One

Introduction

1.1 OsiriX

The goal of OsiriX is to deliver the best possible viewer for images produced by radiology equipment, such as MRI, CT, PET, PET–CT, SPECT-CT, Ultrasounds, etc. Its key features are: displaying, reviewing, interpreting and post-processing the images. OsiriX is an image processing software dedicated to DICOM images (*.dcm* extension).

OsiriX is an open-source project, distributed under the LGPL license. Several versions of OsiriX are available. If you plan to use OsiriX in a medical environment, you will require a certified version for primary diagnostic imaging. Using OsiriX as a primary diagnostic workstation also requires the use of certified monitors for medical imaging.

OsiriX is the result of more than 5 years of research and development in digital imaging. OsiriX fully supports the DICOM standard for an easy integration in your workflow environment. OsiriX offers advanced post-processing techniques in 2D and 3D. OsiriX offers a complete integration with any PACS, including the well known open-source project: dcm4chee[1]. OsiriX supports 64-bit computing and multithreading for the best performances on the most modern processors. By adopting OsiriX you have made the right choice: it meets normal requirements, it is simple to use, it has unlimited power and the ability to evolve!

1.1.1 About DICOM

DICOM is a standard for handling, storing, printing and transmitting information in medical imaging [7]. The DICOM standard has now become the uncontested standard for the exchange and management of biomedical images. OsiriX is fully compliant with the DICOM standard for image communication and image file formats. DICOM is regularly updated through the DICOM committee. The up-to-date version of this standard is available on the DICOM website[2].

1.1.2 History

The OsiriX project started in November 2003. The first version was developed by Antoine Rosset MD, a radiologist from Geneva, Switzerland. He received a grant from the Swiss National Fund[3] to spend one year in UCLA, Los Angeles, with Prof. Osman Ratib, to explore and learn about medical digital imaging.

At first, the goal of the OsiriX project was to simply write a small software program to convert DICOM files to a QuickTime movie file, in order to help a radiologist

[1] www.dcm4che.org

[2] dicom.nema.org

[3] www.snf.ch

friend, Luca Spadola MD, to create a teaching files database. He spent more than two weeks searching a Java library for DICOM management and image manipulation. Indeed he wanted to create a cross-platform software program for Windows, Mac OS and Linux. But at that time, Apple had just released Mac OS X 10.3, the first usable release of their UNIX-based OS. It became obvious that Mac OS X was the best choice to quickly create a robust and modern DICOM viewer:

One DICOM viewer to rule them all

This small project became the unique obsession of this young radiologist. A SourceForge account[4] was created in January 2004 to develop OsiriX as an LGPL software.

The first public version was released in April 2004, version 0.1a, on Antoine Rosset's personal homepage[5]. This first version was developed in less than 6 months: it offered a basic database and a simple DICOM viewer, without post-processing functions or measurement tools.

In October 2004, Antoine Rosset went back to the Geneva University Hospital in Switzerland, to continue his career as a radiologist, but his obsession with OsiriX remained strong.

The reference article about the OsiriX project was written in April 2004 and published in June 2004 in the Journal of Digital Imaging [10].

Joris Heuberger, a mathematician from Geneva, joined the project in March 2005 on a voluntary fellowship of 6 months in UCLA, Los Angeles. During this period, while working on plugins for OsiriX, he added the first *Fly Through* algorithm to the Surface Rendering and Volume Rendering viewers.

In June 2005, during Apple's Worldwide Developer Conference (WWDC) in San Francisco, OsiriX received two prestigious Apple Design Awards: *Best Use of Open Source* and *Best Mac OS X Scientific Computing Solution* (Figure 1.1).

Figure 1.1: The two prestigious Apple Design Awards

Osman Ratib, Professor of Radiology in UCLA, returned to Geneva at the end of 2005 as the chairman of the Nuclear Medicine service. He was the first and unconditional supporter of the OsiriX project. Thanks to his commitment, OsiriX became

[4] `www.sourceforge.net/projects/osirix/`

[5] `homepage.mac.com/rossetantoine/osirix/`

the official DICOM viewer for the Radiology Department of the Geneva University Hospital in 2009.

At this time, Antoine Rosset and Joris Heuberger became the core team of OsiriX software. Over the years, OsiriX has benefitted from many external contributors. The most active external contributor was Lance Pysher MD, a US radiologist. He notably added the foundation for the DICOM Network protocol, through the DCMTK library (Offis) and wrote an Objective-C framework for DICOM files management (DCM Framework).

In March 2009, Antoine Rosset, Joris Heuberger and Osman Ratib created the *OsiriX Foundation*[6] to promote open-source in medicine. This non-profit foundation offers grants to students for developing open-source software in medicine. The foundation also organizes awards with prizes to stimulate development in digital imaging and post-processing. Most of these projects are based around OsiriX.

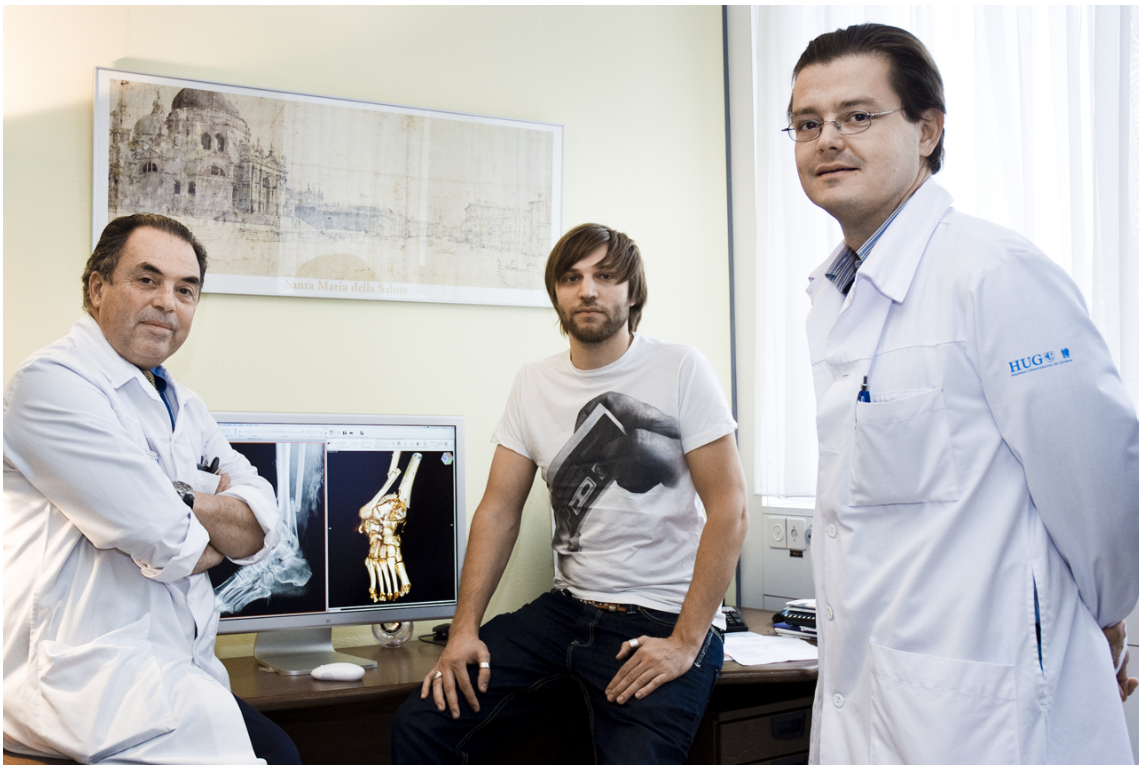


Figure 1.2: From left to right: Osman Ratib, Joris Heuberger and Antoine Rosset

In February 2010, Antoine Rosset and Joris Heuberger created the company Pixmeo[7] to continue to promote open-source solutions in medical imaging, including OsiriX development. The major goal of this company is to certify OsiriX as a FDA cleared product and offer commercial support for open-source solutions, such as OsiriX or dcm4chee.

Today, Antoine Rosset is working as a radiologist in La Tour[8] private hospital. Joris Heuberger is working as a Mac OS and iOS developer.

Their commitment to the OsiriX project is intact, thanks to the support of more than 40,000 users throughout the world. From a student project, OsiriX is today a mature and professional project, benefiting from experienced developers and users and used by thousand of institutions, including the most prestigious medical centers.

[6] `www.osirixfoundation.com`
[7] `www.pixmeo.com`
[8] `www.latour.ch`

The first iPhone (iOS) version of OsiriX was released in November 2008, developed by Joris Heuberger. OsiriX for iOS quickly became a major success.

1.2 Indications For Use

OsiriX is a software device intended for the viewing of images acquired from CT, MR, CR, DR, US and other DICOM compliant medical imaging systems when installed on suitable commercial standard hardware. Images and data can be captured, stored, communicated, processed and displayed within the system and/or across computer networks at distributed locations. Lossy compressed Mammographic images and digitized film screen images must not be reviewed for primary diagnosis or image interpretation. Mammographic images should only be viewed with a monitor approved by FDA for viewing Mammographic images. It is the user's responsibility to ensure monitor quality, ambient light conditions and that image compression ratios are consistent with clinical application.

1.3 Contraindications

None

1.4 Reporting A Problem

Pixmeo, the company that distributes OsiriX, is responsible for ensuring that OsiriX is working correctly, without errors and producing correct image quality and measurements. Each release of OsiriX is fully tested and validated, but some problems can remain unknown, only discovered within a specific situation or when connected to another software environment. When problems occur with OsiriX, users should communicate them to Pixmeo, allowing Pixmeo to correct these problems within a short timeframe and to let other users know about them, if necessary. Each problem will be evaluated to determine how serious it is and a workaround or a patch will be provided.

Pixmeo can be contacted by the following means:

Mail Box	Pixmeo Sarl 266 rue de Bernex CH-1233 Bernex Switzerland
Email	`pixmeo@pixmeo.com`
Web	`http://pixmeo.pixmeo.com/login/`

1.4.1 How to report a problem

For a crash:

1. Go to the */Applications/Utilities* folder
2. Launch the *Console* application
3. In the *View* Menu, choose *Show Log List* if not displayed
4. Find and select: *~/Library/Logs* ▸ *CrashReporter* ▸ *OsiriX.crash.log*
5. Copy and paste the entire crash log in your bug report (see Figure 1.3)

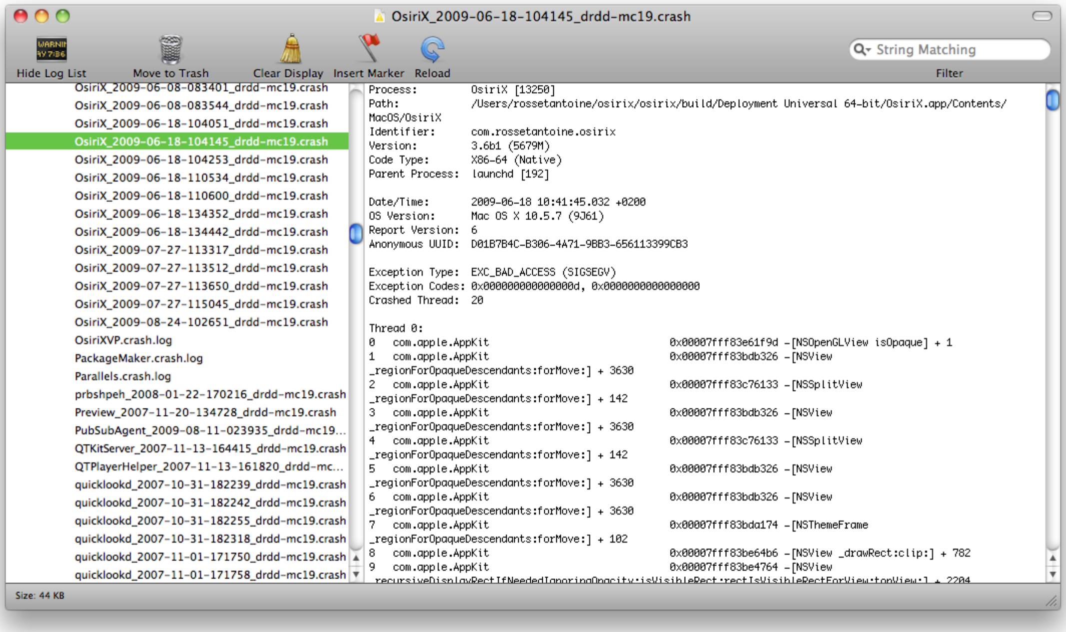


Figure 1.3: Console Application

For a hang (spinning ball):

1. Go to the */Applications/Utilities* folder
2. Launch *Activity Monitor* application (Figure 1.4)
3. Select OsiriX in the list (displayed in red for hanging applications)
4. In the *View* Menu, choose *Sample Process*
5. Click on the *Save...* button in the top right window
6. Copy and paste the entire report in your bug report

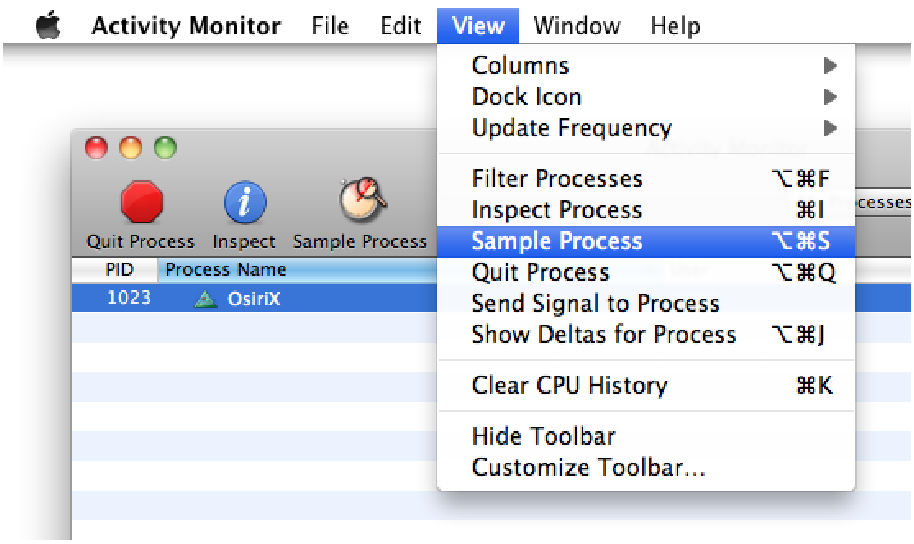


Figure 1.4: Activity Monitor Application

In some cases, corrupted DICOM images may interfere with OsiriX and may cause OsiriX to crash. You can restart OsiriX in a *safe mode* by holding down the shift and option key (⇧ + ⌥) while starting OsiriX. In *safe mode*, OsiriX will NOT read and display the content of DICOM files. This will allow you to delete the studies or series that may be corrupted.

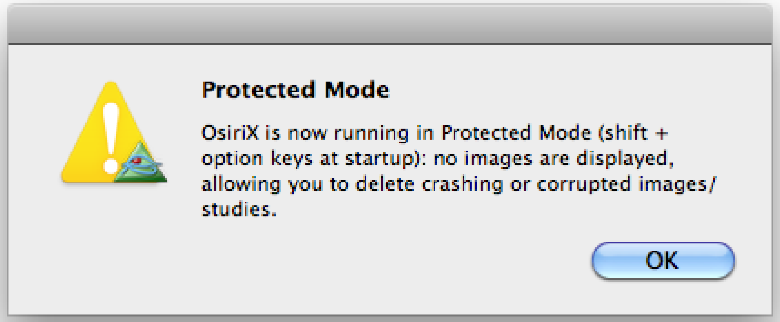


Figure 1.5: Protected Mode

1.5 Requirements

This section describes the software and hardware specifications required to run OsiriX. OsiriX can only be installed on an Apple Mac running Mac OS X. It cannot be installed on a Windows-based PC or Linux system.
The *minimal* requirements are as follows:

Type	Apple Mac
Operating System	Mac OS X 10.5 *Leopard* or higher
CPU	Intel based
Memory	2GB of RAM
Monitors Res.	1280 × 1024 pixels

The *recommended* configuration is the following:

Model	Mac Pro
OS	Mac OS X 10.6 *Snow Leopard*
CPU	Two 2.26GHz Quad-Core Intel Xeon
Memory	8GB of DDR3 ECC SDRAM
HDD	1TB 7200 rpm Serial ATA hard drive or SSD drive
Monitor	One or two monitors certified for medical imaging capable of displaying at least 1280 × 1024 pixels

Although OsiriX will work on any Macintosh platform, performances of some of the advanced image rendering tools can vary significantly. For Volume Rendering of large datasets, it is recommended to use high end machines equipped with multiple cores. OsiriX takes advantage of multi-processors and multi-core machines to speed up rendering functions.

When transferring and reading thousands of files, hard drive performance is important. If you handle multiple CT or MR studies with thousands of images, it is a good idea to install a SSD (Solid State Drive) instead of a standard hard drive. Performance can be up than 20 times faster compared to a standard hard drive.

Seven

3D Volume Rendering & 3D Maximum Intensity Projection

This Chapter describes how to visualize and render 3D datasets in OsiriX. The VR/MIP Viewer is a window displaying a set of DICOM images rendered as a 3D volume.

This 3D rendering technique is commonly used to visualize volumes of soft tissue data. It assigns different colors and transparencies to different intensity values in the data set. This technique can be applied to CT and MRI images with some adjustments. It is the most commonly used technique for *pseudo-realistic* rendering of 3D medical images. The predefined settings allow the user to generate reasonable images with very little adjustments. The simplest adjustment that the user will have to make is to the contrast and intensity of the image. This sets the threshold values of the image rendering algorithm that assigns a given opacity to the lowest level of intensity displayed. Thanks to this very simple maneuver it is then very easy to set the rendering of different tissue densities (skin, muscle or bones). The contrast and intensity assigned to the images will select the threshold density value used for rendering the opaque tissue level. High contrast will select bone density and therefore remove soft tissue and show only dense bone structures, while soft contrast will show only soft tissue such as skin and muscles.

7.1 The Anatomy of the VR/MIP Viewer

The window of the VR/MIP Viewer is divided in 2 parts:

- The toolbar
- The Image View

The toolbar contains buttons allowing the user to access the most useful functions available. As with any other Mac OS X toolbar, it can be customized to fit your needs: you can re-organize the tools, remove the ones you never use and add the ones you need. See 7.1.1 for a complete reference of the available tools.

The image view is the place where DICOM images are rendered as 3D images. You can interact with this area using the mouse and the keyboard and by using several tools that are described in this chapter.

7.1.1 Toolbar

The VR/MIP Viewer window provides a variety of tools and functions that can be accessed through icons displayed on the toolbar or through items listed in the *3D Viewer* menu. This section describes each tool available in the toolbar.

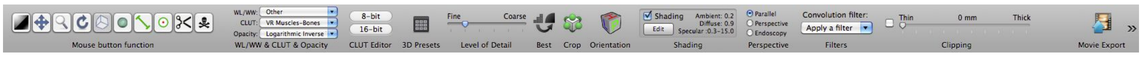
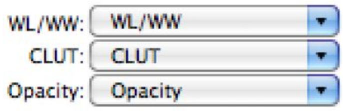


Figure 7.1: VR/MIP Viewer Window Toolbar

7.1.1.1 WL/WW

This allows the user to change the settings for WL/WW using presets.



7.1.1.2 CLUT

This allows the user to change the settings for the CLUT using presets. You can also create a new CLUT using the 16-bit CLUT Editor (see 7.2.5).

7.1.1.3 Opacity

This allows the user to change the settings for the opacity table using presets.

7.1.1.4 CLUT Editor

This allows the user to have easy access to the 8-bit and 16-bit editors.

7.1.1.5 3D Presets

This allows the user to choose the rendering settings from a list of presets. You can select one of the proposed presets.

7.1.1.6 Level of Detail

This allows the user to set the level of details for the Volume Rendering of the 3D view. You can choose the level of detail by moving a slider between *Fine* and *Coarse*.

7.1.1.7 Best Rendering

This allows the user to temporarily force the 3D view to render images in high quality.

7.1.1.8 Cropping Cube

This allows the user to resize the rendering area, by limiting the rendered dataset. You can manipulate the 6 sides of the bounding box of the volume. Any data outside of this box will not be rendered.

7.1.1.9 *Orientation Cube*

This allows the user to display/hide the orientation labels and cube. The tool displays 4 labels (1 on each side of the 3D view) and a cube. It computes the orientation of the volume and displays the cube of the same orientation with the following labels on its sides:

- L – Left
- R – Right
- P – Posterior
- A – Anterior
- S – Superior
- I – Inferior

The 4 labels can contain any combination of these letters depending on the volume orientation.

7.1.1.10 *Shadings*

This allows the user to edit the settings for the shadings of the Volume Rendering. This item can only be used when the rendering mode is VR and not MIP. You can choose to turn the shadings on and off. You can modify the following parameters:

- Ambient coefficient, from 0.0 to 1.0
- Diffusion coefficient, from 0.0 to 1.0
- Specular coefficient, from 0.0 to 4.0
- Specular power, from 0.0 to 50.0

You can also choose these parameters from predefined presets.

7.1.1.11 *Perspective*

This allows the user to choose perspective parameters for the rendering. You can choose one of the following perspectives:

- Parallel
- Perspective
- Endoscopy

7.1.1.12 *Orientations*

This allows the user to switch from different predefined camera positions. You can choose to see the volume in one of the following positions:

- Axial
- Coronal
- Left Sagittal
- Right Sagittal

7.1.1.13 *Mouse button function*

This allows the user to select a tool for the mouse left-click. You can choose to assign one of the following tools to the left mouse button:

- WL/WW
- Pan
- Zoom
- Rotate
- Volume rotate *[default left click]*
- Plane rotate
- Length
- 3D Point
- Scissors
- Bones removal

7.1.1.14 *Mode*

You can choose for the 3D view to be either rendered in VR or in MIP.

7.1.1.15 *Fusion*

This allows the user to modify the fusion percentage. If no Series is fused, the tool is not activated. You can modify the fusion percentage.

7.1.1.16 *4D Player*

The user can animate the time parameter of a 4D Series. You can play/pause the animation, choose the frame rate and choose the time position. In order to be activated, a 4D dataset is required.

7.1.1.17 *Stereo*

This allows the user to turn the stereo display on and off. It creates an anaglyph image and requires two color glasses (red/blue).

7.1.1.18 *Movie Export*

This allows the user to create a QuickTime movie of the 3D volume. You can set the following parameters:

- The number of frames to be generated (from 1 to 360)
- The direction of the rotation (horizontal or vertical)
- The amplitude of the rotation (180 ° or 360 °)
- The quality of the resulting rendering (Current or Best)
- The size of the resulting movie. Choose from the following:
  - Current
  - 512×512
  - 768×768

7.1.1.19 *iPhoto*

This allows the user to export the currently rendered image to iPhoto. This opens Apple's iPhoto application and adds the JPEG image to the Album defined in 2.2.1.

7.1.1.20 *Export QTVR*

This allows the user to create a QuickTime VR movie of the 3D volume. You can set the following parameters:

- The type of QuickTime VR movie to create:
  - X only axis rotation (18 frames)
  - X only axis rotation (36 frames)
  - 3D Rotation (100 frames)
  - 3D Rotation (400 frames)
  - 3D Rotation (1600 frames)
- The quality: Current or Best.
- The size of the resulting movie. Choose from the following:
  - Current
  - 512×512
  - 768×768

7.1.1.21 *Email*

This opens Apple's Mail application and creates a new email message containing the JPEG image as an attachment.

7.1.1.22 Reset

This resets the 3D Volume Rendering to the default viewing settings. It changes the following view settings to the default value:

- Zoom
- Rotation
- WL/WW
- Pan
- Camera Position

7.1.1.23 Revert

This reloads the data from the original DICOM files and re-renders the 3D volume. It cannot be undone.

7.1.1.24 Save as DICOM

This creates new DICOM images from snapshots of the 3D scene. It writes the resulting data on the disk and adds the new DICOM files to the database. You can choose from the following options:

- The name of the resulting Series
- The size of the resulting images. Choose from the following:
  - Current
  - 512×512
  - 768×768
- The image format:
  - 8-bit RGB
  - 16-bit black and white, only available with MIPrendering.
- The sequence to record:
  - The current image only
  - Animating the 4th dimension (if the data has 4 dimensions)
  - A new series from the rotating volume, choosing:
    - the number of frame to render (from 1 to 360)
    - the amplitude of the rotation (180° or 360°)
    - the direction of the rotation (vertical or horizontal)
- The rendering quality:
  - Current
  - Best
- Mark the resulting images as key images: on/off.
- Send the resulting images to a DICOM node: on/off.

7.1.1.25 *Fly Thru*

This allows the user to create an animated series or movie, along a custom-made camera path as described in 6.1. The camera path is created by adding any number of key positions (or steps), as well as deleting and reordering them. A *reset* button allows you to remove all the steps. The steps can also be saved to an XML file and/or loaded from an XML file.

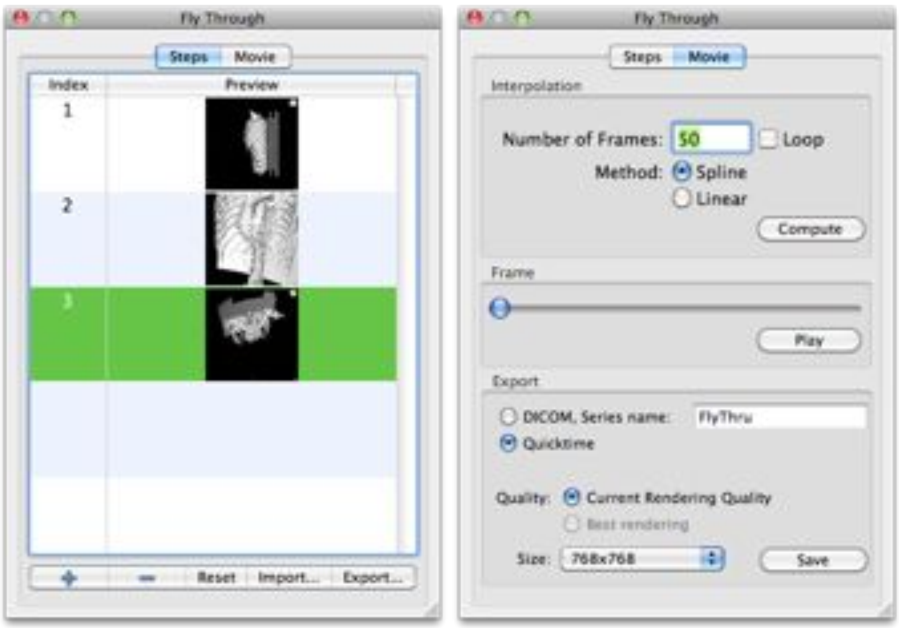
For the interpolation, you can set the following parameters:

- The number of frames
- To make a movie loop: on/off.
- The interpolation method: Spline or Linear.

You can preview the result of the interpolation, by either playing the full animation or by scrolling through it.

You can choose to export the animation as a new DICOM series or as a QuickTime movie. The resulting animation can be rendered in *Current* or *Best quality*. The size of the resulting images can be chosen from the following:

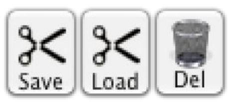
- Current
- 512×512
- 768×768



(a) Fly Thru Steps (b) Fly Thru Export Settings

Figure 7.2: Fly Thru Panel

7.1.1.26 *3D Scissors State*



This saves or loads a *3D Scissors State* done with the 3D Scissors tool. These items are also available in the *3D Viewer* menu.

7.1.1.27 ROI Manager

This allows the user to manage the appearance of 3D ROIs and compute their volume. For each ROI, you can set the following parameters:

- Map a texture: on/off.
- Red value from 0.0 to 1.0
- Green value from 0.0 to 1.0
- Blue value from 0.0 to 1.0
- Opacity value from 0.0 (transparent) to 1.0 (opaque)

7.1.1.28 Filters

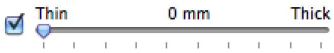
This applies a filter of convolution to the image data. The following filters are available:

- Basic Smooth 5×5
- Blur 3×3
- Blur 5×5
- Bone Filter 3×3
- Edge 3×3
- Emboss heavy
- Emboss north
- Emboss west
- Excessive edges
- Gaussian blur
- Hat
- Highpass 5×5
- Inverted blur
- Laplacian
- Laplacian 4
- Laplacian 8
- Lowpass
- Negative blur
- 3×3 Sharpen
- 5×5 sharpen

7.1.1.29 *Background Color*

This allows the user to change the color of the background of the 3D view. You can choose any color from the standard Mac OS X color picker.

7.1.1.30 *Clipping*



This allows the user to activate, deactivate and move 2 clipping planes. You can turn the clipping planes on and off. You can also define the distance between them (from *thin* to *thick*). Any data outside the 2 planes will not be rendered.

7.2 3D Viewing Functions

7.2.1 Volume Rendering

To render a 2D projection of the 3D data set, the user first needs to define a camera in 3D space relative to the volume. The user then needs to define the opacity and color of every voxel. This is done using an RGBA (for red, green, blue and alpha) transfer function that defines the RGBA value for every possible voxel value (Color Look Up Table, Opacity Table and Shading Table).

The resulting image is created through a volume ray casting algorithm. The technique of volume ray casting can be derived directly from the rendering equation. It provides high quality results and is usually considered to provide the best image quality. In this technique, a ray is generated for each desired image pixel. Using a simple camera model, the ray starts at the center of projection of the camera (the eye point,) and passes through the image pixel onto the imaginary image plane, floating in between the camera and the volume to be rendered. The ray is clipped by the boundaries of the volume in order to save time. The ray is then sampled at regular intervals throughout the volume (Level Of Detail, or LOD, parameter). The data is interpolated at each sample point (SuperSampling parameter), the transfer function (CLUT, Opacity and Shading Tables) applied to form an RGBA sample, the sample is composited onto the accumulated RGBA of the ray and the process repeated until the ray exits the volume. The RGBA color is converted to an RGB color and deposited in the corresponding image pixel. The process is repeated for every pixel on the screen to form the completed image.

7.2.2 Maximum Intensity Projection

The VR/MIP Viewer also supports the Maximum Intensity Projection mode. MIP rendering is similar to VR mode. It uses the same ray casting algorithm, but instead of according an opacity and color value to each voxel, depending on their positions, the ray casting keeps only the pixel with the highest value. This implies that two MIP renderings from opposite viewpoints are symmetrical images. Hence, the resulting 2D image doesn't provide a good sense of depth of the original data, in comparison to the 3D VR engine. The resulting image is not in RGB, but only with a single channel. The MIP image can then be saved as a 16-bit image. The 16-bit image keeps the entire dynamic values of the original dataset.

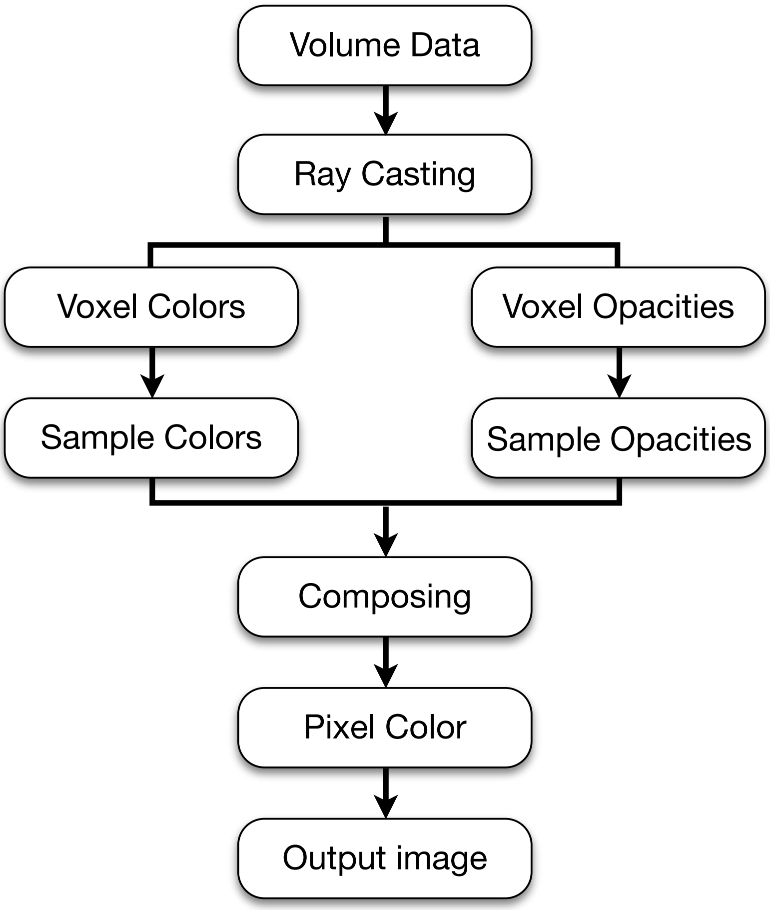


Figure 7.3: Volume Rendering Pipeline.

(a) VR Rendering (b) MIP Rendering

Figure 7.4: The same dataset rendered in VR and in MIP

7.2.3 3D Volume Rendering Engine

The rendering image is computed in real-time on CPUrather than on GPUṪhis engine is multi-processor and multi-core enhanced (it fully utilizes all the available processors and cores). Its performance is therefore related to the CPU performance (number of processor cores). The 3D engine is the most CPU intensive task of OsiriX. If you are using OsiriX specifically for 3D renderings, you should choose a MacPro computer with a maximum number of processors or cores.

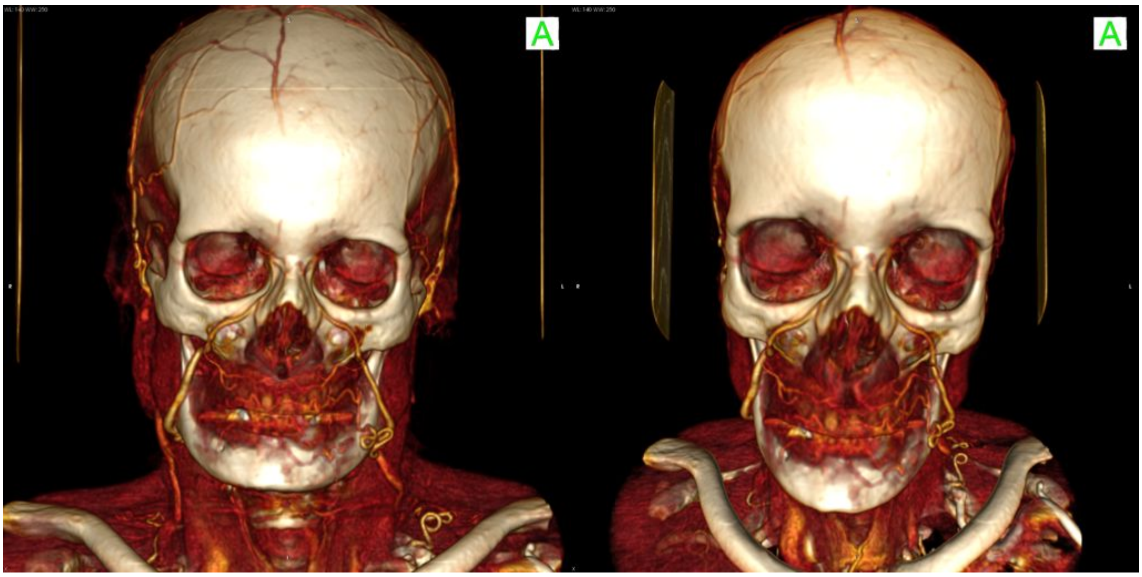
7.2.4 Projections mode

There are three options for the 3D projection mode: parallel, perspective or endoscopy. Perspective and endoscopy modes use a one-point linear perspective projection: the farther the object is, the smaller it appears.

The perspective mode uses a 30° view angle for the field of view. The camera position is *outside* the volume data.

The endoscopy mode uses a 60° view angle for the field of view. The camera position is *inside* the volume data.

The parallel mode uses parallel ray casting. All objects are displayed at the same ratio. There is no perspective representation. The disadvantage of this representation is that it is a non realistic representation. The advantages are a faster rendering speed and the possibility to measure distances on the resulting image, as there is no perspective. You can for example measure the diameter of a mass on the rendered image.



(a) Parallel Projection (b) Perspective Projection

Figure 7.5: The same dataset with parallel and perspective projection

7.2.5 16-bit CLUT Editor

This allows for the edition and creation of a 16-bit CLUT. You can assign a color and an opacity to each of the intensity levels of the table. It records the RGB values of the colors and the opacity in association with the intensity level value. By default, OsiriX provides several 16-bit CLUTs.

When activated, a new drawer will open at the bottom of the VR/MIP Viewer containing the 16-bit CLUT Editor. This drawer displays the histogram of your series (in gray) as well as a colored curve. For example in CT studies, on the left, the histogram shows the low density values while high density values are on the right. You can create a new color curve either by right clicking on the histogram and selecting the *New Curve* item, or by clicking on the ′+′ button on the left.

You can also change the color of the whole curve or of some specific points. To do so, double click on the square widget to modify the color of the whole curve, or on a round widget to modify only the selected point.

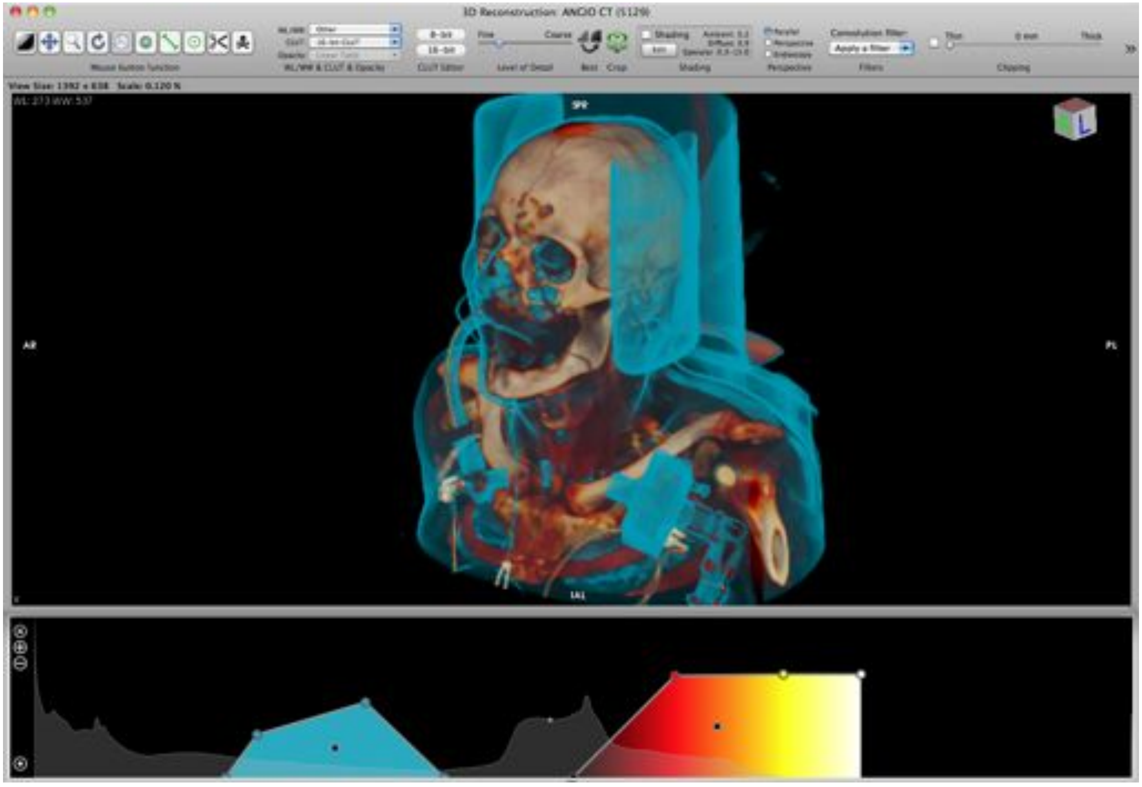


Figure 7.6: 16-bit CLUT Editor at the window bottom

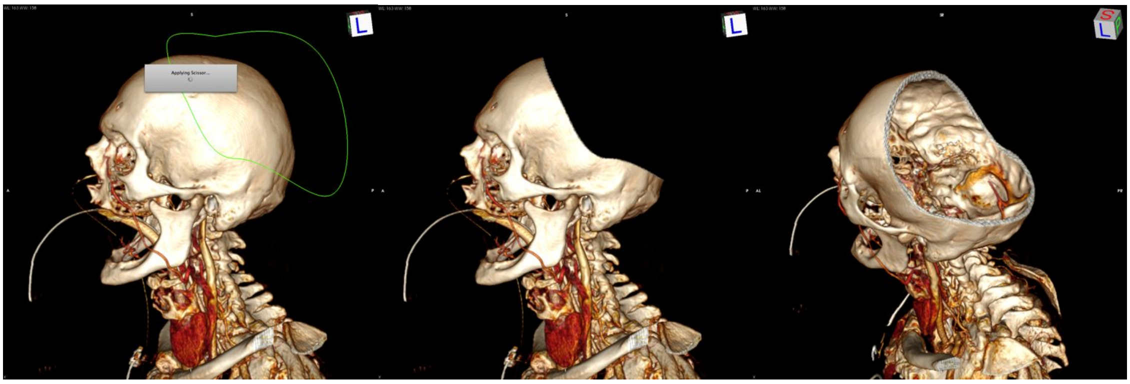
You can move the points (or the whole curve) vertically. This will determine the opacity: the bottom represents a transparent curve while the top represents an opaque curve.

You can also save your CLUT to use it later with other predefined CLUTs.

7.2.6 Sculpting 3D image

You can sculpt the 3D volume data to remove parts of the 3D dataset. By removing certain structures you can display hidden structures. For example, you can remove the rib cage to show the heart structures on a thoracic CT.

To remove a structure, simply select the sculpting tool in the toolbar tool. Next, draw an irregular region of interest over the 3D image, by clicking on the points of a closed polygon. To remove a polygon, because it is not placed as you would like it to be, you can start over by pressing the *esc* key on the keyboard.



(a) Region to be removed (b) Delete key ⌫ (c) Result

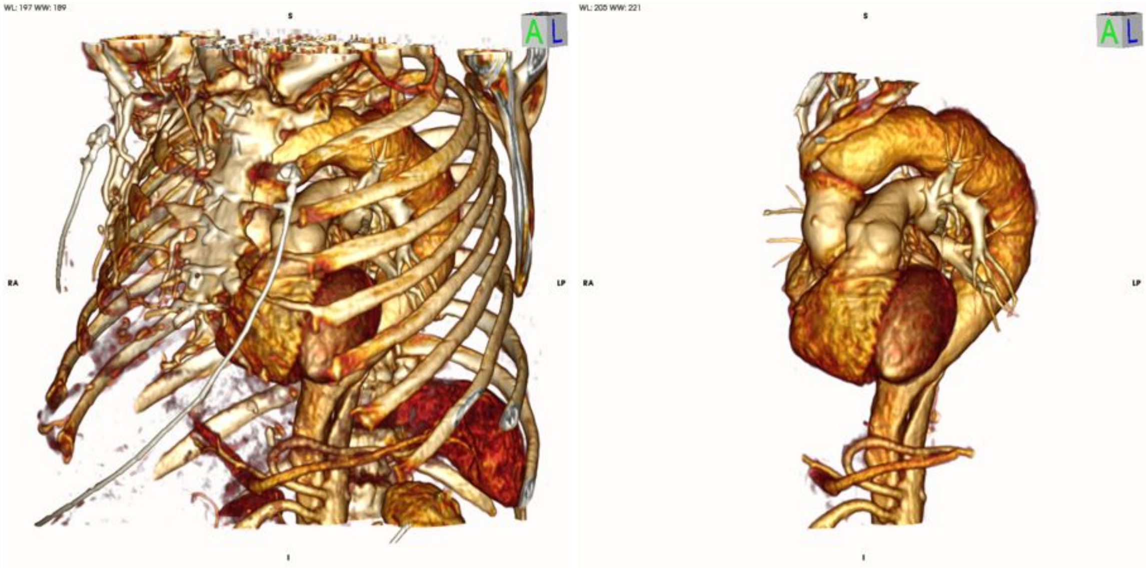
Figure 7.7: A 3D area under the drawn region is removed.

You then have three options:

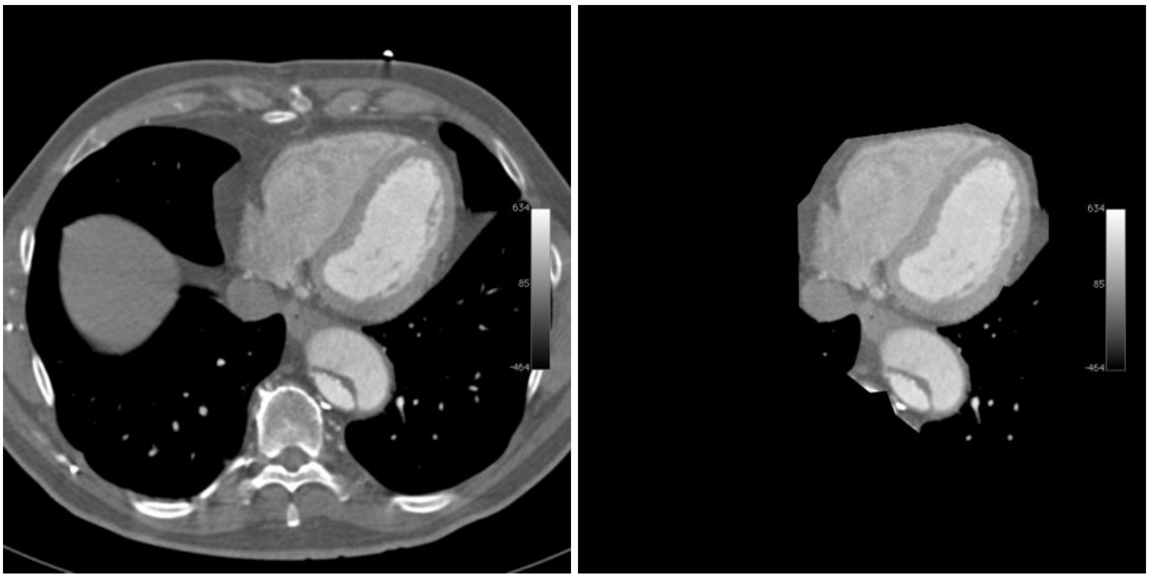
- To delete the 3D area under the drawn region, press the delete key ⌫.
- To delete everything except the drawn region, press the return key ↩.
- To revert the 3D area under the drawn region (reload the pixels from the original files), press the tab key ⇥.

The drawn region can be a polygon or a B-Spline rendered area, based on the 3D Preferences settings (see 2.2.2). The region represents the limits of an in-depth cut through the data. To undo a sculpting operation you can use the *Revert series* from the *2D Viewer* Menu (see 13.6.5) or *Undo* from the *Edit* Menu.

When deleting part of the image with the Scissors tool, the raw data is modified. That means the pixels of the original images are modified: the pixel intensity is set to the minimum value of the series pixels. For example, in a CT series, the pixels are modified to a value of -1024. When you close the VR/MIP Viewer window and go back to the 2D Viewer window, you will see the sculpted image, with missing pixels (as illustrated in Figure 7.8). You can reload these missing pixels at any time by using the *Revert series* from the *2D Viewer* Menu (see 13.6.5) or of the *3D Viewer* menu (see 13.7.9).

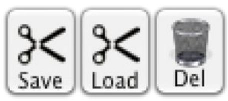


(a) Original 3D VR (b) Sculpted 3D VR



(c) Original CT (d) Modified CT

Figure 7.8: The 3D VR of a CT series (a) before and (b) after 3D sculpting. The original CT series (c) and the modified one (d).



Sculpting a 3D dataset with the Scissors tool can sometimes take a long time. It can be useful to save this sculpted dataset, to be able to reload it later. This function is available as a toolbar tool: *3D Scissors State*, or in the *3D Viewer* menu: *Scissors Editing*. It allows you to save the sculpted dataset and reload it later. Only one sculpted dataset can be saved for the current series. The sculpted dataset is saved as a single file in the *3DSTATE* folder, in the *OsiriX Data* folder.

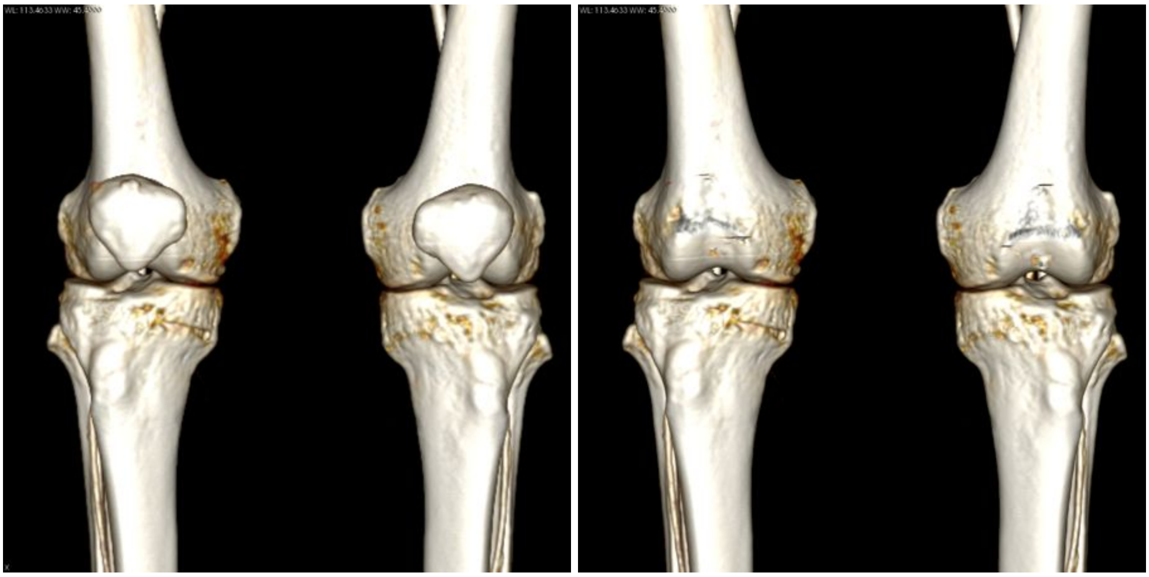
The effects of this tool can be undone, with the *Undo* item of the *Edit* menu.

The sculpting tool is often used in association with the cropping tool.

7.2.7 Bone Removal

A special tool for the removal of bone structures in CT studies is provided. It allows the user to eliminate all the parts of the image around a seed point that are within a certain range of intensities. The default values for bones are set between 250 and 2000 Hounsfield values for CT images. You can modify these values by clicking on the bone removal button while holding the option key pressed (⌥).

To remove bone structures simply click on it and all the adjacent and contiguous bone structures will be removed. You may repeat the operation on multiple bones that are not connected (see Figure 7.9).



(a) Original Image (b) After Bone Removal

Figure 7.9: The *Bone Removal* tool used to take the patellas away.

Sometimes some remaining fragments that don't have the perfect bone density may need to be manually removed using the sculpting tools (see 8.2.4).

This tool uses the same technique as the Scissors tool: it modifies pixel intensity, in order to hide them. This means that the pixels of the original images are modified: the pixels intensity is set to the minimum value of the series pixels.

The effects of this tool can be undone, with the *Undo* item of the *Edit* menu.

7.2.8 Cropping 3D volume

A cropping tool allows the user to restrict the data volume being rendered, by selecting the limits in X, Y and Z directions. Activating this tool in the toolbar causes green spheres to appear on the image together with a wire-frame parallelogram showing the limits of the volume being rendered. By clicking and dragging the green dots you can move each limit and adjust the rendered volume. Each plane of the 6 sides of the parallelogram can be moved independently. Notice that reducing the volume size not only allows you to hide unwanted parts of the data but also significantly speeds up the speed of image rendering. To hide the cropping-box click on the crop tool again.

Unlike the Scissors tool, the pixels are only hidden, not deleted. Hence, the original pixels values are not modified.

The cropping tool is often used in association with the sculpting tool.

7.2.9 Image Fusion

The VR/MIP Viewer supports image fusion. This means that if you open the VR/MIP Viewer window, from a 2D Viewer containing a series fused (see 5.5.9) with another one, you can see these two series in 3D. The fused 3D dataset is displayed *on* *top* of the other 3D dataset, without taking into account the original position of the structure. This means that the 3D dataset will always be in *front*. The fused 3D dataset doesn't support shading (see 7.1.1.10). It is always displayed without shading.

The fused 3D dataset is *locked* to the other dataset: they move and rotate together. To modify the WL/WW of the fused 3D dataset, you have to display the original 2D Viewer window, which contains the original dataset.

The cropping box tool will be applied to both datasets. The Scissors tool is not applied to the fused 3D dataset. If you want to use the Scissors tool on the fused dataset, you need to first sculpt it by opening a VR/MIP Viewer window from the 2D Viewer window, which contains the original dataset. Then you fuse this sculpted dataset with the second 2D Viewer window. Finally, open a VR/MIP Viewer window from the 2D Viewer window containing the fused datasets.

7.2.10 4D Dataset

The VR/MIP Viewer window also supports 4D datasets (see 3.1.1.2). Temporal sequences of 3D data are either obtained in separate series (CT and MRI image sets), or are stored in a single series (PET and SPECT image sets).

OsiriX allows the user to render these images in 3D while maintaining the fourth dimension of time activated resulting in a 4D dynamic image of a beating heart for example that can be manipulated in 3D. To do so you must first load the set of dynamic images using the 4D Viewer function in the database window (see 3.1.1.2). Once the dynamic set has been loaded, you can open the VR/MIP Viewer window. You can then use the 4D player controls on the toolbar (see 7.1.1.16). The dynamic display of 3D images can be activated by hitting the play button. You can also select a given frame of the cine sequence by using the slide control labelled *Pos*. Dynamic 4D images can also be exported in DICOM format or in QuickTime(see 7.3).

7.2.11 ROIs

The VR/MIP Viewer window can display some ROIs: 3D ROIs or Point ROI. The definition and the creation process of 3D ROIs is described in the 2D Viewer chapter (see 5.6.1.22). You have to create a 3D ROI in the 2D Viewer window: the VR/MIP Viewer window can display it, but cannot create it.

To display a 3D ROI, you have to open the 3D ROI Manager panel. You can either open it from the *ROI Manager* button, in the toolbar, or from the *ROI* Menu by selecting the *ROI Manager* item. You can then display or hide each 3D ROI and change the rendering settings for the corresponding 3D ROI(see 7.1.1.27).

The Point ROIs can be displayed and created in the VR/MIP Viewer window. To create a Point ROI in the VR/MIP Viewer window, select the *Point* button in the *Mouse button function* tool (see 7.1.1.13), in the toolbar. You can then drop a Point ROI directly onto the 3D rendered image. OsiriX will use the current WL/WW settings to determine where to drop the point in 3D: a ray casting is computed and the point is dropped when the 3D rendered structure is completely opaque. For example, in the case of an abdominal CT dataset, the point will be dropped on the skin if the rendered image shows the patient's skin and on the bone if the rendered image shows the patient's bone, depending on the WL/WW settings. The Point ROI created in the VR/MIP Viewer window will be also displayed in the 2D Viewer window.

You can change the appearance of the 3D Points ROIs, by double-clicking on a point (Figure 7.10).

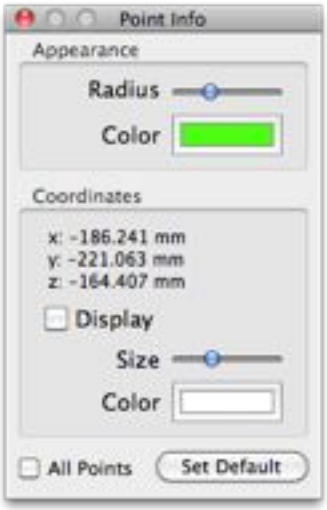
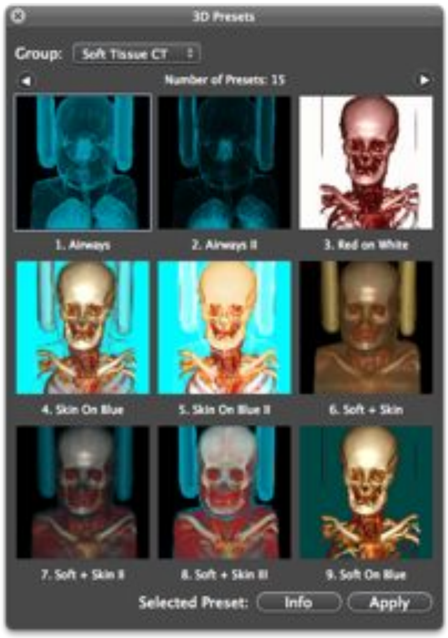


Figure 7.10: Settings to change the appearance of 3D Points ROIs.

7.2.12 3D Rendering Presets



OsiriX allows the user to store 3D rendering settings and color tables in *presets*. A set of predefined presets are provided by default, but you can add as many presets as needed. These presets can be stored in different groups. They can be easily classified according to imaging modality or by tissue or organ type, for example.

While in the 3D Volume Rendering mode, you can select a preset from the list by clicking on the *3D Presets* button or selecting the *Select 3D Preset...* item in the *3D Viewer* menu. A floating window with a list of presets with corresponding thumbnail views of the rendering mode is displayed.

It is important to note that the thumbnails are actually calculated images based on the current image displayed in the 3D window. To best view the effect of the presets on the image being viewed, it is important to zoom into the image in order to fill the screen and orient it in a suitable angle. This will allow the user to easily identify the real effect of each preset before applying it the full resolution image.

You can select between different lists from a pull down list of different groups. To apply a given preset select the preset from the list and click on the *Apply* button.

You can also review the detailed description of the settings by clicking on the *Info* button which will display a second floating window with the different parameters.

Some presets will use 8-bit colors tables while other will use the 16-bit color tables. When a preset is based on a 16-bit color table, it will automatically display the histogram and 16-bit color maps at the bottom of the window.

A preset also keeps the settings of filters that may be applied to the image as well as to the background color (note that this can be changed through the *Color* button of the toolbar).

You can save your presets by choosing the *Save Current State as 3D Preset...* item in the *3D Viewer* menu. A dialog box will prompt for a name and a group in which this preset should be saved. Unlimited numbers of presets can be saved in any number of groups.

7.3 Exporting 3D Images

The volume rendered images can be exported in different formats. They can be exported in DICOM format, generating a new series of images in the database window. These images can then be exported to a PACS or another DICOM compliant workstation for display.

If the images are rendered in MIP rendering mode, you have the option to export the images with 16-bit depth. This option allows the user to keep the original pixel intensity. You can then measure the Hounsfield Units values for a CT study, for example.

The images can also be exported in a QuickTime movie format as a sequence of images. There are two different formats that QuickTime supports: either a straight sequence of images displayed sequentially (as a rotating object over 180° or 360°) or as a 3D object that can be manipulated interactively in QuickTime VR format. It is important to reduce the matrix size of the image to minimize the file size, especially when a large number of images are needed for the output file. The resulting size of the image (matrix size) can be adjusted by dragging the lower left corner of the 3D view by clicking and dragging the *X* character (see Figure 7.11).

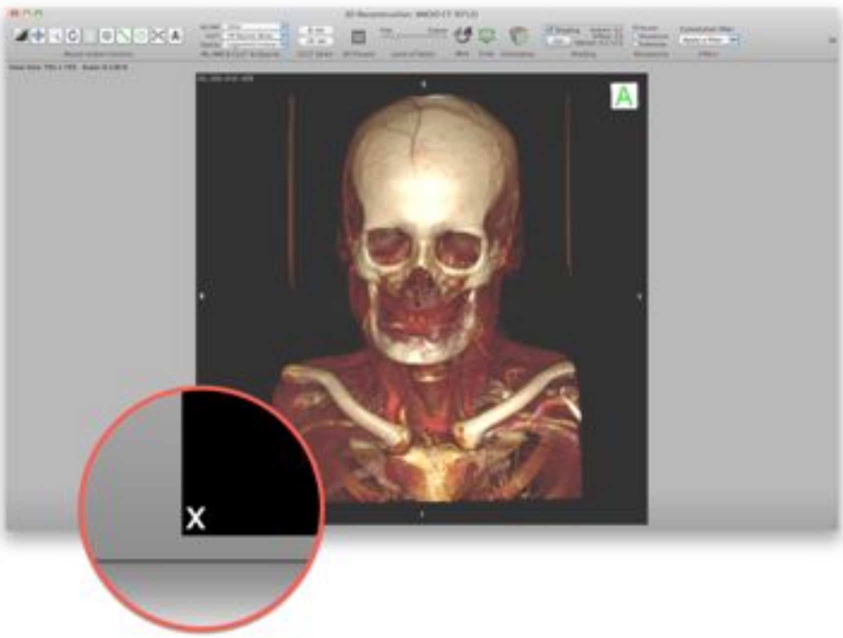


Figure 7.11: To resize the 3D view, drag the *X* located in the lower left corner.

Different compression schemes can be selected when generating QuickTime files to obtain different file sizes. A good compression algorithm is the JPEG compression: it allows the user to enjoy high quality on still images, while reducing the final file size. If you choose a video compression algorithm, such as MPEG4 or AVI, the still image quality will be low. These video algorithms can be optimized for video with a high frame rate (>25 images/sec).

QuickTime files can also be converted into WMV files for native support in computers running Windows, with *Flip4Mac*[1].

[1] http://www.telestream.net/flip4mac-wmv/overview.htm

7.3.1 Exporting QuickTime Movies

To export a 3D image sequence to a QuickTime movie simply select the QuickTime icon on the toolbar, or select it from the Export menu in the File menu. When exporting to QuickTime, a dialog will allow you to select the direction of rotation of the 3D object and the number of frames. The more frames the smoother the movie motion will be, but the file size will increase proportionally.

7.3.2 Exporting QuickTime VR Movies

Exporting a 3D rendered image set in QuickTime VR format generates a movie that can be interactively manipulated by the viewer by grabbing the 3D object and rotating it with the mouse. A zoom in and out option is also available. A QuickTime VR file is a very convenient of exporting 3D rendered images for users who do not have 3D rendering software but want to manipulate the images interactively in order to view them from different angles. When exporting to QuickTime VR a dialog box will allow you to set up the parameters of the generated movie. The motion of the object may be restricted in the horizontal or vertical directions, or can be unlimited in all 3D directions allowing the user to move the object in any orientation. The total number of images generated will determine the *smoothness* of the motion when the object is manipulated. The more images the better, but this will also increase the file size proportionally.