132-143.*

24 License Agreement

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

1. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder (THE COMPUTATIONAL VISUALIZATION CENTER at THE UNIVERSITY OF TEXAS AT AUSTIN) or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

   A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

   The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

   "Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

2. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

3. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 2 above, provided that you also meet all of these conditions:

   - The modified work must itself be a software library.
   - You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
   - You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
   - If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful. (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 3d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

   In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

4. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

   Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

   This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

5. You may copy and distribute the Library (or a portion or derivative of it, under Section 3) in object code or executable form under the terms of Sections 2 and 3 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 2 and 3 above on a medium customarily used for software interchange.

   If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

6. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a ”work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

   However, linking a ”work that uses the Library” with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a ”work that uses the library”. The executable is therefore covered by this License. Section 7 states terms for distribution of such executables.

   When a ”work that uses the Library” uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

   If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 7.)

   Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 7 Any executables containing that work also fall under Section 7, whether or not they are linked directly with the Library itself.

7. As an exception to the Sections above, you may also combine or link a ”work that uses the Library” with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer’s own use and reverse engineering for debugging such modifications.

   You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

   - Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 2 and 3 above); and, if the work is an executable linked with the Library, with the complete machine-readable ”work that uses the Library”, as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
   - Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (2) uses at run time a copy of the library already present on the user’s computer system, rather than copying library functions into the executable, and (3) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

- Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 7a, above, for a charge no more than the cost of performing this distribution.
- If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

8. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   - Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
   - Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

9. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

11. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

12. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free

redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

13. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

14. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and ”any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

15. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

CREDITS

WE REQUEST THAT YOU AGREE TO ACKNOWLEDGE THE USE OF THE SOFTWARE THAT RESULTS IN ANY PUBLISHED WORK, INCLUDING SCIENTIFIC PAPERS, FILMS AND VIDEOTAPES BY CITING THE REFERENCES IN CODE FILE AND DOCUMENTATION BOUNDED WITH THE SOFTWARE.

C. Bajaj, Z. Yu, M. Auer
Volumetric Feature Extraction and Visualization of Tomographic Molecular Imaging Journal of Structural Biology, Volume 144, Issues 1-2, October 2003, Pages 132-143. (pdf)

This software has been developed at the Computational Visualization Center at The University of Texas at Austin under

Dr Chandrajit Bajaj
Computational Applied Mathematics Chair in Visualization
Professor of Computer Sciences
Director of Computational Visualization Center
The Institute of Computational Engineering and Sciences
The University of Texas at Austin
201 East 24th Street, ACES 2.324A
1 University Station, C0200
Austin, TX 78712-0027
email: bajaj@ices.utexas.edu
URL: http://www.cs.utexas.edu/users/bajaj/

NO WARRANTY

16. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY ”AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

17. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY

TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

18. WE REQUEST THAT YOU PROVIDE A MENU OR A SCREEN THAT IS TO BE SHOWN WITH OUR CENTER NAME(CVC) AND THE UNIVERSITY OF TEXAS AT AUSTIN TO SHOW THAT WE ARE THE PROVIDER OF THE LIBRARY OR SOURCE CODE. AND WE ALSO REQUEST THAT YOU PROVIDE OUR WORLD WIDE WEB ADDRESS (HTTP://WWW.ICES.UTEXAS.EDU/CVC) IN YOUR SOFTWARE.

19. IF YOU DESIRE TO USE THIS CODE FOR A PROFIT VENTURE, OR IF YOU DO NOT WISH TO ACCEPT THIS LGPL, BUT DESIRE USAGE OF THIS CODE, PLEASE CONTACT CHANDRAJIT BAJAJ AT THE ADDRESS ABOVE FOR A DIFFERENT LICENSE.

    END OF TERMS AND CONDITIONS