

User's Guide

PMOD 3D Rendering Tool (P3D)

Version 3.3



PMOD Technologies

Contents

PMOD 3D Rendering Tool Introduction	2
--	----------

P3D Data Processing	5
----------------------------	----------

Preparation Steps.....	7
Image Data Loading	7
Segmentation Methods.....	8
Object Rendering.....	15
Visualization Options	17
Object Management and Adjustments of Properties	17
Viewing Options for Surface Rendering (SR) Objects	18
Viewing Options for Volume Rendering (VR) Objects.....	19
Image Plane Objects.....	24
Marker Objects	27
Cutting away Parts of the VR or SR Information	28
Rendering of VOIs	30
Scene Operations.....	31
Protocol Files	34

3D Rendering Examples	35
------------------------------	-----------

Example 1: Surface Rendering of a Brain Tumor	35
Example 2: Animated Texture on Cardiac Surface	37
Example 3: Fused Volume Rendering for Angio CT	38

3D Scatter Plots (from PFUS)	41
-------------------------------------	-----------

P3D Configuration	44
--------------------------	-----------

Index	46
--------------	-----------

PMOD 3D Rendering Tool Introduction

Physicians trained in cross-sectional imaging are able to understand the spatial relationship between tissue structures by mere exploration of cross-sectional slice images. However, for communication purposes it is often helpful to generate simulated views of isolated objects of interest from different viewpoints. 3D image processing allows to derive such objects from slice images and calculate virtual reality scenes which can be explored interactively.

Surface Rendering (SR)

One way to derive a virtual object is to track (*segment*) an organ boundary in the slice images, build a 3D surface from the contours, and shade the surface. The P3D tool supports such segmentations by applying thresholds and/or region growing, optionally restricted within volumes-of-interest. As a special option, the object can be colored by projecting the information of matched images onto its surface (*texturing*), even animated in time. This feature allows, for instance, to visualize the concentration changes of the NH_3 perfusion tracer at the myocardium surface throughout a dynamic acquisition.

Volume Rendering (VR)

A different way to derive a virtual object is to take a certain viewpoint in front of the object, cast rays through the object and record the *ray values* when the rays pass a plane behind the object, thereby producing an image. The ray value recorded in the image plane is a combination of the values in all the pixels met along the way from the viewpoint to the image plane, thus the name *Volume Rendering*. Typically, the combination is just the sum of the pixel values, each multiplied by a certain weighting function, called *opacity*. The result depends heavily on the image values and the opacity function. There are very successful configurations available for contrast enhanced CT examinations which provide the illusion of a three-dimensional representation, especially if the viewpoint is interactively changed.

Combination of 3D objects from Matched Series

A unique feature of P3D is the ability to combine and manipulate different types of virtual reality objects (VRO) in one common scene, even when they stem from different studies. Once a scene has been created, it can be saved in a VRML format file (SR objects only). These files can later be loaded into P3D (or an external VRML-Browser) to continue scene exploration. It is in principle also possible to import VRML-scenes from some dedicated 3D rendering programs and combine them with P3D objects, but the import may sometimes fail. So far, AMIRA data has been successfully imported.

CAUTION: When combining multiple studies in a single rendering, please match beforehand all images series in the PFUS tool and save the resampled images. Otherwise, unexpected shifts between the objects might occur in the scene.

Hardware Requirements and Performance

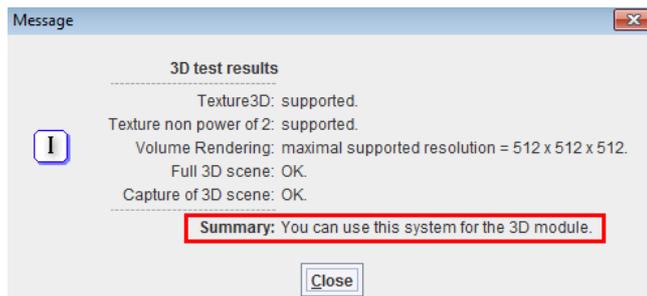
The P3D tool is based on the OpenGL implementation in the *Java3D library* (<http://java.sun.com/products/java-media/3D/index.jsp>). Therefore, using P3D requires a graphics board which supports OpenGL, preferably with at least 512 MB video RAM to accommodate high quality texture displays. As the 3D rendering implementation in Java3D is quite memory demanding (particularly with VR), a 64-Bit operating system and a RAM size of at least 4GB is highly recommended when working with highly resolved or dynamic data. We strongly suggest that you test P3D on your platform with your own data before purchasing this PMOD option.

An automatically **Acceptance test** is taking place when P3D module is first started. There are several 3D scene available for visual inspection and a dialog window opens for each scene. Please answer the questions and do not perform any other operation until test completion.

The following tests are performed:

- 1) test of texture 3D support.
- 2) test of a non power of 2 texture support.
- 3) test of maximum resolution of Volume rendering (from 128x128x128 up to 512x512x512).
- 4) test of a full 3D scene loaded from protocol.
- 5) test of a scene capture.

At the end of all steps the test results will be summarized in a message window. If the test succeeded the results should be similar like the one below:

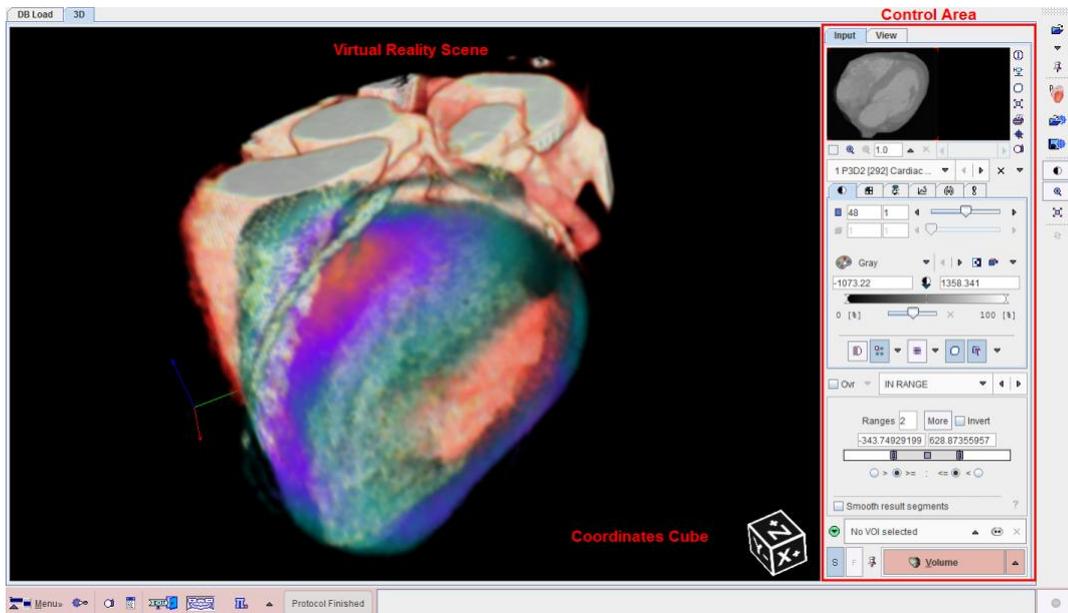


Starting the 3D Rendering Tool

The P3D tool is started from the PMOD ToolBox by clicking the button.



The P3D application window consists of a large display area for showing the 3D scene, a control area organized by the two tabs **Input** and **View**, and a detachable taskbar with shortcuts along the right edge.



P3D Data Processing

Overview

The P3D tool starts with an empty scene. The user can then create SR or VR objects by loading images and applying the available segmentation techniques. As soon as an object is created, it is displayed in the scene with default properties such as color and transparency. If the appearance of an object is not optimal, its properties can be changed and the scene refreshed until the user is satisfied.

Finally, the composition of the objects will show a meaningful scene which can be interactively explored to understand the spatial relationships of the segmented tissue structures. To this end the scene can be rotated in any direction, zoomed, and objects obstructing the view to deeper ones can be temporarily hidden or set to a high degree of transparency. Furthermore, planes showing the original image data can be added, or volumes-of-interest (VOI). To document meaningful results the current rendering can be saved as a screen capture, or a movie of a rotating scene can be generated. It is also possible to save the current configuration into a protocol file, so that it may be reconstructed from the original data at a later time.

Image Data Requirements

There are situations where objects are derived from a single data set, for example the boundary of a brain tumor and the outer brain contour. In this case the spatial alignment of the generated objects will be inherently correct.

However, if objects from different image series are rendered in a single scene, some requirements must be met to ensure that the objects are correctly positioned in the scene. The positioning rules in P3D are as follows:

- 1) The generated objects are positioned relative to the *center* of the scene volume (i.e. the center of the image volume is aligned with the center of the scene).
- 2) For the placement and the size of the objects the pixel size is taken into account.

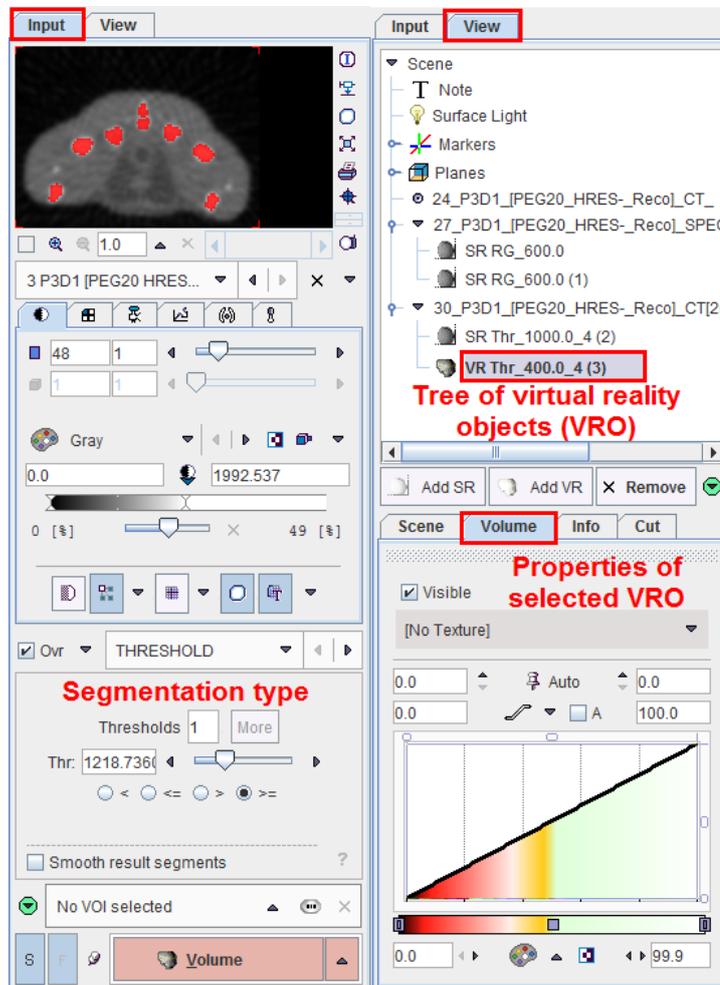
So if the centers of two image volumes are aligned, objects derived from them can be combined, even if the pixel size is not the same.

Note: To ensure this requirement it is highly recommended to match the studies to be combined in the image fusion tool PFUS, save the fused studies, and use them for the 3D rendering.

When importing virtual reality (VRML) objects generated by other programs this behavior may not be adequate. In this case it can be changed in the P3D tool configuration.

Control Tabs

Along the right border there are two control tabs which group the different functions needed for data processing and image rendering. While they are stacked in practice, the graphics below shows them side by side to provide an overview of the user interface elements.



They implement the following functionality.

<p>Input</p>	<ul style="list-style-type: none"> Loading of the different images which can be used for segmentation or texturing. Defining the image color tables, window/level settings, and the current slice/frame of the plane images shown in the scene. Pre-processing of the images, eg. smoothing, masking, scaling. Loading a volume-of-interest (VOI) definition for restricting a segmentation spatially, or for rendering VOIs as 3D objects. Defining the segmentation process. Defining the 3D rendering type (SR, VR, VR HD) to apply. Starting the creation of 3D objects.
---------------------	---

View	<p>Inspecting the available 3D objects which are organized in a tree structure.</p> <p>Modifying the properties of each object (color, texture, visibility, etc.) to adjust the scene.</p> <p>Adding a marker to the scene which highlights an important location.</p> <p>Adding plane images to the scene.</p> <p>Adding light to the scene.</p> <p>Cutting parts out of the objects.</p> <p>Rotating or zooming the scene.</p> <p>Recording a movie of a rotating scene.</p> <p>Saving the SR objects of a scene as a VRML file.</p> <p>Saving screen captures of the scene.</p>
-------------	--

The procedures to generate and manipulate 3D scenes are described in the next sections.

Preparation Steps

Image Data Loading

In P3D image data are used for two different purposes:

- 1) For the derivation of virtual reality objects by segmentation processes (**Input** images).
- 2) For the coloring of the objects (**Texture** images). When using texturing for surface rendering (SR) objects the color at a surface point is derived from the color at the same point in the texture images. When using textures for volume rendering (VR) objects the color is derived by a weighted addition of the texture colors during ray tracing.

The images can be loaded from the **Menu** with **Load Input**, from the **DB Load** page or using the  button the taskbar to the right.



Note that several images can be loaded at once. It is also possible to incrementally load additional images later on. To do so, the append pushpin  below the  button should be activated to enter the *append* mode . The format of the loaded images can be changed with the small  button.

Segmentation Methods

Segmentation is a crucial step of 3D rendering. In P3D this task is done in the lower section of the **Input** tab.

The aim of segmentation is to find the boundary of a tissue structure, which can be transformed into a virtual 3D object. In general, segmentation is a process which is difficult to fully automate. The best results are obtained if the structure is clearly separated from the remainder by a substantial image contrast, and if prior anatomical knowledge is used to guide the segmentation process.

P3D offers several general segmentation methods for finding the object contours. The contour finding process can further be restricted by an enclosing volume-of-interest (VOI). The **Smooth** box allows enabling a smoothing function during the segmentation. Note, however, that smoothing will change the volume of the segments and is thus not recommended for small objects.

Each **Input** study has its own segmentation definition which will be updated when switching between the studies. Therefore, to begin with a segmentation task, first select the appropriate **Input** study, and then adjust the the segmentation method.

ITK-based Implementations

Some of the segmentation methods are based on the use of the ITK (Insight Toolkit) libraries. ITK can be used under the open-source BSD license which allows unrestricted use, including use in commercial products (see www.itk.org). The ITK-based methods are clearly denoted with ITK in brackets.

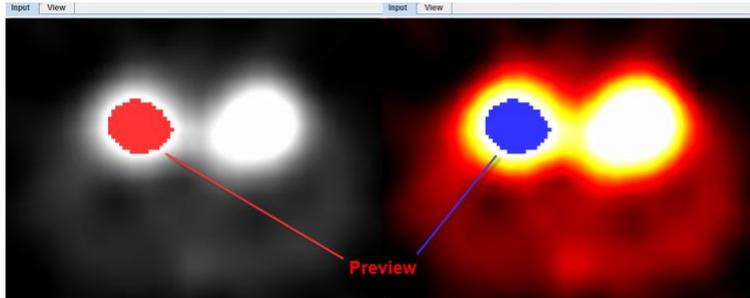
Note: PMOD Technologies cannot be held liable for permanent support of the ITK interface, nor for the performance of the provided libraries.

Region Growing Methods

"Region growing algorithms have proven to be an effective approach for image segmentation. The basic approach of a region growing algorithm is to start from a seed region (typically one or more pixels) that are considered to be inside the object to be segmented. The pixels neighboring this region are evaluated to determine if they should also be considered part of the object. If so, they are added to the region and the process continues as long as new pixels are added to the region. Region growing algorithms vary depending on the criteria used to decide whether a pixel should be included in the region or not, the type of connectivity used to determine neighbors, and the strategy used to visit neighboring pixels." *The ITK Software Guide*. (<http://www.itk.org/ItkSoftwareGuide.pdf>)

Segmentation Preview

As a help during the definition of the segmentation criterion a preview of the expected result is shown in the **Input** display if the **Ovr** (Overlay) box is checked. The example below illustrates **REGION GROWING** after clicking into a mouse kidney. Note how the color of the segmentation overlay changes depending on the color table used to display the images.



To start with segmentation, navigate to a slice (in any direction) which shows the tissue of interest and click with the left mouse button onto a central point. Then adjust the criterion until the overlay indicates a promising segmentation.

Please use the  toggle button in the taskbar for enlarging/shrinking the area of the segmentation preview image. Note that the image display controls are not accessible in the enlarged mode, so the image presentation should be adjusted beforehand.

For further expanding of the 2D view port activate the **Hide segmentation controls**  button located under the segmentation settings. Note that upon activation of this button the **Segmentation** panel collapse and only the segmentation method selection list is still available. Therefore, any further settings should be set beforehand.

CAUTION: The preview only shows the segmentation within the current slice. As there may exist indirect connections through neighboring slices, additional pixels may be contained in the final segment.

REGION GROWING Segmentation

Region growing is a method by which the user defines a starting point (*seed*) within the object of interest, and the algorithm tries to find all connected pixels which fulfill a certain criterion.

After selecting **REGION GROWING** the following control elements appear.



The slider and the number field serve for defining the pixel value. Together with the **Direction** radio button the criterion for pixel inclusion is formed.

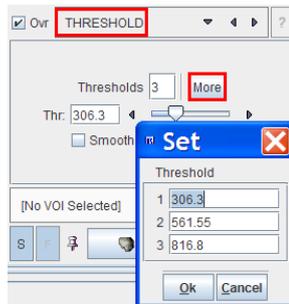
- ▶▶ = include all connected pixels with a value threshold \pm **Deviation**;
- ▶▶ \geq include all connected pixels above the defined threshold value;
- ▶▶ \leq include all connected pixels below the defined threshold value.

A preview of the result is shown in the **Input** display if the **Ovr** (Overlay) box is checked, for example after clicking into a hot spot. To start the segmentation, navigate to a slice (in any direction) which shows the tissue of interest and click with the left mouse button onto a central point. Then adjust the criterion until the overlay indicates a promising segmentation.

THRESHOLD Segmentation

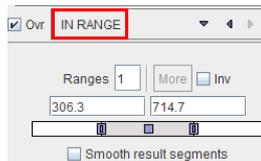
The **THRESHOLD** segmentation is conceptually simple. All pixels above the threshold are included in the segment. Sometimes it is helpful to segment at several threshold levels at once. This can easily be realized by setting the number in the **Thresholds** field accordingly, select **More**, and enter the threshold values in the appearing **Set** dialog.

In the example below, three segments of decreasing volume will be generated at thresholds 306.3, 561.55 and 816.8. During visualization, the user can show or hide each of the segments separately.



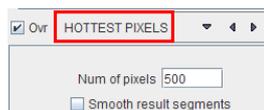
IN RANGE Segmentation

The **IN RANGE** segmentation is similar to the **THRESHOLD** method, except that an upper limit is also defined. This can be helpful to find objects with intermediate intensities. The complementary pixels which are outside the specified range can be obtained by checking the **Inv** box. Similar to the **THRESHOLD** segmentation, multiple **Ranges** can be defined and segmented at once.



HOTTEST PIXELS Segmentation

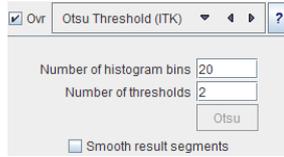
The **HOTTEST PIXELS** segmentation allows to get a 3D object of the (potentially disconnected) pixels with the highest values. The number of included pixels can be specified in the **Num of pixels** field.



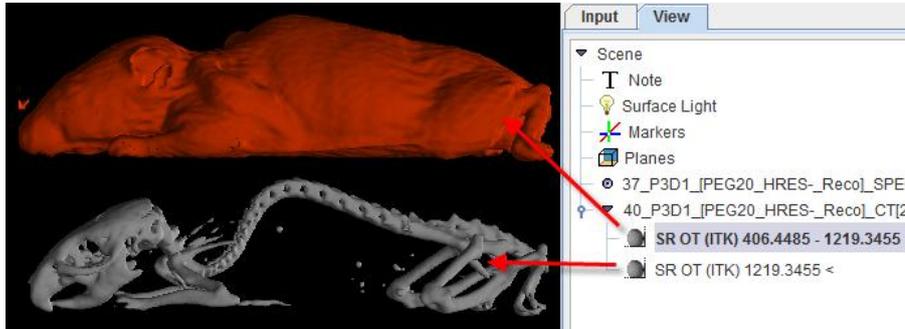
Otsu Threshold (ITK)

"Another criterion for classifying pixels is to minimize the error of misclassification. The goal is to find a threshold that classifies the image into two clusters such that we minimize the area under the histogram for one cluster that lies on the other cluster's side of the threshold. This is equivalent to minimizing the within class variance or equivalently maximizing the

between class variance." *The ITK Software Guide*.
 (<http://www.itk.org/ItkSoftwareGuide.pdf>)

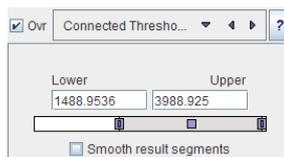


With the mouse CT, Otsu with the settings above finds appropriate thresholds for the skeleton object and the soft tissue, as illustrated below.

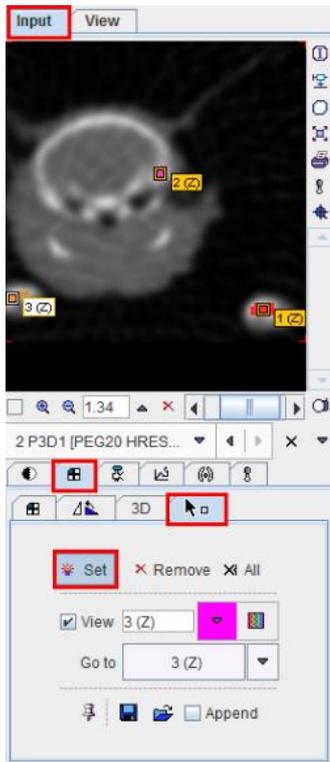


Connected Threshold (ITK)

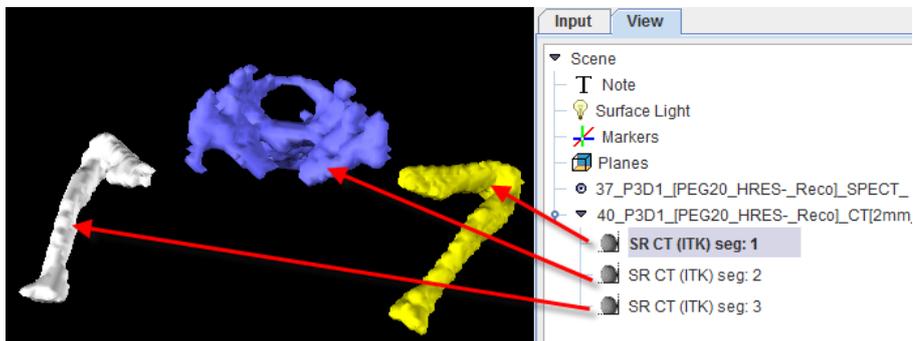
Connected Threshold is a region growing method with the criterion that (similar to the **IN RANGE** method) the included pixels must have values between a **Lower** and an **Upper** threshold.



As usual with region growing methods, the user has to specify a seed point by clicking into the image. The ITK region growing methods are particular in that multiple seed points can be specified at once by the use of markers as illustrated below.



Once the markers tab has been activated and the **Set** button enabled, a marker is created as a seed point for each clicking into the image. When the segmentation is started, the region growing with the specified value range is performed from each marker.

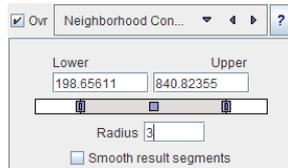


Neighborhood Connected (ITK)

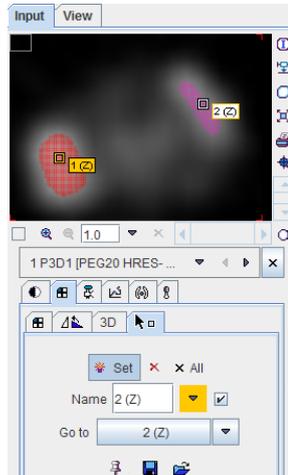
Neighborhood Connected is a region growing method with two criteria:

- 1) The included pixels must have values between a **Lower** and **Upper** threshold.
- 2) All pixels in the neighborhood (within a specified pixel radius) must also be within the value range.

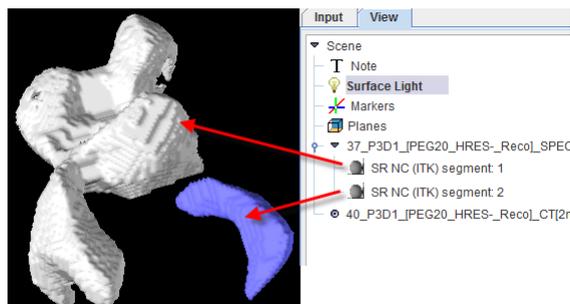
By the combination of these criteria small structures are less likely to be accepted in the region.



Multiple seed points can be specified by the use of markers.



and correspondingly multiple structures can be found at once.



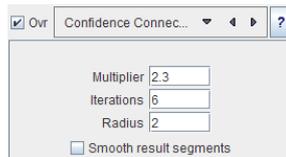
Confidence Connected (ITK)

"The criterion used by the **Confidence Connected** method is based on simple statistics of the current region. First, the algorithm computes the mean and standard deviation of intensity values for all the pixels currently included in the region. A user-provided factor is used to multiply the standard deviation and define a range around the mean. Neighbor pixels whose intensity values fall inside the range are accepted and included in the region. When no more neighbor pixels are found that satisfy the criterion, the algorithm is considered to have finished its first iteration. At that point, the mean and standard deviation of the intensity levels are recomputed using all the pixels currently included in the region. This mean and standard deviation defines a new intensity range that is used to visit current region neighbors and evaluate whether their intensity falls inside the range. This iterative process is repeated until no more pixels are added or the maximum number of iterations is reached."

The number of iterations is specified based on the homogeneity of the intensities of the anatomical structure to be segmented. Highly homogeneous regions may only require a couple of iterations. Regions with ramp effects, like MRI images with inhomogeneous fields, may require more iterations. In practice, it seems to be more important to carefully select the multiplier factor than the number of iterations. However, keep in mind that there is no reason to assume that this algorithm should converge to a stable region. It is possible that by letting the algorithm run for more iterations the region will end up engulfing the entire image.

The initialization of the algorithm requires the user to provide a seed point. It is convenient to select this point to be placed in a typical region of the anatomical structure to be segmented. A small neighborhood around the seed point will be used to compute the initial mean and standard deviation for the inclusion criterion."

The ITK Software Guide. (<http://www.itk.org/ItkSoftwareGuide.pdf>)



Volume-of-Interest (VOI) Restriction

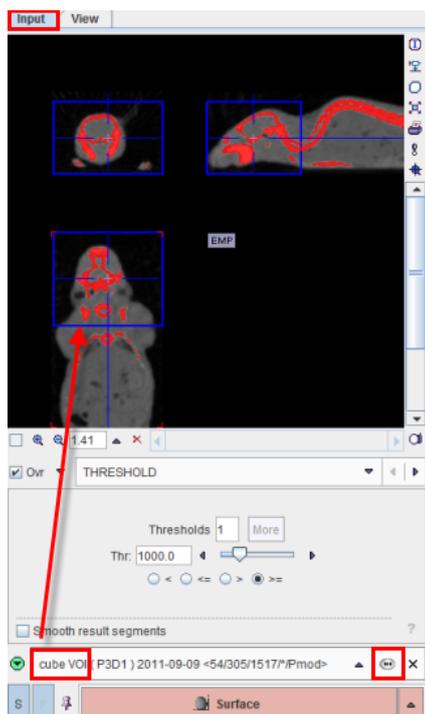
The segmentation methods alone may not be sufficient to separate an object from other structures. If this happens, the user can define a VOI which prevents segmentation from leaving the area of main interest.

If such a VOI already exists, it can be selected using the button

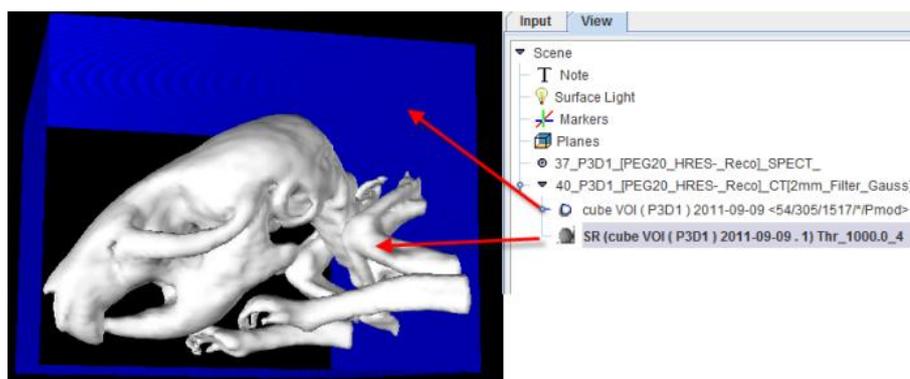


Otherwise, open the VOI tool using the  button to the right of the **Input** image, draw a VOI, save it, quit the VOI tool, and select the VOI. Note that the VOIs are also rendered and will be available for display in the object tree.

The example below shows how a box object VOI was used which only encloses the head of the mouse.

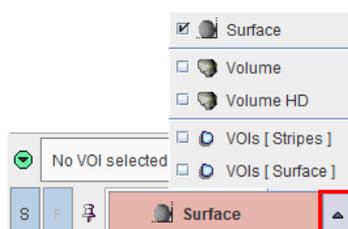


After **Surface Rendering**, the head is shown together with the VOI which is also represented as an object and can be hidden.



Object Rendering

After the segmentation procedure has been defined, the 3D objects can be generated. P3D allows the rendering of several studies as well as dynamic series simultaneously. Therefore there are a few user interface elements in the lowest row on the **Input** tab



which determine the rendering result.

- S** This toggle button is important if several input series were loaded. If it is enabled, only the *currently selected* input study is processed, otherwise all loaded studies are processed with their respective definitions.
- F** This toggle button is applicable if a dynamic series was loaded. If it is enabled, only the currently shown *frame* is processed, otherwise all of the dynamic frames. The purpose of dynamic segmentation is to construct a series of objects which can be animated over time.
- Surface Rend** This configuration button defines whether the data resulting from segmentation are subject to Surface Rendering or Volume Rendering. With the **Surface Rend** setting (default), a SR object will be generated.
- Volume Rend** With this configuration setting a Volume Rendering is performed.
-
- Note:** VR is only performed on the current input series, not on several series concurrently.
-
- Volume Rend HD** With this configuration a High Definition Volume Rendering is performed.
-
- Note:** Volume rendering is interpolated to the homogeneous smallest pixel size. Any other VR object available in the **View** tree becomes an HD VR once a VR HD object is appended to the scene.
-
- VOIs [Stripes]** With this configuration setting the selected set of VOIs is rendered as a set of stripes with a width corresponding to the slice thickness. Note that VOIs can also be independently rendered without loading image data.
- VOIs [Surface]** With this configuration setting the selected set of VOIs is interpreted as a volume and surface rendered.
-  Mode of object creation. If the pushpin is *fixed* () the created objects are **appended** to the current list of objects, otherwise the scene is cleared and populated with the new objects only.

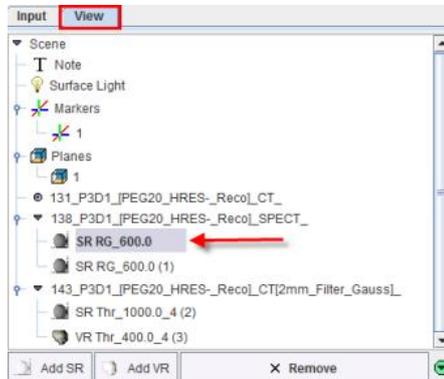
After the segmentation has been configured, select the rendering button, eg **Surface Rend**, to start the segmentation and the subsequent rendering. Note that if the rendering method is changed, segmentation automatically starts.

We recommend incrementally rendering study by study (S) in the append  mode.

Visualization Options

Object Management and Adjustments of Properties

All segmentation operations result in VROs (Virtual Reality Objects) which are arranged as an object tree in the **View** tab as illustrated in the example below.

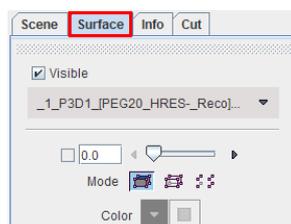


Entries with **SR** represent surface renderings, **VR** volume renderings, the **Surface Light** is the optional lightning source for SR objects, the **Markers** are synthetic objects usable for indicating points of interest, the **Planes** are sets of image slices, and **Note** is a text shown in a corner of the 3D scene. If an object is not needed any more, it can be removed from the tree by the **Remove** button.

The **Add SR** and **Add VR** buttons are used to create *ghost* objects which are empty SR or VR objects with defined visualization attributes. These attributes will be used for rendering the objects resulting from the next segmentation. The advantage of using ghost objects is avoiding lengthy rendering operations with inadequate default settings.

To delete a 3D object available in the tree initially select the corresponding object and then the **Remove** button. To **Collapse all Scene nodes** in the **View** tree activate the green button  available in the lower right corner. Immediately the color button switch to blue. The details of the **View** tree can be restore any time activating the blue button  **Expand all Scene nodes**.

To modify the appearance of a 3D object, the corresponding object must first be selected in the tree. Its properties are then shown in the lower tab named **Surface** or **Volume** etc. according to the object type. If a sub-tree or several objects in the tree are selected, the common properties may be manipulated for all contained objects at once.



The meaning and use of the properties are explained in the *Viewing Options* sections below for the different types of objects. There are also sections explaining the function of the other tabs **Scene**, **Info** and **Cut**.

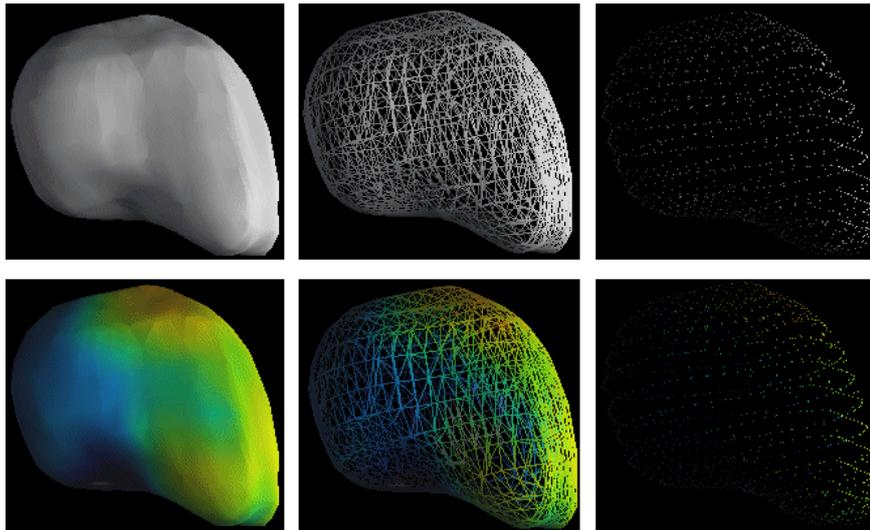
Viewing Options for Surface Rendering (SR) Objects

A SR object only consists of a surface. Its properties define how the surface is represented in the scene.



The **Visible** box determines whether or not the object is shown. The texture selection allows to switch off texturing (**No Texture**), or to select one of the loaded image series for coloring the surface. If no texture is active, the surface color can be defined using the **Color** selection.

There are three **Modes** how to render a surface: **solid**, as a **wire** frame, and by **points**. The wire frame and the points modes have the advantage that inner objects are visible. The examples below illustrate the different modes without and with (lower row) texturing.



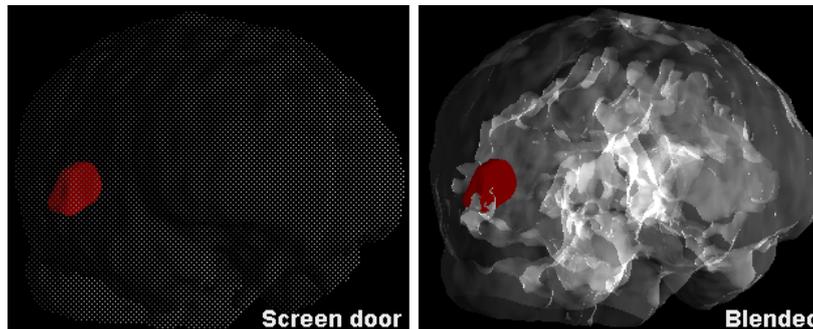
Transparency

To also allow viewing into **solid** objects it has a transparency property. High transparency settings can be used to make enclosed objects partly visible. Currently there are two ways how transparency is implemented.

- 1) The standard way (**Screen door**, box not checked) is to punch holes into the surface. The higher the transparency, the bigger the holes.

- 2) The other way (**Blended**, box checked), which provides a smoother view, *is only available as long as no VR objects are generated*, and if Java3D 1.4 or above is in use.

The check box next to the transparency slider allows switching the two modes.

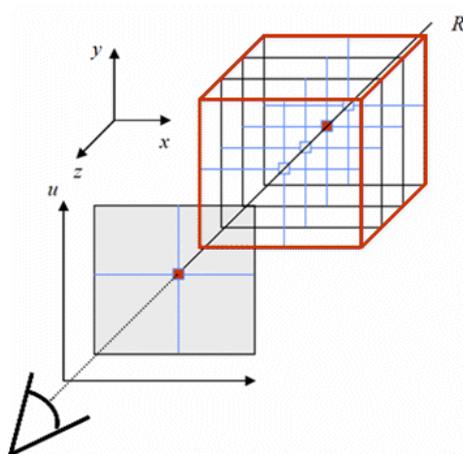


Note: Property changes of SR objects are immediately reflected in the scene.

Viewing Options for Volume Rendering (VR) Objects

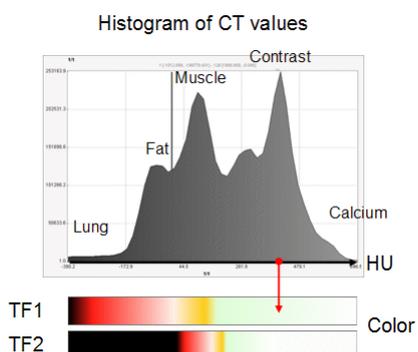
VR Principle

VR objects are computed from all the data, not just from a derived surface. The method is based on ray tracing from a certain viewpoint. Rays are cast through the image volume, and the *ray values* recorded when they pass a plane behind the object, thereby producing an image.

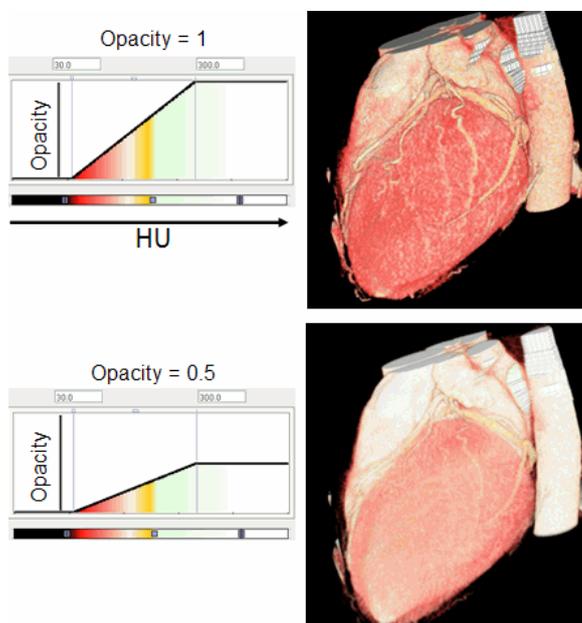


Each voxel is regarded as a "particle" that emits a certain color and absorbs a fraction of the passing light from the other voxels along the same ray.

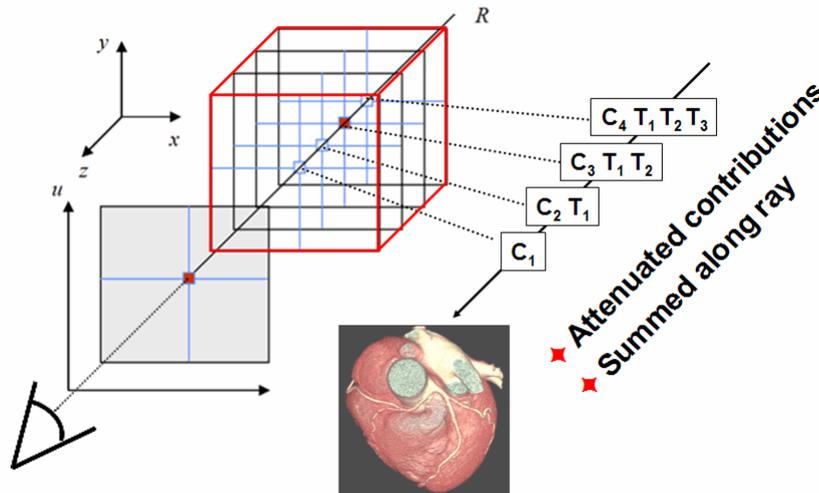
The *color transfer function* (TF) describes the voxel color as a function of the voxel value. For CT data, for example, the voxel values range from -1000 to 3000, and color mappings as illustrated below are often used.



The *opacity function* describes the voxel opacity as a function of the voxel value. Opacity values range from 0 to 1, whereby 0 means full transparency, and 1 full opacity. Often, ramp functions are applied as illustrated below. If lower opacity values are used as in the lower example, information from the inner of the object may become apparent such as the contrast agent in the ventricular cavity.

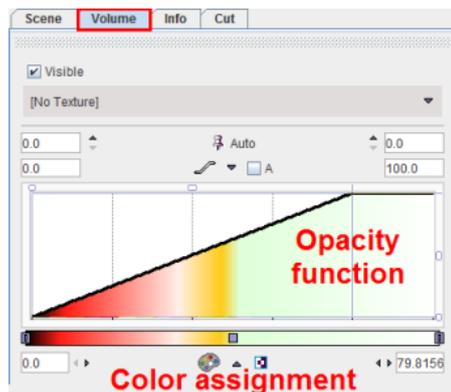


The ray value (the VR image pixel color) is calculated by summing up the contributions along a ray as illustrated below. C_i denotes the color of a voxel i , and T_i the transparency (1-opacity) of a voxel. Contributions of voxels far from the surface will be attenuated by multiple intervening voxels and thus be faint, depending on the opacity function.



VR Implementation in P3D

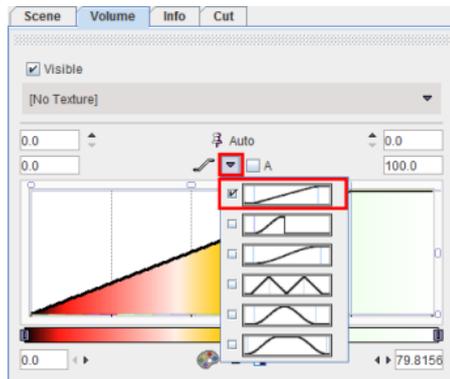
The properties tab **Volume** of VR objects shows several elements.



The **Visible** box serves for hiding the VR object altogether. The **Texture** selection allows combining volume rendering with color texturing from a matched study. The lower part of the tab contains the elements for defining the color and opacity transfer functions. The **Auto** refresh button serves for enabling/disabling automatic updating the display after any settings has been changed.

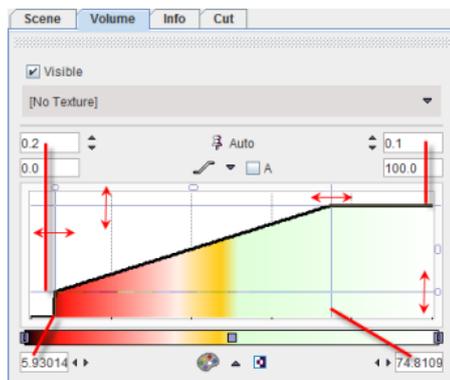
Opacity Functions

There are different shapes of opacity functions which can be selected using the arrow button highlighted below.



The purpose of the different shapes is to implement a variety of weighting functions, so that certain intensity ranges can be emphasized or suppressed. Usually, the ramp function is applied.

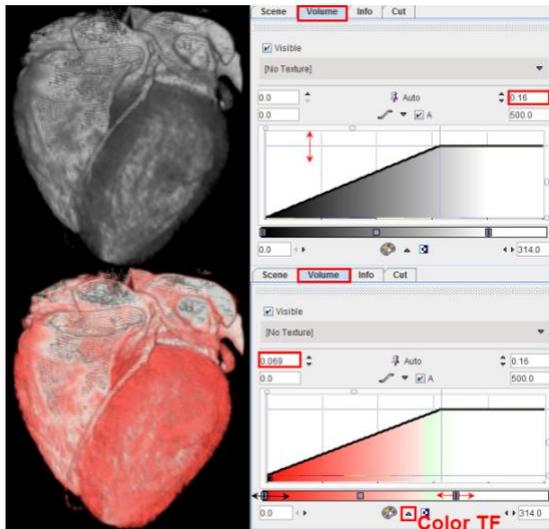
Once an opacity function has been selected, it can be manipulated in several ways as illustrated below:



- 1) The opacity value is in the vertical direction. The minimal and the maximal values of the opacity function can be set numerically or by dragging the horizontal handle lines. Example: 0.2 (= minimal opacity), 0.1 (= transparency at highest opacity)
- 2) The pixel value is in the horizontal direction. The min/max range of the transfer function can be specified as absolute numbers (if the **A** box is checked), or in relative percent values (if the **A** box is not checked as in the example). This is only for the convenience of the transfer function adjustments and has no impact on the rendering. The range can be specified by entering numbers directly (example: 5.9, 74.8), or by dragging the vertical handle lines.
- 3) Sometimes the transfer function should be defined in a small sub-range of the entire value range. In this case the x-range can be zoomed by changing the limits of the display range. The example above displays the whole range from 0% to 100%.

Color Transfer Functions

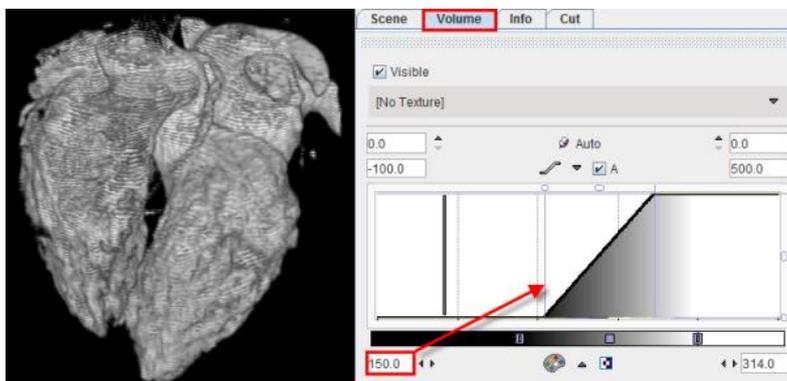
There are different color transfer functions available as illustrated in the example below.



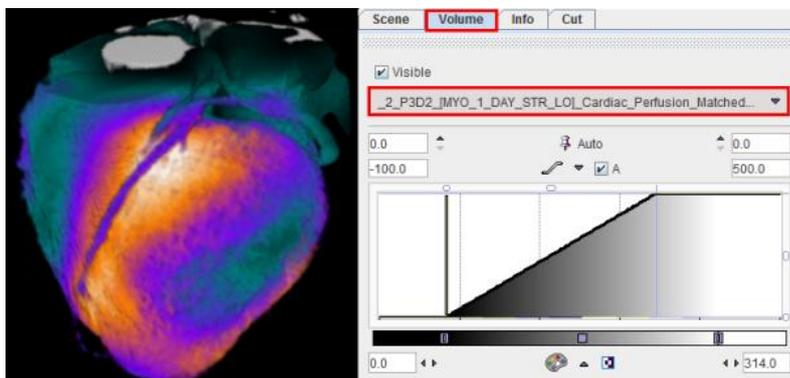
The upper rendering of the angio CT uses a gray color TF, and the lower rendering the **Heart VR** color TF. Note the indicated down arrow button which allows selecting among the available color TF. After a color TF has been chosen, the mapping of voxel values to colors can be adjusted by moving the upper/lower color TF limits as indicated by the horizontal arrows above. The **Heart VR** table makes lower values look fleshy, and higher values white.

Note: With big data sets refreshing of scenes with VR objects may be quite time-consuming. In this case it is recommended to switch the **Auto** refreshing to off. As soon as the VR properties are changed, the  alert icon appears in the taskbar to the right. Once all VR properties have been adjusted, the user can trigger scene refresh by clicking at .

Increasing the lower value of the opacity function allows cutting pixels with the density of myocardium, and the result only shows the contrast filled ventricles and the coronary arteries.



Volume rendering can also be combined with textures to introduce functional information into an anatomical rendering. The example below shows such a 3D fusion rendering of an angio CT combined with the texture color from a matched SPECT perfusion scan. The opacity function is linear and with the same range as above.



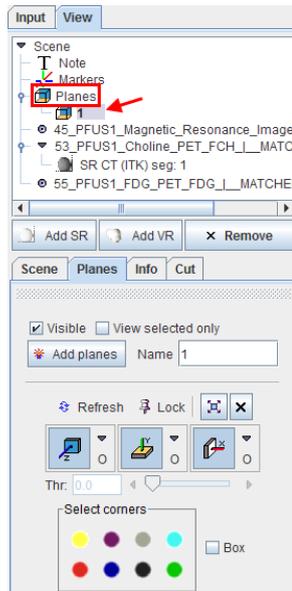
Notes:

1. The 3D fusion result also depends on the color and window level settings of the texture study which can be changed on the **Input** tab.
 2. Voxels in the segment which have values below the lower or above the upper thresholds of the texture color table obtain the color of the selected color TF (Gray scale in the example above).
-

Image Plane Objects

Additional elements for composing a meaningful scene are the slice images from the different image series. Planes also serve for defining the octants to be cut out of VR or SR objects.

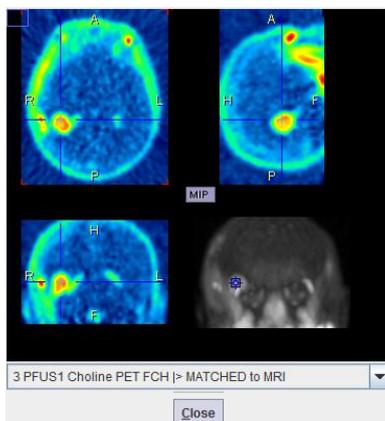
Plane objects can be created by selecting **Planes** in the object tree, and then on the appearing **Planes** tab activating the **Add planes** button.



A new planes object appears in the tree, per default named 1. This name can be edited in the **Name** field. Initially, all three orthogonal planes are displayed, but they can individually be switched on/off using the **Z**, **Y** and **X** toggle buttons. The **x** button switches off all three planes. The planes show the slice images of the study selected on the **Input** tab. There the image coloring can be adjusted.

Navigator Window

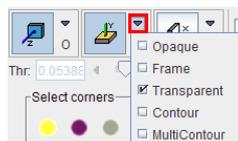
The position of the planes can best be changed using the planes navigator, which is started using the  button. The appearing window shows three orthogonal slice images and a list selection to choose among the loaded series.



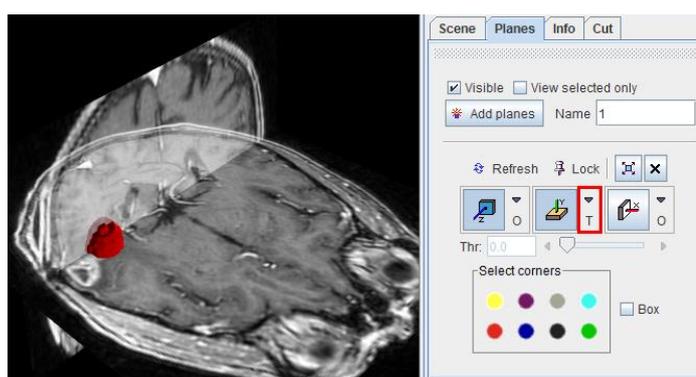
The triangulation point can be changed by simply clicking into the navigator images. As an alternative, the slice locations can be changed by selecting other slices in the **Input** series. In case the plane locations of the **Planes** object should no more be changed, select the **Lock** pushpin button.

Planes Styles

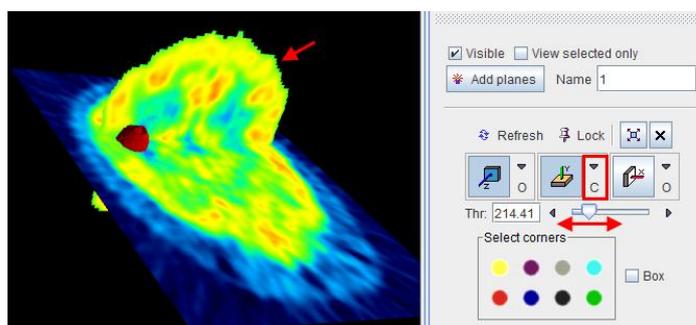
The planes can be rendered in different styles which can be selected from the list to the right of the plane buttons.



The axial slice in the example below is **Opaque**, while the coronal slice is **Transparent**. Transparent slices do not obstruct the view, but this style is *not applicable if a VR object has been rendered*.



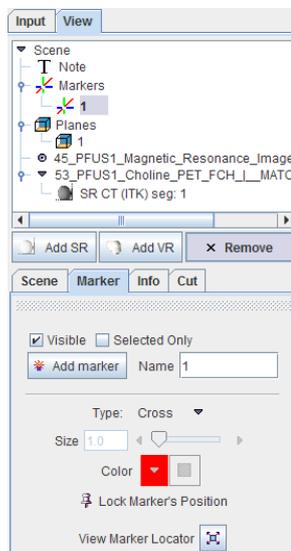
The **Contour** style suppresses the image information below a threshold which can be defined with the **Thr** slider. Only the biggest connected area will be shown. In contrast, **MultiContour**, shows all area above the threshold.



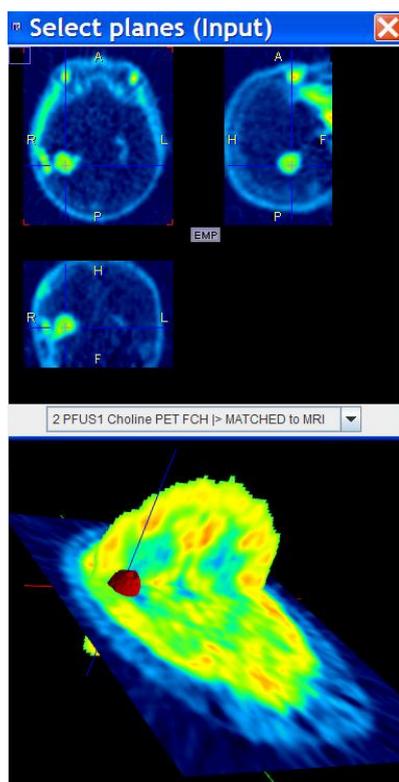
If multiple plane sets have been created, **Selected only** allows showing only the planes belonging to the object currently selected in the tree.

Marker Objects

Marker objects can be helpful for labeling points of interest in the scene. A marker object can be created by selecting **Markers** in the object tree, and then on the appearing **Marker** tab activating the **Add marker** button.



The marker is initially positioned at the plane intersection point of the **Input** series. Its position can be changed by changing the triangulation point there, or by opening the navigator window with  and clicking to the point of interest. Once its final location has been found, activate **Lock Marker's Position** to prevent unintentionally moving the marker.



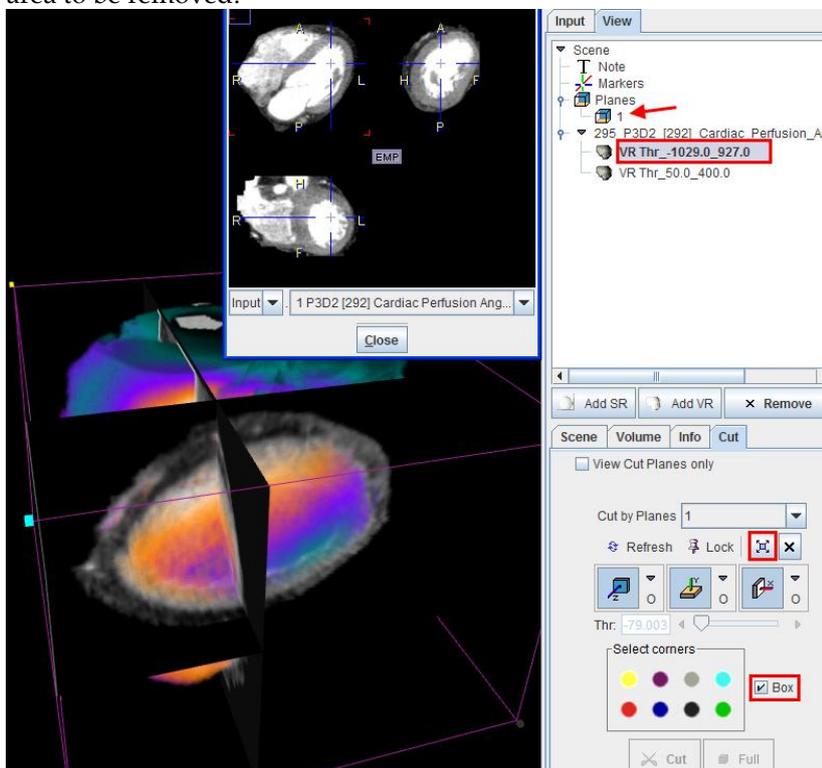
The marker can have the shape of a **Cross** as illustrated above, or a **Sphere**. If multiple markers have been created, **Selected only** allows showing only the marker belonging to the object currently selected in the tree.

Cutting away Parts of the VR or SR Information

Sometimes it may be helpful to cut parts of the scene. In P3D it is possible to remove parts from VR or SR objects, limited by orthogonal planes.

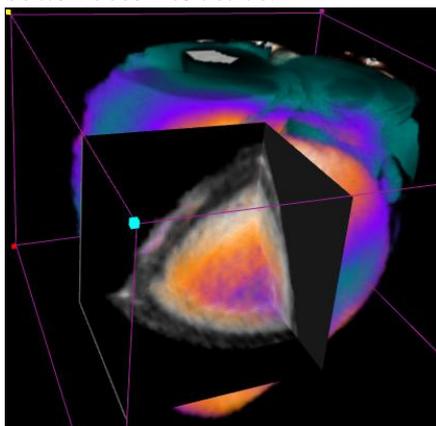
Cutting Procedure

- 1) Make sure a **Planes** object is available. If not, then create one.
- 2) Select the VR or SR object(s) in the tree which should be cut. Multiple objects can be cut at the same time.
- 3) Select the **Cut** tab belonging to this object (NOT the Cut tab of the Planes object).
- 4) Enable the image planes display with the  button.
- 5) Position the plane intersection point such that the three orthogonal planes enclose the area to be removed.

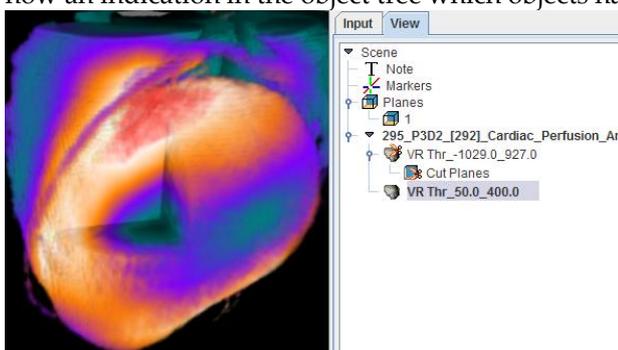


- 6) Enable **Box** to see the wire-frame box with the colored corners. Define the part to be removed by selecting color circles in the **Select corners** area. The color circles represent octants identified by the colored bullets in the corners of the wire box. After selecting the button(s), only the plane parts enclosing the parts to be removed are shown, and the **Cut**

button becomes active.

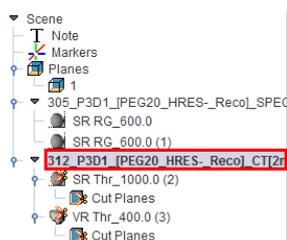


- 7) The **Cut** button starts a process with clears all information of the selected object(s) in the defined area and refreshes the rendering. Use the x button to remove the planes. There is now an indication in the object tree which objects have been cut.



- 8) The **Full** button undoes cutting and brings back the rendering of the full volume.

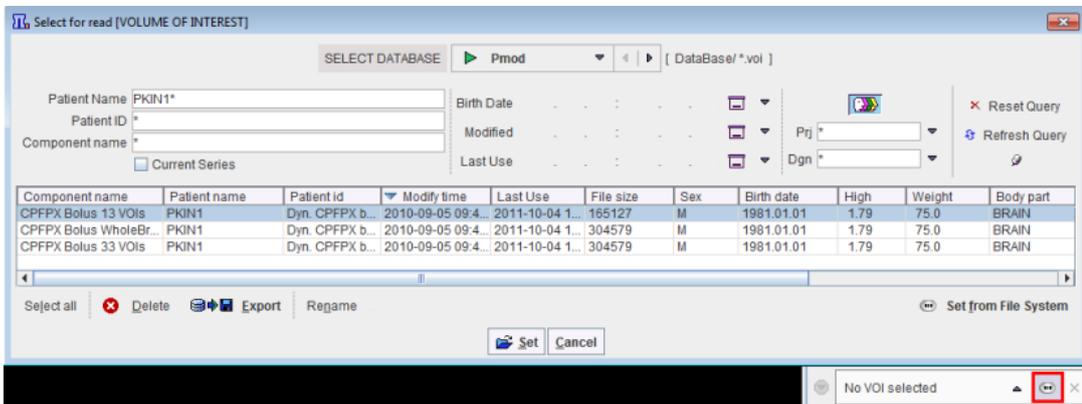
To apply the cutting procedure to all objects belonging to an image series the root object can be selected in the object tree as illustrated below.



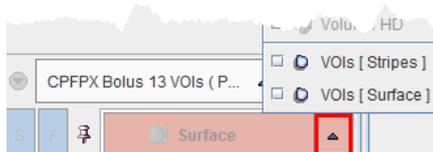
Rendering of VOIs

Volumes of Interest

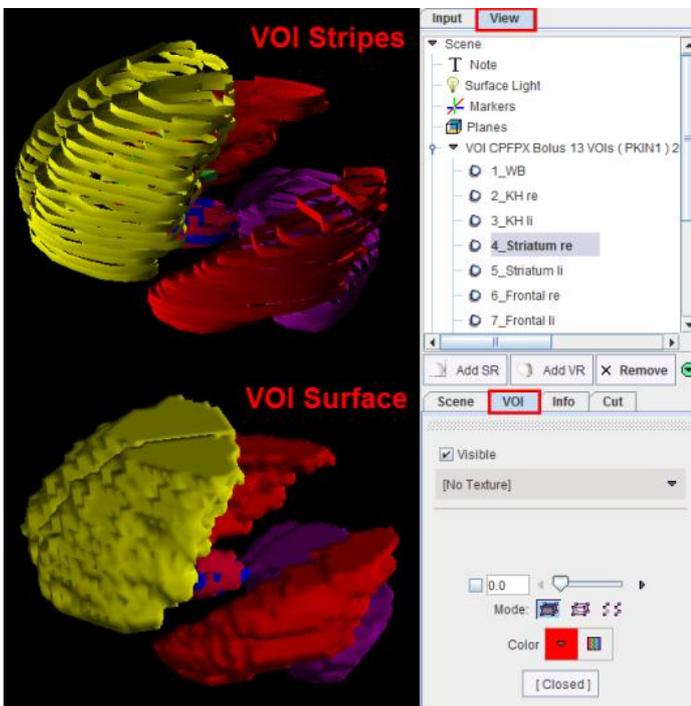
Volumes-of-interest can also be rendered by P3D. One application is the visualization of VOIs independent of an image study. To this end the VOI file is selected with the button indicated below on the **Input** tab.



For rendering VOIs, there are two choices available, VOIs [Stripes] and VOIs [Surface].



The results of both rendering types are illustrated below. Note that each VOI results in an object which can be selected in the tree and modified.



In case a VOI was used for restricting an image segmentation, it is rendered together with the segment and will also be available in the object tree.

Note: When the P3D option has been purchased, the 3D rendering of the VOIs can also directly be initiated from the VOI definition tool.

Scene Operations

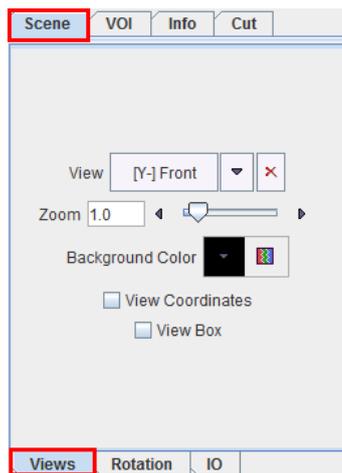
The main advantage of 3D renderings is that a user can interactively manipulate the scene to see it from different viewpoints. The scene can be explored in a mouse-operated way:

- ▶▶ To rotate the scene, click the left mouse button into the scene and drag the mouse.
- ▶▶ To zoom the scene, click the middle mouse button into the scene and drag the mouse.
Note that this mouse-operated zooming changes the rotation center. In order to keep the rotation center in the center of the object, the zoom slicer on the **Scene** pane should be used.

To shift the scene out of the center, for instance to examine a zoomed lesion, click the right mouse button into the scene and drag. To have a full view of the scene use the  button in the taskbar for hiding the controls to the right. Activating the same button again brings the controls back.

There are three sub-panes on the **View** pane which group different types of operation.

Scene Views

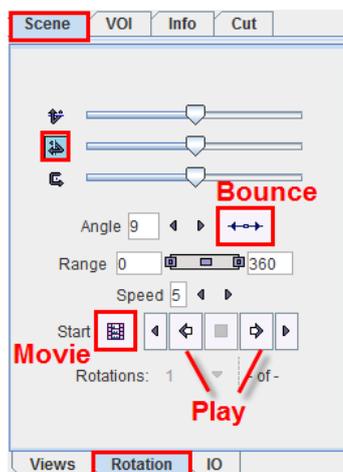


View	The list selection elements bring the viewpoint into a well-defined position, eg. frontal, from left, etc. The X button to its right resets scene positioning to the default.
Zoom	Zoom factor for scene rendering.
Background Color	List selection for changing the background color.
View	Show/Hide the box indicating the anatomical directions

Coordinates	
View Box	Show the wire-frame box.

Scene Rotation and Cines

The **Rotation pane** contains multiple elements for creating cines or movies.



The sliders allow well-controlled rotations of the scene.

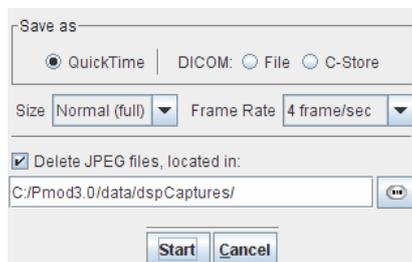
A cine is configured by the following properties:

- 1) the initial rotation angles (sliders),
- 2) the rotation axis (icons left to sliders),
- 3) the **Angle** increment in degrees (smaller increments produce smoother cines),
- 4) the rotation **Range** in degrees,
- 5) the behavior at the end of the rotation range (bouncing),
- 6) the **Speed**.

The cine can then be started with one of the play buttons, eg . Note that the rotation axis may be changed while the cine is playing, and scene manipulations by the mouse are still active.

Creating Movies

To record a cine as a movie first configure the cine loop appropriately and test it. Then enable the movie button  and start. A dialog appears

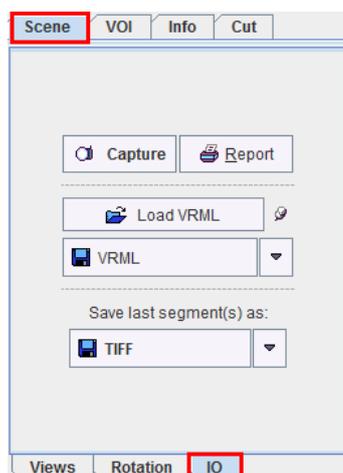


which allows defining the movie format (QuickTime, or DICOM), quality settings and the destination path.

After confirming with **Start**, the scene is stepped through the angles until the defined number of **Rotations** are covered, creating a jpeg file at each angle. Finally, the jpeg images are compiled into the movie, and the jpeg files (optionally) deleted.

Scene IO

The **Scene IO** tab houses several possibilities for saving and loading data.



Capture	Create a screen capture of the current scene for documentation purposes.
Report	Create a screen capture of the current scene and put it into a report page.
Load VRML	Load a scene description saved in VRML (Virtual Reality Modeling Language) format. Note that the import may fail with files from third party programs. The pushpin enables the append mode for adding the loaded objects to the current scene. Otherwise the scene is first cleared.
 ...	Save the SR objects of the current P3D scene. Three output modes are supported: STL (stereo lithography format), VMRL , and VMRL (sensors) .
Save last	Allows saving the result of the last segmentation operation as a binary

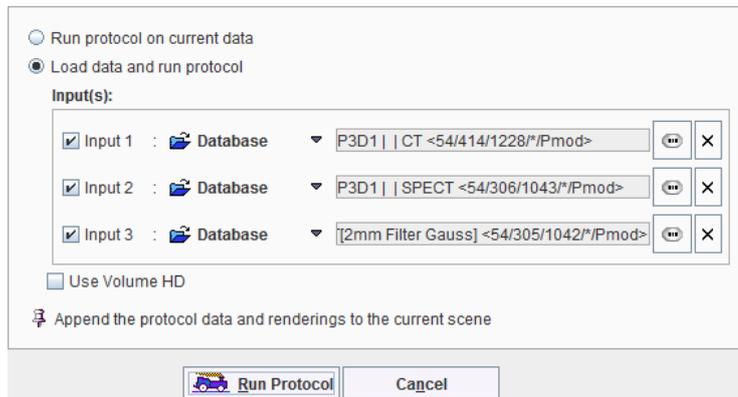
segments	image in the selected format.
----------	-------------------------------

Protocol Files

Saving a scene in VRML is limited to SR objects, and may create huge files. A more flexible alternative is to save a description of the P3D rendering in a protocol file using the save button in the taskbar to the right.



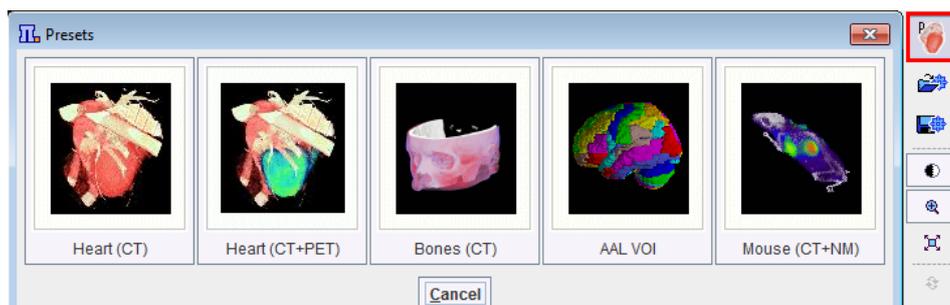
When such a P3D protocol is loaded again, a dialog window appears which indicates the data to be loaded.



If the user wants to apply the same rendering to similar data, he can replace the input files with the refresh icon. If he wants to apply the procedure to data already loaded, the **Run protocol on current data** can be enabled.

Normally the scene will be wiped out and replaced by the result of the protocol. To add to the current scene, the **Append the protocol data and renderings to the current scene** pushpin should be fixed. To create high definition VR objects the **Use Volume HD** option should be enabled.

The heart icon shows a dialog window containing standard renderings definitions.



Currently, the following predefined protocols are supported:

- 1) Angio CT renderings are offered with and without multi-modal fusion.
- 2) Bones (CT)
- 3) AAL VOI
- 4) Mouse (CT+NM)

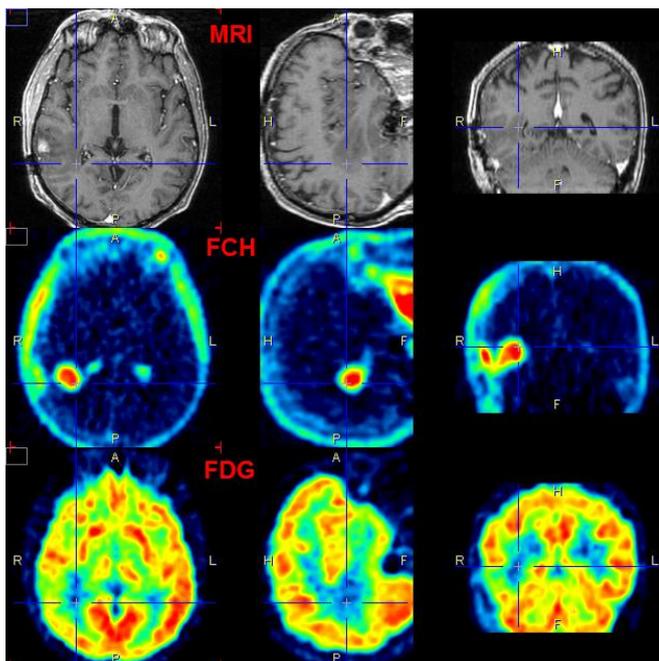
Note that when a predefined protocol is selected, appropriate input file have to be set using the  button. Alternatively, the predefined protocol can be applied to data already loaded enabling the **Run protocol on current data** radio button. Optionally, the **Use Volume HD** checkbox can be activated to create high definition VR objects.

3D Rendering Examples

The main visualization capabilities of P3D are illustrated by some examples.

Example 1: Surface Rendering of a Brain Tumor

This example can be reproduced with data of patient **PFUS1** available in the example PMOD database after installation.



These three matched studies are loaded as **Input** series. The aim is to localize and visualize the brain tumor. A simple yet useful approach is the following:

FCH The tumor is well delineated in the FCH data but needs a restriction.

On slice 29, click into the tumor and apply a **REGION GROWING** segmentation with criterion ≥ 1.7 .

Select the ellipsoid VOI **Tumor enclosure**.

Start rendering with **SR** (current study only).

Switch the VOI to invisible.

Select the red surface color in the **Surface** section of the tumor SR object.

FDG The FDG data is used to obtain the outline shape of the brain.

First smooth the FDG data with a 6mm Gaussian.

Activate the append mode by selecting the pushpin.

Apply a **THRESHOLD** segmentation to the smoothed FDG study with a value of 7.9.

Render as a SR, then set yellow color.

Set blended transparency using a value of 0.7.

MRI MRI slices can be included for a better anatomical understanding.

On the **Input** tab select the MRI study.

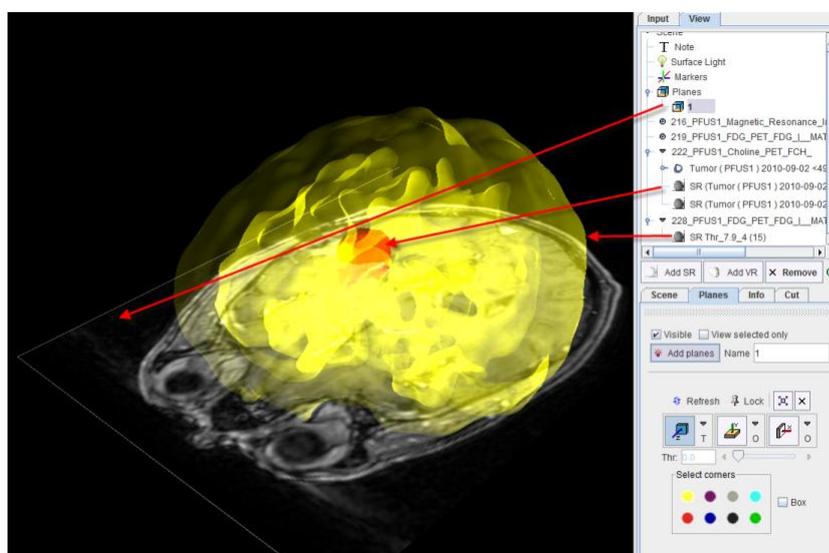
In the object tree select **Planes** and then activate **Add planes**. The three orthogonal MRI planes are shown.

Switch off the y and the x plane for display in the scene. Only axial MRI slice is shown.

Select **Transparent** mode (selection right to z-plane button)

Define the location of the plane using the navigator window, pointing at the location of interest. The scene gets updated accordingly.

The result from a specific viewpoint is shown below.

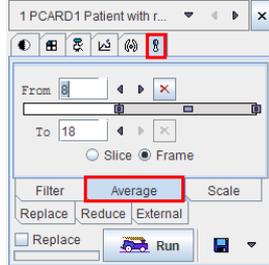


Example 2: Animated Texture on Cardiac Surface

This example illustrates the capability of P3D to project a dynamic texture onto a static surface using a dynamic NH_3 cardiac PET study. It can be reproduced with data of patient **PCARD1** available in the example PMOD database after installation.

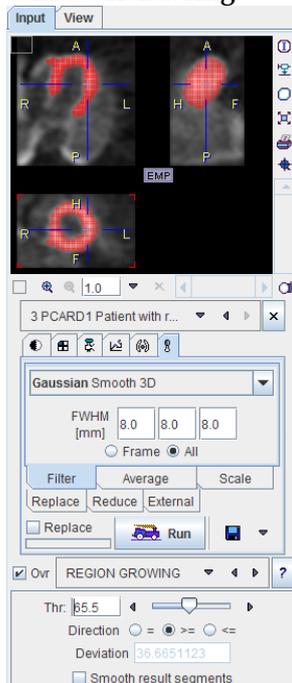
Anatomy Load the dynamic **NH₃, Stress** study of patient **PCARD1** into the **Input** area.

With the external tools, average the frames 8 to 18.



Smooth the generated average image in a similar way by applying an 8mm **Gaussian Smooth 3D** on the **Filter** tab.

Define **REGION GROWING** segmentation with threshold 65.5 for the smoothed, averaged data set. Click into the myocardium and start the **Surface Rendering**.

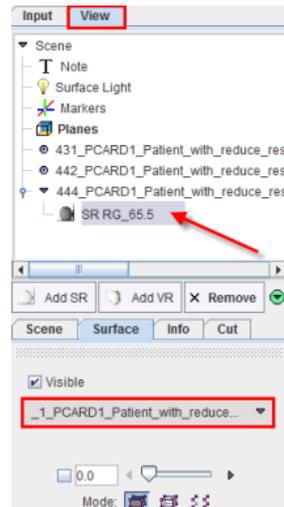


A smooth shape of the myocardium is shown in the scene, represented by the **SR RG_65.5** object in the tree.

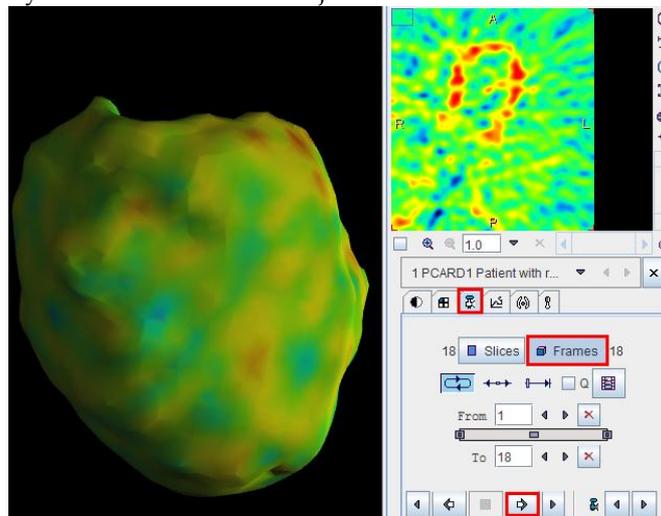
**Cine of
Dynamic
Uptake**

Load the same dynamic **NH₃, Stress** study of patient **PCARD1** into the **Texture** area.

Select the **SR RG_65.5** object in the tree, and switch the texturing to the dynamic series.



On the **Input** pane, configure the cine player to show a movie in time (**Frames** mode), then start. The uptake over time is now shown as a dynamic texture on the object surface.



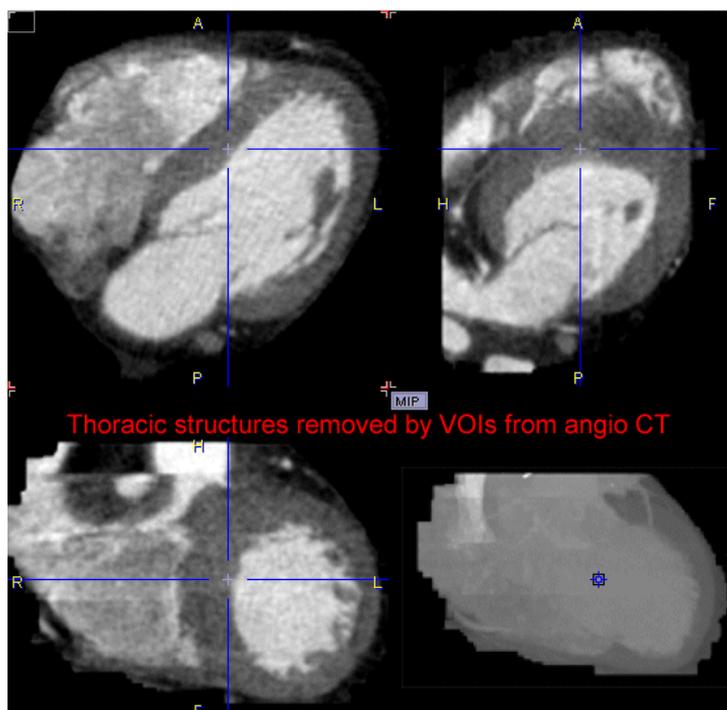
Note: When a dynamic study is segmented and the **F** button (single frame) is not selected, one object per time will be created. In this case a cine of the **Input** images can be run to show an animation of the object shape over time. An obvious application for this feature is the rendering of a beating heart from a gated acquisition.

Example 3: Fused Volume Rendering for Angio CT

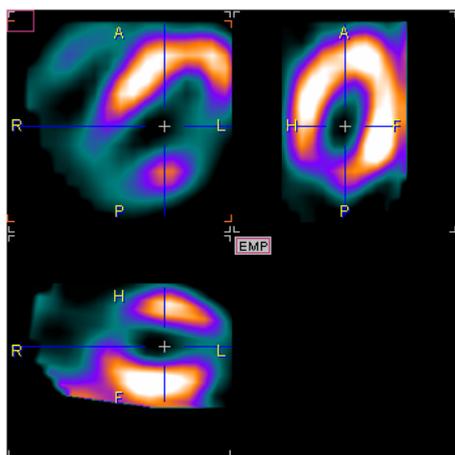
This example illustrates the use of the **Predefined** protocol for an angio-CT/SPECT fusion. It can be reproduced with data of patient **P3D2** available in the example PMOD database after installation.

First the CT data was prepared. A cubic sub-volume containing the heart is extracted from the data using the **Resize** function available in the external tools. The remaining thoracic

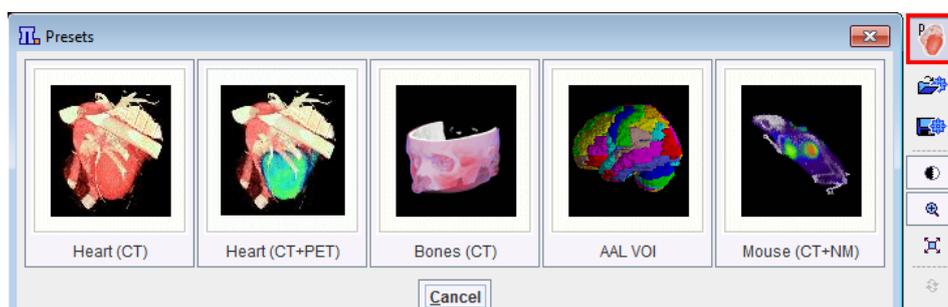
structures are interactively removed by drawing VOIs and setting the values outside the VOI to -1024.



Next the SPECT data was matched to the CT in the PFUS tool, and the resliced images saved.

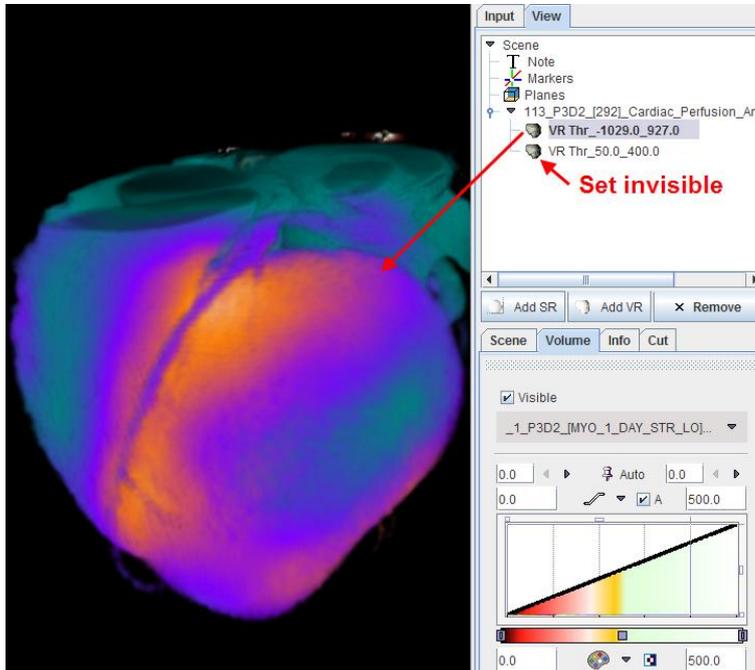


In P3D, load the prepared CT study **Angio CT Reduced** series of patient **P3D2** and the matched SPECT. Then, select **Heart (CT+PET)** from the **Predefined** protocols. Select **Run protocol on current data** and then **Run Protocol** button.

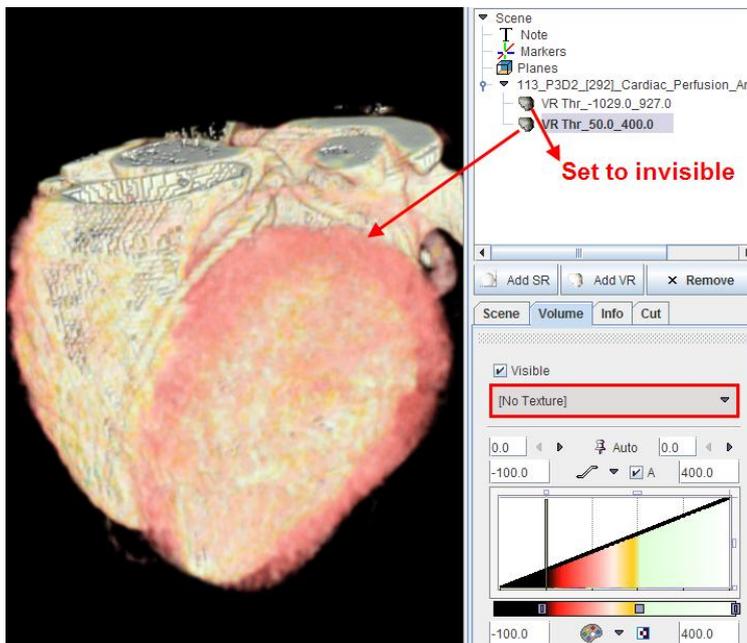


The protocol performs a segmentation with two **RANGE** thresholds (-1029-927 HU, 50-400 HU), and performs a VR rendering for each of the resulting segments.

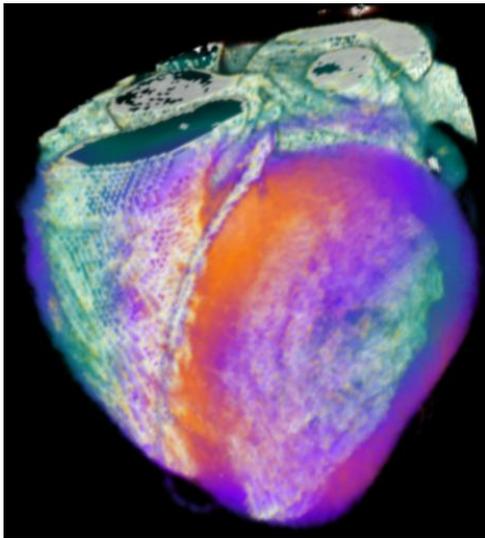
The first segment is textured using the SPECT colors, and results in a rendering as illustrated below. To see this segment only remove the **Visible** check of the second and VR object activate **Refresh**. After adjusting the SPECT colors a bit the result looks like illustrated below



Note that the coronary arteries are colored because the SPECT study is not sharply bonded. Therefore, the second segment is used for highlighting the coronary artery structure.



When both VR renderings are combined in a single scene (checking both **Visible** boxes), the following result is finally obtained.

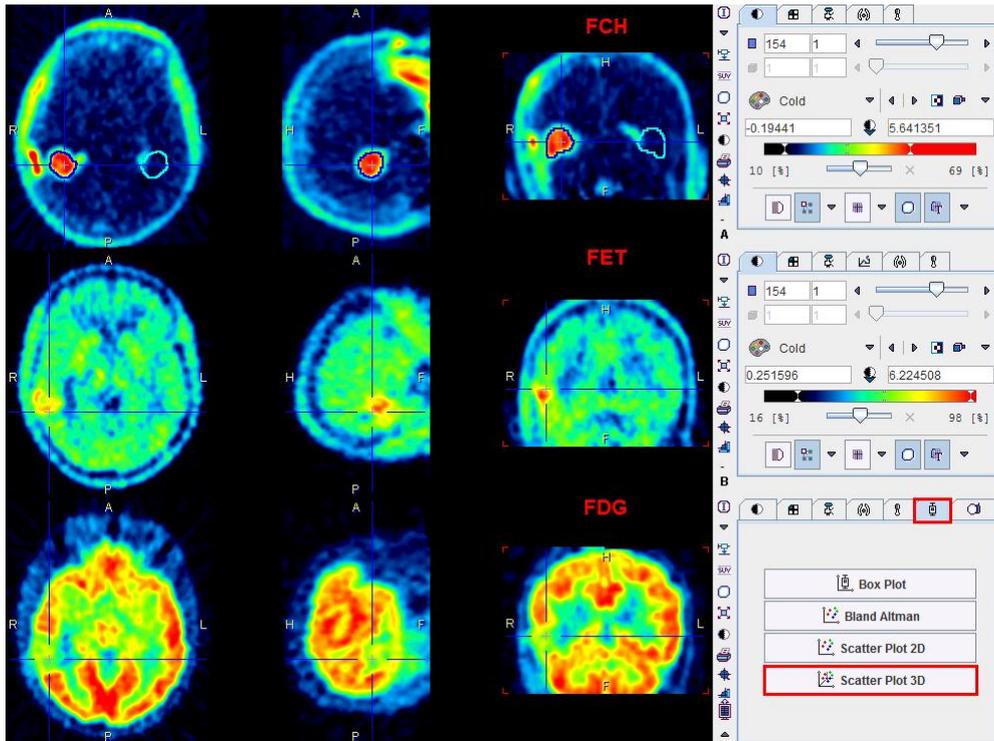


This scene could be extended by additional objects. For instance, if an important coronary

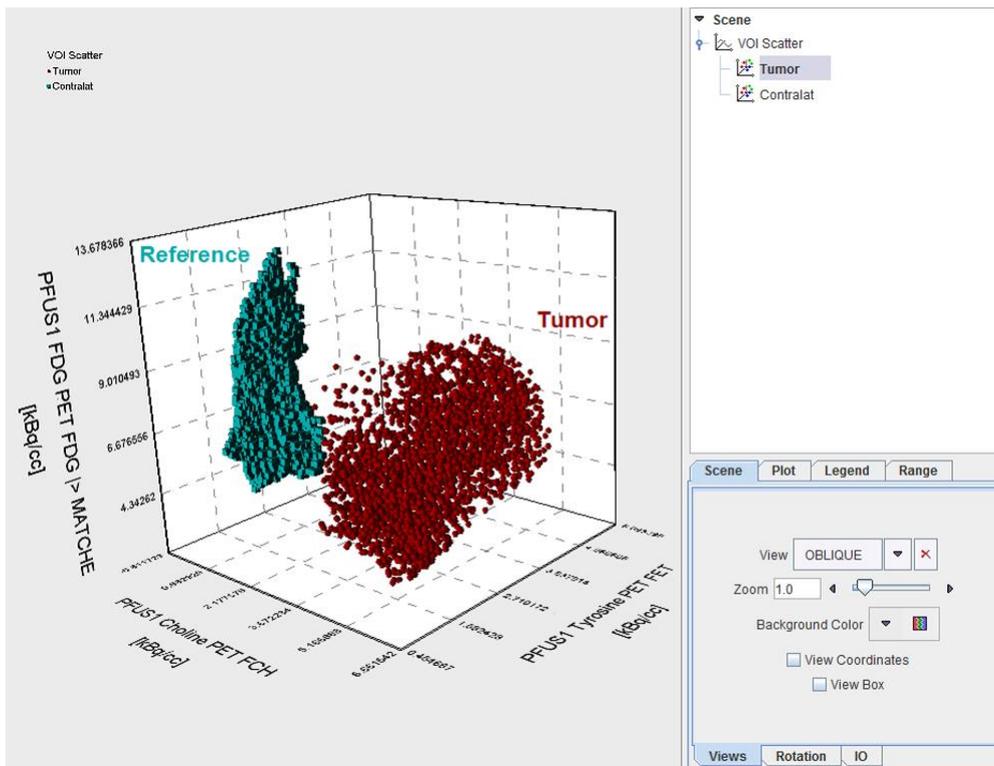
3D Scatter Plots (from PFUS)

artery could be individually segmented, it could be added as a clearly visible SR object.

The 3D tool has a dedicated application for showing a scatter plot of the pixel values belonging to VOIs of three matched studies in PFUS. The example below shows the matched images of a brain tumor patient studied with fluorocholine, fluoroethyl tyrosine, and FDG PET. Two VOIs have been outlined, tumor and gray matter.



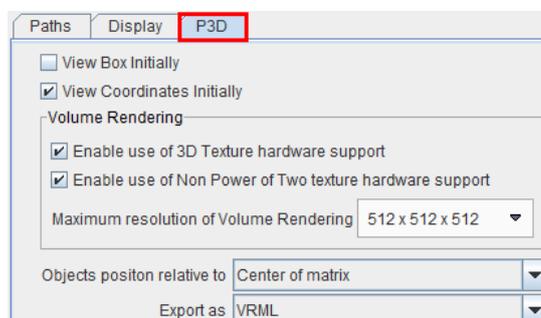
When **Scatter Plot 3D** is activated, the following rendering is generated. Each pixel is shown as a 3D symbol, with the VOI pixels grouped by the symbol color. In the 3D scatter mode there are some new P3D properties for adjusting the rendering as well as the annotations.



Note that the numeric scatter data can be saved as an ASCII text file for statistical analyses using the **Save** button on the **IO** sub-panel.

P3D Configuration

There are a few specific configurations for the P3D tool. They can be accessed by the **Menu** entry **Settings/Modify**, or directly using the  button.



The **View Box Initially** check is for enabling the wire frame box, and **View Coordinates Initially** the orientation cube at start-up.

Some graphics boards may also support volume rendering textures in hardware. To exploit this acceleration feature, check the **Enable use of 3D Texture hardware support** box, and/or the **Enable use of Non Power OF Two Texture hardware support** box. **Maximum resolution of Volume Rendering** is identified during the **Acceptance test**. However it can be decrease selecting a smaller resolution available on the list.

Objects generated in P3D are positioned relative to the center of the scene volume (i.e. the center of the image volume is aligned with the center of the scene, **Center of matrix**). When importing virtual reality (VRML) objects generated by other programs this behavior may not be adequate. It can be modified by setting **Objects position relative to** to **Begin of matrix**, **Center of matrix** or **Data origins**.

For **Exporting** the scene, two formats are available: **VRML** (Virtual Reality Modeling Language, with or without sensors), or **SLT** (Stereolithography), a format which is understood by many CAD programs.

Copyright © 1996-2011 PMOD Technologies Ltd.
All rights reserved.

The PMOD software contains proprietary information of PMOD Technologies Ltd; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

Due to continued product development the program may change and no longer exactly correspond to this document. The information and intellectual property contained herein is confidential between PMOD Technologies Ltd and the client and remains the exclusive property of PMOD Technologies Ltd. If you find any problems in the document, please

report them to us in writing. PMOD Technologies Ltd does not warrant that this document is error-free.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of PMOD Technologies Ltd.



PMOD Technologies Ltd

Sumatrastrasse 25

8006 Zürich

Switzerland

+41 (44) 350 46 00

support@pmod.com

<http://www.pmod.com>

Index

3

- 3D Rendering Examples • 35
- 3D Scatter Plots (from PFUS) • 41

C

- Confidence Connected (ITK) • 13
- Connected Threshold (ITK) • 11
- Cutting away Parts of the VR or SR Information • 28

E

- Example 1
 - Surface Rendering of a Brain Tumor • 35
- Example 2
 - Animated Texture on Cardiac Surface • 37
- Example 3
 - Fused Volume Rendering for Angio CT • 38

H

- HOTTEST PIXELS Segmentation • 10

I

- Image Data Loading • 7
- Image Plane Objects • 24
- IN RANGE Segmentation • 10

M

- Marker Objects • 27

N

- Neighborhood Connected (ITK) • 12

O

- Object Management and Adjustments of Properties • 17
- Object Rendering • 15
- Otsu Threshold (ITK) • 10

P

- P3D Configuration • 44
- P3D Data Processing • 5
- PMOD 3D Rendering Tool Introduction • 2
- Preparation Steps • 7
- Protocol Files • 34

R

- REGION GROWING Segmentation • 9
- Rendering of VOIs • 30

S

- Scene IO • 33
- Scene Operations • 31
- Scene Rotation and Cines • 32
- Scene Views • 31
- Segmentation Methods • 8

T

- THRESHOLD Segmentation • 10

V

- Viewing Options for Surface Rendering (SR) Objects • 18
- Viewing Options for Volume Rendering (VR) Objects • 19
- Visualization Options • 17
- Volume-of-Interest (VOI) Restriction • 14