

Volume Reconstruction

1.0

Generated by Doxygen 1.8.0

Thu May 17 2012 18:03:12

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	Ui Namespace Reference	7
5	Class Documentation	9
5.1	Calibration Class Reference	9
5.1.1	Detailed Description	10
5.1.2	Member Typedef Documentation	10
5.1.2.1	DataType	10
5.1.3	Member Function Documentation	10
5.1.3.1	Calibrate	10
5.1.3.2	ClearImagePoints	10
5.1.3.3	ClearTransformations	10
5.1.3.4	getEstimatedUSCalibrationParameters	10
5.1.3.5	InsertImagePoints	10
5.1.3.6	InsertTransformations	10
5.1.3.7	New	10
5.1.4	Member Data Documentation	10
5.1.4.1	data	10
5.1.4.2	estimatedUSCalibrationParameters	10
5.1.4.3	imagePoints	10
5.1.4.4	transformations	11
5.2	MainWindow Class Reference	11
5.2.1	Detailed Description	12
5.2.2	Constructor & Destructor Documentation	12

5.2.2.1	MainWindow	12
5.2.2.2	~MainWindow	12
5.2.3	Member Function Documentation	12
5.2.3.1	addImages	12
5.2.3.2	addLogText	12
5.2.3.3	displaySelectedImage	12
5.2.3.4	getDisplayWidget	12
5.2.3.5	openVolumeData	12
5.2.3.6	print	12
5.2.3.7	probeCalibration	12
5.2.3.8	volumeReconstruction	12
5.2.4	Member Data Documentation	12
5.2.4.1	Connections	12
5.2.4.2	displayWidget	13
5.2.4.3	imagesFileNames	13
5.2.4.4	textOnTextArea	13
5.2.4.5	ui	13
5.2.4.6	volumeCalibrationData	13
5.2.4.7	volumeImagesFileNames	13
5.2.4.8	volumeRotationData	13
5.2.4.9	volumeTranslationData	13
5.3	ProbeCalibrationWidget Class Reference	13
5.3.1	Detailed Description	14
5.3.2	Constructor & Destructor Documentation	15
5.3.2.1	ProbeCalibrationWidget	15
5.3.2.2	~ProbeCalibrationWidget	15
5.3.3	Member Function Documentation	15
5.3.3.1	calibrate	15
5.3.3.2	crop	15
5.3.3.3	cropProbeImage	15
5.3.3.4	getCoordinates	15
5.3.3.5	loadRotationsFile	15
5.3.3.6	loadTranslationsFile	15
5.3.3.7	saveCalibration	15
5.3.3.8	setImage	15
5.3.3.9	setImageStack	15
5.3.3.10	setMainWindow	16
5.3.4	Member Data Documentation	16
5.3.4.1	calibrationParameters	16
5.3.4.2	coords	16

5.3.4.3	image	16
5.3.4.4	imageStack	16
5.3.4.5	mainWindow	16
5.3.4.6	rotations	16
5.3.4.7	rotations_2	16
5.3.4.8	translations	16
5.3.4.9	workWithStack	16
5.4	QVTKImageWidget Class Reference	17
5.4.1	Detailed Description	19
5.4.2	Constructor & Destructor Documentation	19
5.4.2.1	QVTKImageWidget	19
5.4.2.2	~QVTKImageWidget	19
5.4.3	Member Function Documentation	19
5.4.3.1	computeTransformation	19
5.4.3.2	displayImage	19
5.4.3.3	displaySelectedImage	19
5.4.3.4	displayVolume	19
5.4.3.5	displayVolumeImages	20
5.4.3.6	getImageDisplayedIndex	20
5.4.3.7	getImageHeight	20
5.4.3.8	getImageSize	20
5.4.3.9	getImageStack	20
5.4.3.10	getImageType	20
5.4.3.11	getImageViewer	20
5.4.3.12	getImageWidth	20
5.4.3.13	getNumOfDimensions	20
5.4.3.14	getPixelType	20
5.4.3.15	getQVTKWidget	20
5.4.3.16	getTransformScale	21
5.4.3.17	getTransformStack	21
5.4.3.18	getVolumeImageStack	21
5.4.3.19	getXPicked	21
5.4.3.20	getYPicked	21
5.4.3.21	setAndDisplayImage	21
5.4.3.22	setAndDisplayImage	21
5.4.3.23	setAndDisplayMultipleImages	21
5.4.3.24	setAndDisplayMultipleImages	22
5.4.3.25	setAndDisplayVolumeImages	22
5.4.3.26	setImageProperties	22
5.4.3.27	setXPicked	22

5.4.3.28	setYPicked	22
5.4.4	Member Data Documentation	22
5.4.4.1	cornerAnnotation	22
5.4.4.2	imageDisplayedIndex	22
5.4.4.3	imageHeight	23
5.4.4.4	imageStack	23
5.4.4.5	imageType	23
5.4.4.6	imageView	23
5.4.4.7	imageWidth	23
5.4.4.8	isImageStackLoaded	23
5.4.4.9	isVolumeImageStackLoaded	23
5.4.4.10	itkImage	23
5.4.4.11	numDimensions	23
5.4.4.12	pixelType	23
5.4.4.13	qvtkWidget	23
5.4.4.14	renderer	23
5.4.4.15	renwin	24
5.4.4.16	rgbItkImage	24
5.4.4.17	scale	24
5.4.4.18	transformStack	24
5.4.4.19	volumeDataCalibration	24
5.4.4.20	volumeDataRotations	24
5.4.4.21	volumeDataTranslations	24
5.4.4.22	volumeImageActorStack	24
5.4.4.23	volumeImageStack	24
5.4.4.24	vtkImage	24
5.4.4.25	xPicked	24
5.4.4.26	xPosition	24
5.4.4.27	yPicked	25
5.4.4.28	yPosition	25
5.5	QVTKImageWidgetCommand Class Reference	25
5.5.1	Detailed Description	25
5.5.2	Constructor & Destructor Documentation	25
5.5.2.1	QVTKImageWidgetCommand	25
5.5.2.2	~QVTKImageWidgetCommand	25
5.5.3	Member Function Documentation	26
5.5.3.1	Execute	26
5.5.3.2	New	26
5.5.3.3	SetAnnotation	26
5.5.3.4	SetImageWidget	26

5.5.3.5	SetPicker	26
5.5.4	Member Data Documentation	26
5.5.4.1	Annotation	26
5.5.4.2	ImageWidget	26
5.5.4.3	Picker	26
5.6	VolumeReconstruction Class Reference	26
5.6.1	Detailed Description	27
5.6.2	Member Function Documentation	27
5.6.2.1	calcImagePlane	27
5.6.2.2	calcMaxDistance	28
5.6.2.3	calcVoxelValue	28
5.6.2.4	generateVolume	28
5.6.2.5	New	28
5.6.2.6	setImageBoundsStack	28
5.6.2.7	setScale	28
5.6.2.8	setTransformStack	28
5.6.2.9	setVolumeImageStack	28
5.6.2.10	setVolumeOrigin	28
5.6.2.11	setVolumeSize	28
5.6.3	Member Data Documentation	28
5.6.3.1	imageBoundsXStack	28
5.6.3.2	imageBoundsYStack	29
5.6.3.3	imageBoundsZStack	29
5.6.3.4	imagePlaneStack	29
5.6.3.5	maxDistance	29
5.6.3.6	scale	29
5.6.3.7	transformStack	29
5.6.3.8	volumeImageStack	29
5.6.3.9	volumeOrigin	29
5.6.3.10	volumeSize	29
5.7	VolumeReconstructionWidget Class Reference	29
5.7.1	Detailed Description	31
5.7.2	Constructor & Destructor Documentation	31
5.7.2.1	VolumeReconstructionWidget	31
5.7.2.2	~VolumeReconstructionWidget	31
5.7.3	Member Function Documentation	31
5.7.3.1	calcImageBounds	31
5.7.3.2	calcImageCoords	31
5.7.3.3	calcVolumeSize	31
5.7.3.4	displayVolume	31

5.7.3.5	generate	31
5.7.3.6	invert	31
5.7.3.7	save	32
5.7.3.8	setDisplayProperties	32
5.7.3.9	setMainWindow	32
5.7.3.10	setTransformStack	32
5.7.3.11	setVolumeColorMap	32
5.7.3.12	setVolumeImageStack	32
5.7.3.13	setVolumeOpacity	32
5.7.3.14	transparency	32
5.7.4	Member Data Documentation	32
5.7.4.1	imageBoundsXStack	32
5.7.4.2	imageBoundsYStack	32
5.7.4.3	imageBoundsZStack	32
5.7.4.4	imageCoordsXStack	33
5.7.4.5	imageCoordsYStack	33
5.7.4.6	imageCoordsZStack	33
5.7.4.7	invertColors	33
5.7.4.8	mainWindow	33
5.7.4.9	scale	33
5.7.4.10	transformStack	33
5.7.4.11	ui	33
5.7.4.12	volume	33
5.7.4.13	volumeData	33
5.7.4.14	volumeFinal	33
5.7.4.15	volumeImageStack	33
5.7.4.16	volumeOrigin	33
5.7.4.17	volumeProperty	34
5.7.4.18	volumeSize	34
6	File Documentation	35
6.1	Calibration.cpp File Reference	35
6.2	Calibration.h File Reference	35
6.3	main.cpp File Reference	35
6.3.1	Function Documentation	35
6.3.1.1	main	35
6.4	mainwindow.cpp File Reference	35
6.5	mainwindow.h File Reference	36
6.6	ProbeCalibrationWidget.cpp File Reference	36
6.6.1	Variable Documentation	36

6.6.1.1	setCoordsSize	36
6.7	ProbeCalibrationWidget.h File Reference	36
6.8	QVTKImageWidget.cpp File Reference	37
6.9	QVTKImageWidget.h File Reference	37
6.9.1	Typedef Documentation	38
6.9.1.1	ImageType	38
6.9.1.2	RGBImageType	38
6.9.1.3	RGBPixelFormat	38
6.10	QVTKImageWidgetCommand.cpp File Reference	38
6.11	QVTKImageWidgetCommand.h File Reference	39
6.12	VolumeReconstruction.cpp File Reference	39
6.13	VolumeReconstruction.h File Reference	39
6.14	VolumeReconstructionWidget.cpp File Reference	39
6.15	VolumeReconstructionWidget.h File Reference	39

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Ui	7
----------	---

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Calibration	
Implements LSQRRecepies methods	9
MainWindow	
Main window for user interaction	11
ProbeCalibrationWidget	
Obtain data for calibration process	13
QVTKImageWidget	
Display VTK images	17
QVTKImageWidgetCommand	
Interaction with mouse	25
VolumeReconstruction	
Generate a new volume	26
VolumeReconstructionWidget	
Has the interaction methods for the user to generate a new volue	29

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Calibration.cpp	35
Calibration.h	35
main.cpp	35
mainwindow.cpp	35
mainwindow.h	36
ProbeCalibrationWidget.cpp	36
ProbeCalibrationWidget.h	36
QVTKImageWidget.cpp	37
QVTKImageWidget.h	37
QVTKImageWidgetCommand.cpp	38
QVTKImageWidgetCommand.h	39
VolumeReconstruction.cpp	39
VolumeReconstruction.h	39
VolumeReconstructionWidget.cpp	39
VolumeReconstructionWidget.h	39

Chapter 4

Namespace Documentation

4.1 Ui Namespace Reference

Chapter 5

Class Documentation

5.1 Calibration Class Reference

Implements LSQRRecepies methods.

```
#include <Calibration.h>
```

Public Types

- typedef
IsqrRecipes::SingleUnknownPointTargetUSCalibrationParametersEstimator::DataType [DataType](#)

Public Member Functions

- void [InsertTransformations](#) (vnl_matrix< double > rotationMatrix, vnl_vector< double > translation)
insert the rotation matrix of an image to transformations
- void [ClearTransformations](#) ()
clear transformations
- void [InsertImagePoints](#) (double p[2])
insert the crosswire point of an image to imagePoints
- void [ClearImagePoints](#) ()
clear imagePoints
- bool [Calibrate](#) ()
estimate calibration parameters
- std::vector< double > [getEstimatedUSCalibrationParameters](#) ()

Static Public Member Functions

- static [Calibration](#) * [New](#) ()
Constructor of the class.

Private Attributes

- std::vector< IsqrRecipes::Frame > [transformations](#)
- std::vector< IsqrRecipes::Point2D > [imagePoints](#)
contains the crosswire point in all images
- std::vector< [DataType](#) > [data](#)
contain the data of all images
- std::vector< double > [estimatedUSCalibrationParameters](#)

5.1.1 Detailed Description

Implements LSQRRecepies methods.

This classs have the calibration methods implemented in LSQRRecipes to calibrate an Ultra Sound Probe with a cross wire phantom.

5.1.2 Member Typedef Documentation

5.1.2.1 `typedef IsqrRecipes::SingleUnknownPointTargetUSCalibrationParametersEstimator::DataType Calibration::DataType`

5.1.3 Member Function Documentation

5.1.3.1 `bool Calibration::Calibrate ()`

estimate calibration parameters

5.1.3.2 `void Calibration::ClearImagePoints ()`

clear imagePoints

5.1.3.3 `void Calibration::ClearTransformations ()`

clear transformations

5.1.3.4 `std::vector< double > Calibration::getEstimatedUSCalibrationParameters ()`

5.1.3.5 `void Calibration::InsertImagePoints (double p[2])`

insert the crosswire point of an image to imagePoints

5.1.3.6 `void Calibration::InsertTransformations (vnl_matrix< double > rotationMatrix, vnl_vector< double > translation)`

insert the rotation matrix of an image to transformations

5.1.3.7 `static Calibration* Calibration::New () [inline, static]`

Constructor of the class.

5.1.4 Member Data Documentation

5.1.4.1 `std::vector< DataType > Calibration::data [private]`

contain the data of all images

5.1.4.2 `std::vector<double> Calibration::estimatedUSCalibrationParameters [private]`

5.1.4.3 `std::vector<IsqrRecipes::Point2D> Calibration::imagePoints [private]`

contains the crosswire point in all images

5.1.4.4 `std::vector<IsqrRecipes::Frame> Calibration::transformations` [private]

The documentation for this class was generated from the following files:

- [Calibration.h](#)
- [Calibration.cpp](#)

5.2 MainWindow Class Reference

Main window for user interaction.

```
#include <mainwindow.h>
```

Public Member Functions

- [MainWindow](#) (QWidget *parent=0)
- [~MainWindow](#) ()
- void [addLogText](#) (QString str)
- [QVTKImageWidget *](#) [getDisplayWidget](#) ()
return this display widget

Private Slots

- void [addImages](#) ()
Add image folder to application.
- void [displaySelectedImage](#) (int idx)
Display selected image with the image slider.
- void [probeCalibration](#) ()
Implements a ultrasound 3D probe calibration, for navigate with the probe.
- void [openVolumeData](#) ()
Set the image, rotation, translation and calibration parameters file name.
- void [print](#) ()
Print message in logger.
- void [volumeReconstruction](#) ()
*Calls the [VolumeReconstructionWidget.h](#) to generate a new volume *with the loaded data.*

Private Attributes

- Ui::MainWindow * [ui](#)
- QStringList [imagesFileNames](#)
The filename of each selected image.
- QStringList [volumeImagesFileNames](#)
The filenames of each selected volume image.
- QString [volumeRotationData](#)
The filename of the rotation data for each image.
- QString [volumeTranslationData](#)
The filename of the translation data for each image.
- QString [volumeCalibrationData](#)
The filename of the estimated parameters in the calibration.
- QString [textOnTextArea](#)
- [QVTKImageWidget *](#) [displayWidget](#)
- [vtkSmartPointer](#)
< [vtkEventQtSlotConnect](#) > [Connections](#)

5.2.1 Detailed Description

Main window for user interaction.

This class has the main interaction function to load volume images.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 `MainWindow::MainWindow (QWidget * parent = 0)` `[explicit]`

5.2.2.2 `MainWindow::~~MainWindow ()`

5.2.3 Member Function Documentation

5.2.3.1 `void MainWindow::addImages ()` `[private, slot]`

Add image folder to application.

5.2.3.2 `void MainWindow::addLogText (QString str)`

5.2.3.3 `void MainWindow::displaySelectedImage (int idx)` `[private, slot]`

Display selected image with the image slider.

5.2.3.4 `QVTKImageWidget * MainWindow::getDisplayWidget ()`

return this display widget

Parameters

out	<i>this</i>	display widget
-----	-------------	----------------

5.2.3.5 `void MainWindow::openVolumeData ()` `[private, slot]`

Set the image, rotation, translation and calibration parameters file name.

5.2.3.6 `void MainWindow::print ()` `[private, slot]`

Print message in logger.

5.2.3.7 `void MainWindow::probeCalibration ()` `[private, slot]`

Implements a ultrasound 3D probe calibration, for navigate with the probe.

5.2.3.8 `void MainWindow::volumeReconstruction ()` `[private, slot]`

Calls the [VolumeReconstructionWidget.h](#) to generate a new volume *with the loaded data.

5.2.4 Member Data Documentation

5.2.4.1 `vtkSmartPointer<vtkEventQtSlotConnect> MainWindow::Connections` `[private]`

5.2.4.2 QVTKImageWidget* MainWindow::displayWidget [private]

Central widget for display image purposes

5.2.4.3 QStringList MainWindow::imagesFileNames [private]

The filename of each selected image.

5.2.4.4 QString MainWindow::textOnTextArea [private]

5.2.4.5 Ui::MainWindow* MainWindow::ui [private]

5.2.4.6 QString MainWindow::volumeCalibrationData [private]

The filename of the estimated parameters in the calibration.

[Calibration](#) data filename

5.2.4.7 QStringList MainWindow::volumeImagesFileNames [private]

The filenames of each selected volume image.

List that contains the volume images filenames.

5.2.4.8 QString MainWindow::volumeRotationData [private]

The filename of the rotation data for each image.

Rotation data filename.

5.2.4.9 QString MainWindow::volumeTranslationData [private]

The filename of the translation data for each image.

Translation data filename.

The documentation for this class was generated from the following files:

- [mainwindow.h](#)
- [mainwindow.cpp](#)

5.3 ProbeCalibrationWidget Class Reference

Obtain data for calibration process.

```
#include <ProbeCalibrationWidget.h>
```

Public Member Functions

- [ProbeCalibrationWidget](#) (QWidget *parent=0)
- virtual [~ProbeCalibrationWidget](#) ()
- void [setImageStack](#) (std::vector< vtkSmartPointer< vtkImageData > > [imageStack](#))
Set this stack of vtkImageData.
- void [setImage](#) (vtkSmartPointer< vtkImageData > [image](#))

Set this vtkImageData.

- void [setMainWindow](#) ([MainWindow](#) *mainwindow)

Private Slots

- void [crop](#) ()
crop the images to delete the extra information
- void [getCoordinates](#) ()
get the cross point coordinates
- void [calibrate](#) ()
Calls Calibrate.h to estimate the calibration parameters.
- void [loadRotationsFile](#) ()
Load the rotations file.
- void [loadTranslationsFile](#) ()
Load the translation file.
- void [saveCalibration](#) ()
Save the Estimated Parameters in a .txt file.

Private Member Functions

- [vtkSmartPointer< vtkImageData > cropProbImage](#) ([vtkSmartPointer< vtkImageData > image](#), int depth-Type)

Private Attributes

- bool [workWithStack](#)
- [std::vector< vtkSmartPointer< vtkImageData > > imageStack](#)
an Array of vtkImageData to work
- [vnl_matrix< double > translations](#)
a vnl_matrix to store the translations of each image given by the tracker
- [vnl_matrix< double > rotations](#)
a vnl_matrix to store the rotations of each image given by the tracker
- [float rotations_2 \[7\]\[4\]](#)
- [MainWindow * mainWindow](#)
- [vtkSmartPointer< vtkImageData > image](#)
the vtkImageData to work
- [vnl_matrix< double > coords](#)
a vnl_matrix to store the selected coordinates on each image
- [std::vector< double > calibrationParameters](#)
the estimate calibration parameters by [Calibration.h](#)

5.3.1 Detailed Description

Obtain data for calibration process.

This class obtain the necessary data to use the Calibrion class. It let the user to set the croos wire point on the images, load Rotation and Translation Data. It also allows the user to save the calibration estimated paraeters in a .txt file

5.3.2 Constructor & Destructor Documentation

5.3.2.1 ProbeCalibrationWidget::ProbeCalibrationWidget (QWidget * *parent* = 0)

Constructor

5.3.2.2 ProbeCalibrationWidget::~~ProbeCalibrationWidget () [virtual]

5.3.3 Member Function Documentation

5.3.3.1 void ProbeCalibrationWidget::calibrate () [private, slot]

Calls Calibrate.h to estimate the calibration parameters.

5.3.3.2 void ProbeCalibrationWidget::crop () [private, slot]

crop the images to delete the extra information

5.3.3.3 vtkSmartPointer< vtkImageData > ProbeCalibrationWidget::cropProbeImage (vtkSmartPointer< vtkImageData > *image*, int *depthType*) [private]

Crop ultrasound image depending of the depth type

5.3.3.4 void ProbeCalibrationWidget::getCoordinates () [private, slot]

get the cross point coordinates

5.3.3.5 void ProbeCalibrationWidget::loadRotationsFile () [private, slot]

Load the rotations file.

5.3.3.6 void ProbeCalibrationWidget::loadTranslationsFile () [private, slot]

Load the translation file.

5.3.3.7 void ProbeCalibrationWidget::saveCalibration () [private, slot]

Save the Estimated Parameters in a .txt file.

5.3.3.8 void ProbeCalibrationWidget::setImage (vtkSmartPointer< vtkImageData > *image*)

Set this vtkImageData.

Parameters

in	<i>a</i>	smart Pointer of vtkImageData
----	----------	-------------------------------

5.3.3.9 void ProbeCalibrationWidget::setImageStack (std::vector< vtkSmartPointer< vtkImageData > > *imageStack*)

Set this stack of vtkImageData.

Parameters

in	a	std Vector of vtkImageData
----	---	----------------------------

5.3.3.10 void **ProbeCalibrationWidget::setMainWindow** (**MainWindow** * *mainwindow*)

Set the window to display the crop images

5.3.4 Member Data Documentation

5.3.4.1 **std::vector<double> ProbeCalibrationWidget::calibrationParameters** [private]

the estimate calibration parameters by [Calibration.h](#)

5.3.4.2 **vnI_matrix<double> ProbeCalibrationWidget::coords** [private]

a vnI_matrix to store the selected coordinates on each image

5.3.4.3 **vtkSmartPointer<vtkImageData> ProbeCalibrationWidget::image** [private]

the vtkImageData to work

5.3.4.4 **std::vector< vtkSmartPointer<vtkImageData> > ProbeCalibrationWidget::imageStack** [private]

an Array of vtkImageData to work

5.3.4.5 **MainWindow* ProbeCalibrationWidget::mainWindow** [private]

the main window to call it

5.3.4.6 **vnI_matrix<double> ProbeCalibrationWidget::rotations** [private]

a vnI_matrix to store the rotations of each image given by the tracker

5.3.4.7 **float ProbeCalibrationWidget::rotations_2[7][4]** [private]

5.3.4.8 **vnI_matrix<double> ProbeCalibrationWidget::translations** [private]

a vnI_matrix to store the translations of each image given by the tracker

5.3.4.9 **bool ProbeCalibrationWidget::workWithStack** [private]

The documentation for this class was generated from the following files:

- [ProbeCalibrationWidget.h](#)
- [ProbeCalibrationWidget.cpp](#)

5.4 QVTKImageWidget Class Reference

Display VTK images.

```
#include <QVTKImageWidget.h>
```

Public Member Functions

- [QVTKImageWidget](#) (QWidget *parent=0)
- virtual [~QVTKImageWidget](#) ()
- void [setAndDisplayImage](#) (QString imageFilename)
- void [setAndDisplayImage](#) (vtkSmartPointer< vtkImageData > image)
Sets and display the given vtkImageData.
- void [setAndDisplayMultipleImages](#) (QStringList filenames)
Set and display multiple images from a given images filenames. Display the image corresponding to the first element on the filenames list.
- void [setAndDisplayVolumeImages](#) (QStringList ImagesFilenames, QString rotationFilename, QString translationFilename, QString calibrationFilename)
Set and display images from a given images filenames that represent the volume data. Display the image corresponding to the first element on the filenames list.
- void [setAndDisplayMultipleImages](#) (std::vector< vtkSmartPointer< vtkImageData > > [imageStack](#))
Set and display multiple images from a given vtkImageData Array.
- void [displaySelectedImage](#) (int idx)
display an image stored in this imageStack.
- void [displayVolume](#) (vtkSmartPointer< vtkVolume > volume)
- QString [getPixelType](#) ()
Returns the pixel type in loaded images.
- QString [getImageType](#) ()
Returns the type of image displayed.
- QString [getNumOfDimensions](#) ()
Return the number of dimensions of the image.
- std::vector< vtkSmartPointer
 < vtkImageData > > [getImageStack](#) ()
return this image stack
- std::vector< vtkSmartPointer
 < vtkImageData > > [getVolumeImageStack](#) ()
return this volume image stack
- std::vector< vnl_matrix< double > > [getTransformStack](#) ()
return this transform stack
- int * [getImageSize](#) ()
- int [getImageWidth](#) ()
- int [getImageHeight](#) ()
- int [getXPicked](#) ()
Return the mouse x coordinate position when mouse left button is pressed.
- int [getYPicked](#) ()
Return the mouse y coordinate position when mouse left button is pressed.
- vnl_vector< double > [getTransformScale](#) ()
- void [setXPicked](#) (int xPosition)
Set the mouse x coordinate position when mouse left button is pressed.
- void [setYPicked](#) (int yPosition)
Set the mouse y coordinate position when mouse left button is pressed.
- vtkSmartPointer< vtkImageViewer2 > [getImageViewer](#) ()

Return this widget image viewer.

- QVTKWidget * [getQVTKWidget](#) ()

Return this qvtkWidget.

- int [getImageDisplayedIndex](#) ()

If an image stack is loaded, then return the index in the image stack of displayed image.

Public Attributes

- bool [isImageStackLoaded](#)

Flag to know if it's displayed an image stack.

- bool [isVolumeImageStackLoaded](#)

Flag to know if it's displayed an volume image stack.

Private Member Functions

- void [setImageProperties](#) (bool verbose)
- void [displayImage](#) (vtkImageData *image)
- void [displayVolumeImages](#) (std::vector< vtkSmartPointer< vtkImageData > > [volumeImageStack](#), vnl_matrix< double > [volumeDataRotations](#), vnl_matrix< double > [volumeDataTranslations](#), std::vector< double > [volumeDataCalibration](#))
- vnl_matrix< double > [computeTransformation](#) (vnl_vector< double > quaternion, vnl_vector< double > translation, std::vector< double > calibration)

Private Attributes

- QVTKWidget * [qvtkWidget](#)
- ImageType::Pointer [itkImage](#)
- RGBImageType::Pointer [rgbItkImage](#)
- vtkSmartPointer< vtkImageData > [vtkImage](#)
- vtkSmartPointer< vtkRenderWindow > [renwin](#)
- vtkSmartPointer< vtkRenderer > [renderer](#)
- std::vector< vtkSmartPointer
< vtkImageData > > [imageStack](#)

A vtkImageData Vector for keep the image references when load an image stack.

- std::vector< vtkSmartPointer
< vtkImageData > > [volumeImageStack](#)

A vtkImageData Vector for keep the volume image references when load an image stack.

- std::vector< vnl_matrix< double > > [transformStack](#)

A vtkImageData Vector for keep the volume image references when load an image stack Transformed.

- std::vector< vtkSmartPointer
< vtkImageActor > > [volumeImageActorStack](#)

A vtkImageData Vector for keep the volume image actor references when load an image stack.

- std::string [pixelType](#)
- vnl_vector< double > [scale](#)
- int [imageType](#)
- size_t [numDimensions](#)
- int [imageWidth](#)
- int [imageHeight](#)
- int [xPosition](#)
- int [yPosition](#)
- int [xPicked](#)
- int [yPicked](#)

- int [imageDisplayedIndex](#)
- vnl_matrix< double > [volumeDataRotations](#)
a *vnl_matrix* to store the rotations of each image given by the tracker
- vnl_matrix< double > [volumeDataTranslations](#)
a *vnl_matrix* to store the translations of each image given by the tracker
- std::vector< double > [volumeDataCalibration](#)
- vtkSmartPointer< vtkImageViewer2 > [imageView](#)
- vtkSmartPointer
< vtkCornerAnnotation > [cornerAnnotation](#)
Object for display information in the corners of the *vtkImageViewer2*.

5.4.1 Detailed Description

Display VTK images.

This class allows the user to display vtkImages on QT. It can display 2D and 3D scenes

5.4.2 Constructor & Destructor Documentation

5.4.2.1 QVTKImageWidget::QVTKImageWidget (QWidget * parent = 0)

Constructor for this ImageWidget

5.4.2.2 QVTKImageWidget::~QVTKImageWidget () [virtual]

Destructor

5.4.3 Member Function Documentation

5.4.3.1 vnl_matrix< double > QVTKImageWidget::computeTransformation (vnl_vector< double > quaternion, vnl_vector< double > translation, std::vector< double > calibration) [private]

Compute the transformation matricez of each image

5.4.3.2 void QVTKImageWidget::displayImage (vtkImageData * image) [private]

Display the given vtkImage

5.4.3.3 void QVTKImageWidget::displaySelectedImage (int idx)

display an image stored in this imageStack.

Parameters

in	the	index in the stack position of the image
----	-----	--

5.4.3.4 void QVTKImageWidget::displayVolume (vtkSmartPointer< vtkVolume > volume)

5.4.3.5 void QVTKImageWidget::displayVolumeImages (std::vector< vtkSmartPointer< vtkImageData > > *volumeImageStack*, vnl_matrix< double > *volumeDataRotations*, vnl_matrix< double > *volumeDataTranslations*, std::vector< double > *volumeDataCalibration*) [private]

Display the given volume images

5.4.3.6 int QVTKImageWidget::getImageDisplayedIndex ()

If an image stack is loaded, then return the index in the image stack of displayed image.

5.4.3.7 int QVTKImageWidget::getImageHeigth ()

returns this image heigth

5.4.3.8 int* QVTKImageWidget::getImageSize ()

returns an array with the width and height of the image

5.4.3.9 std::vector< vtkSmartPointer< vtkImageData > > QVTKImageWidget::getImageStack ()

return this image stack

5.4.3.10 QString QVTKImageWidget::getImageType ()

Returns the type of image displayed.

5.4.3.11 vtkSmartPointer< vtkImageViewer2 > QVTKImageWidget::getImageViewer ()

Return this widget image viewer.

Parameters

out	<i>imageViewer</i>	vtkImageViewer2 target 2D image.
-----	--------------------	----------------------------------

5.4.3.12 int QVTKImageWidget::getImageWidth ()

returns this image width

5.4.3.13 QString QVTKImageWidget::getNumOfDimesions ()

Return the numer of dimensions of the image.

5.4.3.14 QString QVTKImageWidget::getPixelType ()

Returns the pixel type in loaded images.

5.4.3.15 QVTKWidget * QVTKImageWidget::getQVTKWidget ()

Return this qvtkWidget.

Parameters

out	<i>the</i>	QVTKWidget
-----	------------	------------

5.4.3.16 `vnl_vector< double > QVTKImageWidget::getTransformScale ()`

5.4.3.17 `std::vector< vnl_matrix< double > > QVTKImageWidget::getTransformStack ()`

return this transform stack

5.4.3.18 `std::vector< vtkSmartPointer< vtkImageData > > QVTKImageWidget::getVolumeImageStack ()`

return this volume image stack

5.4.3.19 `int QVTKImageWidget::getXPicked ()`

Return the mouse x coordinate position when mouse left button is pressed.

Parameters

out	<i>int</i>	x position
-----	------------	------------

5.4.3.20 `int QVTKImageWidget::getYPicked ()`

Return the mouse y coordinate position when mouse left button is pressed.

Parameters

out	<i>int</i>	y position
-----	------------	------------

5.4.3.21 `void QVTKImageWidget::setAndDisplayImage (QString imageFilename)`

Sets and display an image from a given image path

5.4.3.22 `void QVTKImageWidget::setAndDisplayImage (vtkSmartPointer< vtkImageData > image)`

Sets and display the given vtkImageData.

Parameters

in	<i>a</i>	vtkImageData to set and display
----	----------	---------------------------------

5.4.3.23 `void QVTKImageWidget::setAndDisplayMultipleImages (QStringList filenames)`

Set and display multiple images from a given images filenames. Display the image corresponding to the first element on the filenames list.

Parameters

in	<i>a</i>	QStringList that contain the filename of each image
----	----------	---

5.4.3.24 void QVTKImageWidget::setAndDisplayMultipleImages (std::vector< vtkSmartPointer< vtkImageData >
> *imageStack*)

Set and display multiple images from a given vtkImageData Array.

Parameters

in	a	std::vector of vtkImageData
----	---	-----------------------------

5.4.3.25 void QVTKImageWidget::setAndDisplayVolumeImages (QStringList *ImagesFileNames*, QString
rotationFilename, QString *translationFilename*, QString *calibrationFilename*)

Set and display images from a given images filenames that represent the volume data. Display the image corresponding to the first element on the filenames list.

Parameters

in	a	QStringList that contain the filename of each image, a QStringList that contain the filename of the rotation data of each image and a QStringList that contain the filename the translation data of each image.
----	---	---

5.4.3.26 void QVTKImageWidget::setImageProperties (bool *verbose*) [private]

Set the needed image properties (pixelType, imageType, num of dimensions)

5.4.3.27 void QVTKImageWidget::setXPicked (int *xPosition*)

Set the mouse x coordinate position when mouse left button is pressed.

Parameters

out	int	x position
-----	-----	------------

5.4.3.28 void QVTKImageWidget::setYPicked (int *yPosition*)

Set the mouse y coordinate position when mouse left button is pressed.

Parameters

out	int	y position
-----	-----	------------

5.4.4 Member Data Documentation

5.4.4.1 vtkSmartPointer<vtkCornerAnnotation> QVTKImageWidget::cornerAnnotation [private]

Object for display information in the corners of the vtkImageViewer2.

5.4.4.2 int QVTKImageWidget::imageDisplayedIndex [private]

If image stack is displayed this sets a reference to current image displayed

5.4.4.3 `int QVTKImageWidget::imageHeight` [private]

Heigth of the image

5.4.4.4 `std::vector< vtkSmartPointer<vtkImageData> > QVTKImageWidget::imageStack` [private]

A vtkImageData Vector for keep the image references when load an image stack.

5.4.4.5 `int QVTKImageWidget::imageType` [private]

the number of scalar components in the image 1 => grayscale, 3 => rgb

5.4.4.6 `vtkSmartPointer<vtkImageViewer2> QVTKImageWidget::imageView` [private]

the image viewer for display images

5.4.4.7 `int QVTKImageWidget::imageWidth` [private]

Width of the image

5.4.4.8 `bool QVTKImageWidget::isImageStackLoaded`

Flag to know if it's displayed an image stack.

5.4.4.9 `bool QVTKImageWidget::isVolumeImageStackLoaded`

Flag to know if it's displayed an volume image stack.

5.4.4.10 `ImageType::Pointer QVTKImageWidget::itkImage` [private]

The grayscale image displayed in this widget

5.4.4.11 `size_t QVTKImageWidget::numDimensions` [private]

The number of the image dimensions

5.4.4.12 `std::string QVTKImageWidget::pixelType` [private]

The type of the image pixels

5.4.4.13 `QVTKWidget* QVTKImageWidget::qvtkWidget` [private]

The QVTKWidget for display and interact with the images

5.4.4.14 `vtkSmartPointer<vtkRenderer> QVTKImageWidget::renderer` [private]

The VTK renderer

5.4.4.15 `vtkSmartPointer<vtkRenderWindow> QVTKImageWidget::renwin` [private]

The VTK render window

5.4.4.16 `RGBImageType::Pointer QVTKImageWidget::rgbItrkImage` [private]

The RGB image displayed for this widget

5.4.4.17 `vnl_vector<double> QVTKImageWidget::scale` [private]

Estimated scale of the images

5.4.4.18 `std::vector<vnl_matrix<double> > QVTKImageWidget::transformStack` [private]

A vtkImageData Vector for keep the volume image references when load an image stack Transformed.

5.4.4.19 `std::vector<double> QVTKImageWidget::volumeDataCalibration` [private]

The angles and translation estimated

5.4.4.20 `vnl_matrix<double> QVTKImageWidget::volumeDataRotations` [private]

a vnl_matrix to store the rotations of each image given by the tracker

5.4.4.21 `vnl_matrix<double> QVTKImageWidget::volumeDataTranslations` [private]

a vnl_matrix to store the translations of each image given by the tracker

5.4.4.22 `std::vector< vtkSmartPointer<vtkImageActor> > QVTKImageWidget::volumeImageActorStack`
[private]

A vtkImageData Vector for keep the volume image actor references when load an image stack.

5.4.4.23 `std::vector< vtkSmartPointer<vtkImageData> > QVTKImageWidget::volumeImageStack` [private]

A vtkImageData Vector for keep the volume image references when load an image stack.

5.4.4.24 `vtkSmartPointer<vtkImageData> QVTKImageWidget::vtkImage` [private]

The VTK image to display i this window

5.4.4.25 `int QVTKImageWidget::xPicked` [private]

The x coordinate of the picked position over the image

5.4.4.26 `int QVTKImageWidget::xPosition` [private]

current x coordinate of mouse position over the image

5.4.4.27 int QVTKImageWidget::yPicked [private]

current y coordinate of picked position over the image

5.4.4.28 int QVTKImageWidget::yPosition [private]

current y coordinate of mouse position over the image

The documentation for this class was generated from the following files:

- [QVTKImageWidget.h](#)
- [QVTKImageWidget.cpp](#)

5.5 QVTKImageWidgetCommand Class Reference

Interaction with mouse.

```
#include <QVTKImageWidgetCommand.h>
```

Public Member Functions

- [QVTKImageWidgetCommand](#) ()
- [~QVTKImageWidgetCommand](#) ()
- void [SetPicker](#) (vtkSmartPointer< vtkPropPicker > picker)
- void [SetAnnotation](#) (vtkSmartPointer< vtkCornerAnnotation > annotation)
- void [SetImageWidget](#) ([QVTKImageWidget](#) *imageWidget)
Set the 2d image widget related to this 2d event manager.
- virtual void [Execute](#) (vtkObject *, unsigned long vtkNotUsed(event), void *)

Static Public Member Functions

- static [QVTKImageWidgetCommand](#) * [New](#) ()

Private Attributes

- vtkSmartPointer< vtkPropPicker > [Picker](#)
- vtkSmartPointer
 < vtkCornerAnnotation > [Annotation](#)
- [QVTKImageWidget](#) * [ImageWidget](#)

5.5.1 Detailed Description

Interaction with mouse.

The mouse motion callback, to pick the image and recover pixel values

5.5.2 Constructor & Destructor Documentation

5.5.2.1 QVTKImageWidgetCommand::QVTKImageWidgetCommand ()**5.5.2.2 QVTKImageWidgetCommand::~~QVTKImageWidgetCommand ()**

5.5.3 Member Function Documentation

5.5.3.1 `void QVTKImageWidgetCommand::Execute (vtkObject * , unsigned long vtkNotUsedevent, void *)`
[virtual]

5.5.3.2 `QVTKImageWidgetCommand * QVTKImageWidgetCommand::New ()` [static]

5.5.3.3 `void QVTKImageWidgetCommand::SetAnnotation (vtkSmartPointer< vtkCornerAnnotation > annotation)`

5.5.3.4 `void QVTKImageWidgetCommand::SetImageWidget (QVTKImageWidget * imageWidget)`

Set the 2d image widget related to this 2d event manager.

Parameters

in	viewer	QVTKImageWidget target 2D image
----	--------	---

5.5.3.5 `void QVTKImageWidgetCommand::SetPicker (vtkSmartPointer< vtkPropPicker > picker)`

5.5.4 Member Data Documentation

5.5.4.1 `vtkSmartPointer<vtkCornerAnnotation> QVTKImageWidgetCommand::Annotation` [private]

Pointer to the annotation

5.5.4.2 `QVTKImageWidget* QVTKImageWidgetCommand::ImageWidget` [private]

The widget related to the mouse events

5.5.4.3 `vtkSmartPointer<vtkPropPicker> QVTKImageWidgetCommand::Picker` [private]

Pointer to the picker

The documentation for this class was generated from the following files:

- [QVTKImageWidgetCommand.h](#)
- [QVTKImageWidgetCommand.cpp](#)

5.6 VolumeReconstruction Class Reference

Generate a new volume.

```
#include <VolumeReconstruction.h>
```

Public Member Functions

- void [setVolumeSize](#) (vnl_vector< double >)
Set the size of the volume data.
- void [setVolumeOrigin](#) (vnl_vector< double >)
Set the volume data origin in the 3D scene.
- void [setImageBoundsStack](#) (std::vector< vnl_vector< double > >, std::vector< vnl_vector< double > >,
std::vector< vnl_vector< double > >)
Set the image bounds.

- void [setVolumeImageStack](#) (std::vector< vtkSmartPointer< vtkImageData > >)
Set image data stack to generate the volume.
- void [setTransformStack](#) (std::vector< vnl_matrix< double > >)
Set the transformation for each image used in the reconstruction.
- void [setScale](#) (vnl_vector< double >)
Set the scale of the images.
- vtkSmartPointer< vtkImageData > [generateVolume](#) ()
Returns the new volume data with the voxel based method.

Static Public Member Functions

- static [VolumeReconstruction](#) * [New](#) ()
Constructor.

Private Member Functions

- void [calcImagePlane](#) ()
Compute the plane equation for each image.
- double [calcMaxDistance](#) ()
Computes the maximum distance in the volume.
- double [calcVoxelValue](#) (std::vector< vnl_vector< double > >, vnl_vector< double >, vnl_vector< double > >)
Computes the voxel value using three lineal interpolation.

Private Attributes

- vnl_vector< double > [volumeSize](#)
- vnl_vector< double > [volumeOrigin](#)
- std::vector< vnl_vector< double > > [imageBoundsXStack](#)
- std::vector< vnl_vector< double > > [imageBoundsYStack](#)
- std::vector< vnl_vector< double > > [imageBoundsZStack](#)
- std::vector< vtkSmartPointer
< vtkImageData > > [volumeImageStack](#)
- std::vector< vnl_matrix< double > > [transformStack](#)
- vnl_vector< double > [scale](#)
- std::vector< vtkSmartPointer
< vtkPlane > > [imagePlaneStack](#)
- double [maxDistance](#)

5.6.1 Detailed Description

Generate a new volume.

This class generate a new volume data using a voxel based method with the previously loaded data. It requires the images data, the tracker data and the estimated parameters from a calibration. The method implemented a nearest pixel interpolation.

5.6.2 Member Function Documentation

5.6.2.1 void VolumeReconstruction::calcImagePlane () [private]

Compute the plane equation for each image.

5.6.2.2 `double VolumeReconstruction::calcMaxDistance () [private]`

Computes the maximun distance in the volume.

5.6.2.3 `double VolumeReconstruction::calcVoxelValue (std::vector< vnl_vector< double > > , vnl_vector< double > , vnl_vector< double >) [private]`

Computes the voxel value using three lineal interpolation.

5.6.2.4 `vtkSmartPointer< vtkImageData > VolumeReconstruction::generateVolume ()`

Returns the new volume data with the voxel based method.

5.6.2.5 `static VolumeReconstruction* VolumeReconstruction::New () [inline, static]`

Constructor.

5.6.2.6 `void VolumeReconstruction::setImageBoundsStack (std::vector< vnl_vector< double > > imageBoundsXStack, std::vector< vnl_vector< double > > imageBoundsYStack, std::vector< vnl_vector< double > > imageBoundsZStack)`

Set the image bounds.

5.6.2.7 `void VolumeReconstruction::setScale (vnl_vector< double > scale)`

Set the scale of the images.

5.6.2.8 `void VolumeReconstruction::setTransformStack (std::vector< vnl_matrix< double > > transformStack)`

Set the transformation for each image used in the reconstruction.

5.6.2.9 `void VolumeReconstruction::setVolumeImageStack (std::vector< vtkSmartPointer< vtkImageData > > volumeImageStack)`

Set image data stack to generate the volume.

5.6.2.10 `void VolumeReconstruction::setVolumeOrigin (vnl_vector< double > volumeOrigin)`

Set the volume data orgin in the 3D scene.

5.6.2.11 `void VolumeReconstruction::setVolumeSize (vnl_vector< double > volumeSize)`

Set the size of the volume data.

5.6.3 Member Data Documentation

5.6.3.1 `std::vector< vnl_vector<double> > VolumeReconstruction::imageBoundsXStack [private]`

Stacks for the image Bounds in x

5.6.3.2 `std::vector< vnl_vector<double> > VolumeReconstruction::imageBoundsYStack` [private]

Stacks for the image Bounds in Y

5.6.3.3 `std::vector< vnl_vector<double> > VolumeReconstruction::imageBoundsZStack` [private]

Stacks for the image Bounds in Z

5.6.3.4 `std::vector< vtkSmartPointer<vtkPlane> > VolumeReconstruction::imagePlaneStack` [private]

The plane equation for each image

5.6.3.5 `double VolumeReconstruction::maxDistance` [private]

the maximun distance found in the volume

5.6.3.6 `vnl_vector<double> VolumeReconstruction::scale` [private]

scale of the images

5.6.3.7 `std::vector< vnl_matrix<double> > VolumeReconstruction::transformStack` [private]

Contains the transformation for each image

5.6.3.8 `std::vector< vtkSmartPointer< vtkImageData> > VolumeReconstruction::volumeImageStack`
[private]

The stack of images data

5.6.3.9 `vnl_vector<double> VolumeReconstruction::volumeOrigin` [private]

Where the volume data begins in the 3D scene

5.6.3.10 `vnl_vector<double> VolumeReconstruction::volumeSize` [private]

Size of the volume

The documentation for this class was generated from the following files:

- [VolumeReconstruction.h](#)
- [VolumeReconstruction.cpp](#)

5.7 VolumeReconstructionWidget Class Reference

Has the interaction methods for the user to generate a new volue.

```
#include <VolumeReconstructionWidget.h>
```

Public Member Functions

- [VolumeReconstructionWidget](#) (QWidget *parent=0)
- [~VolumeReconstructionWidget](#) ()
- void [setMainWindow](#) (MainWindow *mainwindow)
- void [setTransformStack](#) (std::vector< vnl_matrix< double > >)
- void [setVolumeImageStack](#) (std::vector< vtkSmartPointer< vtkImageData > >)

Private Slots

- void [save](#) ()
Saves the volume in a .mhd and .raw file.
- void [generate](#) ()
Calls [VolumeReconstruction.h](#) to generate the new volume data.
- void [invert](#) ()
Invert the color transfer function.
- void [transparency](#) ()
Changes the oacity of the volume.

Private Member Functions

- void [calcImageCoords](#) ()
Computes every pixel coord of each image in the 3D space.
- void [calcImageBounds](#) ()
Computes the coords of the images bounds int he 3D space.
- void [calcVolumeSize](#) (bool)
Computes the volume size.
- void [setVolumeOpacity](#) (int)
Set the volume opacity.
- void [setVolumeColorMap](#) (bool)
Set the volume colo transfer function.
- void [setDisplayProperties](#) (vtkSmartPointer< vtkImageData >)
Set the display properties of the volume and the rendering method.
- void [displayVolume](#) ()
Display the volume in the 3D scene.

Private Attributes

- Ui::VolumeReconstructionWidget * [ui](#)
- [MainWindow](#) * [mainWindow](#)
- std::vector< vtkSmartPointer
 < vtkImageData > > [volumeImageStack](#)
- std::vector< vnl_matrix< double > > [transformStack](#)
- std::vector< vnl_matrix< double > > [imageCoordsXStack](#)
- std::vector< vnl_matrix< double > > [imageCoordsYStack](#)
- std::vector< vnl_matrix< double > > [imageCoordsZStack](#)
- std::vector< vnl_vector< double > > [imageBoundsXStack](#)
- std::vector< vnl_vector< double > > [imageBoundsYStack](#)
- std::vector< vnl_vector< double > > [imageBoundsZStack](#)
- vtkSmartPointer< vtkVolume > [volume](#)
- vtkSmartPointer< vtkImageData > [volumeData](#)

- vtkSmartPointer
 < vtkVolumeProperty > [volumeProperty](#)
- vnl_vector< double > [volumeOrigin](#)
- vnl_vector< double > [volumeFinal](#)
- vnl_vector< double > [volumeSize](#)
- vnl_vector< double > [scale](#)
- bool [invertColors](#)

5.7.1 Detailed Description

Has the interaction methods for the user to generate a new value.

This class allows the user to choose between a voxel based method or a pixel based method to reconstruct a volume and set the main volume properties. It allows to change the opacity of the generated volume and to change the colors. This class also allows to the user to save the volume in a .mhd and .raw files

5.7.2 Constructor & Destructor Documentation

5.7.2.1 `VolumeReconstructionWidget::VolumeReconstructionWidget (QWidget * parent = 0)` `[explicit]`

5.7.2.2 `VolumeReconstructionWidget::~~VolumeReconstructionWidget ()`

5.7.3 Member Function Documentation

5.7.3.1 `void VolumeReconstructionWidget::calcImageBounds ()` `[private]`

Computes the coords of the images bounds in the 3D space.

5.7.3.2 `void VolumeReconstructionWidget::calcImageCoords ()` `[private]`

Computes every pixel coord of each image in the 3D space.

5.7.3.3 `void VolumeReconstructionWidget::calcVolumeSize (bool usePixelMethod)` `[private]`

Computes the volume size.

Parameters

<i>in</i>	<i>if</i>	bool is true it computes it with the imagecoords, else it uses the image bounds
-----------	-----------	---

5.7.3.4 `void VolumeReconstructionWidget::displayVolume ()` `[private]`

Display the volume in the 3D scene.

5.7.3.5 `void VolumeReconstructionWidget::generate ()` `[private, slot]`

Calls [VolumeReconstruction.h](#) to generate the new volume data.

5.7.3.6 `void VolumeReconstructionWidget::invert ()` `[private, slot]`

Invert the color transfer function.

5.7.3.7 void VolumeReconstructionWidget::save () [private, slot]

Saves the volume in a .mhd and .raw file.

5.7.3.8 void VolumeReconstructionWidget::setDisplayProperties (vtkSmartPointer< vtkImageData > volumeData)
[private]

Set the display properties of the volume and the rendering method.

5.7.3.9 void VolumeReconstructionWidget::setMainWindow (MainWindow * mainwindow)

Set the window to display the volume

5.7.3.10 void VolumeReconstructionWidget::setTransformStack (std::vector< vnl_matrix< double > > transformStack)

Set the transformation stack for the volume image

5.7.3.11 void VolumeReconstructionWidget::setVolumeColorMap (bool invert) [private]

Set the volume color transfer function.

5.7.3.12 void VolumeReconstructionWidget::setVolumeImageStack (std::vector< vtkSmartPointer< vtkImageData > > > volumeImageStack)

Set the image data stack

5.7.3.13 void VolumeReconstructionWidget::setVolumeOpacity (int point) [private]

Set the volume opacity.

5.7.3.14 void VolumeReconstructionWidget::transparency () [private, slot]

Changes the opacity of the volume.

5.7.4 Member Data Documentation

5.7.4.1 std::vector< vnl_vector<double> > VolumeReconstructionWidget::imageBoundsXStack [private]

Contains the transformed bounds in x of each image pixel

5.7.4.2 std::vector< vnl_vector<double> > VolumeReconstructionWidget::imageBoundsYStack [private]

Contains the transformed bounds in y of each image pixel

5.7.4.3 std::vector< vnl_vector<double> > VolumeReconstructionWidget::imageBoundsZStack [private]

Contains the transformed bounds in z of each image pixel

5.7.4.4 `std::vector< vnl_matrix<double> > VolumeReconstructionWidget::imageCoordsXStack` [private]

Contains the transformed coords in x of each image pixel

5.7.4.5 `std::vector< vnl_matrix<double> > VolumeReconstructionWidget::imageCoordsYStack` [private]

Contains the transformed coords in y of each image pixel

5.7.4.6 `std::vector< vnl_matrix<double> > VolumeReconstructionWidget::imageCoordsZStack` [private]

Contains the transformed coords in z of each image pixel

5.7.4.7 `bool VolumeReconstructionWidget::invertColors` [private]

5.7.4.8 `MainWindow* VolumeReconstructionWidget::mainWindow` [private]

the main window to call it

5.7.4.9 `vnl_vector<double> VolumeReconstructionWidget::scale` [private]

Scale of the images

5.7.4.10 `std::vector< vnl_matrix<double> > VolumeReconstructionWidget::transformStack` [private]

Contains the transformation of each image

5.7.4.11 `Ui::VolumeReconstructionWidget* VolumeReconstructionWidget::ui` [private]

5.7.4.12 `vtkSmartPointer<vtkVolume> VolumeReconstructionWidget::volume` [private]

the volume to display

5.7.4.13 `vtkSmartPointer<vtkImageData> VolumeReconstructionWidget::volumeData` [private]

Data of the volume

5.7.4.14 `vnl_vector<double> VolumeReconstructionWidget::volumeFinal` [private]

End of the volume data in the 3D space

5.7.4.15 `std::vector< vtkSmartPointer<vtkImageData> > VolumeReconstructionWidget::volumeImageStack`
[private]

The data of each image

5.7.4.16 `vnl_vector<double> VolumeReconstructionWidget::volumeOrigin` [private]

Start of the volume data in the 3D space

5.7.4.17 `vtkSmartPointer<vtkVolumeProperty> VolumeReconstructionWidget::volumeProperty` [private]

Main volume properties

5.7.4.18 `vnl_vector<double> VolumeReconstructionWidget::volumeSize` [private]

Size of the volume data

The documentation for this class was generated from the following files:

- [VolumeReconstructionWidget.h](#)
- [VolumeReconstructionWidget.cpp](#)

Chapter 6

File Documentation

6.1 Calibration.cpp File Reference

```
#include "Calibration.h"
```

6.2 Calibration.h File Reference

```
#include "SinglePointTargetUSCalibrationParametersEstimator.h"  
#include "RANSAC.h"
```

Classes

- class [Calibration](#)
Implements LSQRRecepies methods.

6.3 main.cpp File Reference

```
#include <QtGui/QApplication>  
#include "mainwindow.h"
```

Functions

- int [main](#) (int argc, char *argv[])

6.3.1 Function Documentation

6.3.1.1 int main (int *argc*, char * *argv*[])

6.4 mainwindow.cpp File Reference

```
#include "ui_mainwindow.h"
```

```
#include "mainwindow.h"
#include "ProbeCalibrationWidget.h"
#include "VolumeReconstructionWidget.h"
#include <QVBoxLayout>
#include <vtkEventQtSlotConnect.h>
```

6.5 mainwindow.h File Reference

```
#include <QMainWindow>
#include <QtGui>
#include "QVTKImageWidget.h"
#include <vtkEventQtSlotConnect.h>
```

Classes

- class [MainWindow](#)
Main window for user interaction.

Namespaces

- namespace [Ui](#)

6.6 ProbeCalibrationWidget.cpp File Reference

```
#include "ProbeCalibrationWidget.h"
#include "Calibration.h"
#include <QErrorMessage>
#include <QString>
#include <QFile>
#include <QTextStream>
#include <vtkExtractVOI.h>
#include <vn1/vn1_quaternion.h>
#include <vn1/vn1_vector_fixed.h>
#include <vn1/algo/vn1_levenberg_marquardt.h>
#include <vn1/vn1_double_2.h>
```

Variables

- bool [setCoordsSize](#) = false

6.6.1 Variable Documentation

6.6.1.1 bool setCoordsSize = false

6.7 ProbeCalibrationWidget.h File Reference

```
#include "ui_ProbeCalibrationWidget.h"
```

```
#include "mainwindow.h"
#include <QWidget>
#include <vtkSmartPointer.h>
#include <vtkImageData.h>
#include <vn1/vn1_matrix.h>
#include <string>
#include <fstream>
#include <stdio.h>
```

Classes

- class [ProbeCalibrationWidget](#)

Obtain data for calibration process.

6.8 QVTKImageWidget.cpp File Reference

```
#include "QVTKImageWidget.h"
#include "QVTKImageWidgetCommand.h"
#include <QSize.h>
#include <QBoxLayout>
#include <QString>
#include <itkImage.h>
#include <itkImageFileReader.h>
#include <vtkImageReader2.h>
#include <vtkImageReader2Factory.h>
#include <vtkCornerAnnotation.h>
#include <vtkPropPicker.h>
#include <vtkTextProperty.h>
#include <vtkImageActor.h>
#include <vtkInteractorStyleImage.h>
#include <vtkImageFlip.h>
#include <vtkVolume.h>
#include <vtkVolumeRayCastMapper.h>
#include <vtkVolumeRayCastCompositeFunction.h>
#include <vtkVolumeProperty.h>
#include <vtkColorTransferFunction.h>
#include <vtkPiecewiseFunction.h>
#include <vtkMath.h>
```

6.9 QVTKImageWidget.h File Reference

```
#include <QtGui>
```

```
#include <QWidget>
#include <QVTKWidget.h>
#include <itkImage.h>
#include <itkRGBPixel.h>
#include <vtkSmartPointer.h>
#include <vtkImageData.h>
#include <vtkRenderWindow.h>
#include <vtkRenderer.h>
#include <vtkCamera.h>
#include <vtkImageActor.h>
#include <vtkCommand.h>
#include <vtkImageViewer2.h>
#include <vtkCornerAnnotation.h>
#include <vtkTransform.h>
#include <vn1/vn1_quaternion.h>
#include <vn1/vn1_matrix.h>
#include <vn1/vn1_vector.h>
```

Classes

- class [QVTKImageWidget](#)
Display VTK images.

Typedefs

- typedef itk::RGBPixel
 < unsigned char > [RGBPixelFormat](#)
- typedef itk::Image< unsigned char > [ImageType](#)
- typedef itk::Image
 < [RGBPixelFormat](#), 2 > [RGBImageType](#)

6.9.1 Typedef Documentation

6.9.1.1 typedef itk::Image< unsigned char > [ImageType](#)

6.9.1.2 typedef itk::Image< [RGBPixelFormat](#), 2> [RGBImageType](#)

6.9.1.3 typedef itk::RGBPixel< unsigned char > [RGBPixelFormat](#)

6.10 QVTKImageWidgetCommand.cpp File Reference

```
#include "QVTKImageWidgetCommand.h"
#include <vtkImageActor.h>
#include <vtkImageData.h>
#include <vtkInteractorStyleImage.h>
#include <vtkRenderWindow.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkVariant.h>
#include <vtkMath.h>
#include <vtkCommand.h>
#include <vtkImageViewer2.h>
```


6.11 QVTKImageWidgetCommand.h File Reference

```
#include "QVTKImageWidget.h"  
#include <vtkCommand.h>  
#include <vtkPropPicker.h>  
#include <vtkCornerAnnotation.h>  
#include <vtkSmartPointer.h>
```

Classes

- class [QVTKImageWidgetCommand](#)

Interaction with mouse.

6.12 VolumeReconstruction.cpp File Reference

```
#include "VolumeReconstruction.h"  
#include <vtkMath.h>  
#include <vn1/vn1_inverse.h>
```

6.13 VolumeReconstruction.h File Reference

```
#include <vtkSmartPointer.h>  
#include <vtkImageData.h>  
#include <vtkPlane.h>  
#include <vn1/vn1_matrix.h>  
#include <vn1/vn1_vector.h>  
#include <math.h>  
#include <vector>
```

Classes

- class [VolumeReconstruction](#)

Generate a new volume.

6.14 VolumeReconstructionWidget.cpp File Reference

```
#include "VolumeReconstructionWidget.h"  
#include "ui_VolumeReconstructionWidget.h"  
#include "VolumeReconstruction.h"  
#include "vtkMetaImageWriter.h"  
#include <QString>
```

6.15 VolumeReconstructionWidget.h File Reference

```
#include <QWidget>
```

```
#include "mainwindow.h"
#include <vtkSmartPointer.h>
#include <vtkImageData.h>
#include <vtkVolume.h>
#include <vtkVolumeRayCastMapper.h>
#include <vtkVolumeRayCastCompositeFunction.h>
#include <vtkVolumeProperty.h>
#include <vtkColorTransferFunction.h>
#include <vtkPiecewiseFunction.h>
#include <vtkMath.h>
#include <vn1/vn1_matrix.h>
#include <vn1/vn1_vector.h>
```

Classes

- class [VolumeReconstructionWidget](#)
Has the interaction methods for the user to generate a new volue.

Namespaces

- namespace [Ui](#)

Index

- ~MainWindow
 - MainWindow, [12](#)
- ~ProbeCalibrationWidget
 - ProbeCalibrationWidget, [15](#)
- ~QVTKImageWidget
 - QVTKImageWidget, [19](#)
- ~QVTKImageWidgetCommand
 - QVTKImageWidgetCommand, [25](#)
- ~VolumeReconstructionWidget
 - VolumeReconstructionWidget, [31](#)
- addImages
 - MainWindow, [12](#)
- addLogText
 - MainWindow, [12](#)
- Annotation
 - QVTKImageWidgetCommand, [26](#)
- calcImageBounds
 - VolumeReconstructionWidget, [31](#)
- calcImageCoords
 - VolumeReconstructionWidget, [31](#)
- calcImagePlane
 - VolumeReconstruction, [27](#)
- calcMaxDistance
 - VolumeReconstruction, [27](#)
- calcVolumeSize
 - VolumeReconstructionWidget, [31](#)
- calcVoxelValue
 - VolumeReconstruction, [28](#)
- Calibrate
 - Calibration, [10](#)
- calibrate
 - ProbeCalibrationWidget, [15](#)
- Calibration, [9](#)
 - Calibrate, [10](#)
 - ClearImagePoints, [10](#)
 - ClearTransformations, [10](#)
 - data, [10](#)
 - DataType, [10](#)
 - estimatedUSCalibrationParameters, [10](#)
 - getEstimatedUSCalibrationParameters, [10](#)
 - imagePoints, [10](#)
 - InsertImagePoints, [10](#)
 - InsertTransformations, [10](#)
 - New, [10](#)
 - transformations, [10](#)
- Calibration.cpp, [35](#)
- Calibration.h, [35](#)
- calibrationParameters
 - ProbeCalibrationWidget, [16](#)
- ClearImagePoints
 - Calibration, [10](#)
- ClearTransformations
 - Calibration, [10](#)
- computeTransformation
 - QVTKImageWidget, [19](#)
- Connections
 - MainWindow, [12](#)
- coords
 - ProbeCalibrationWidget, [16](#)
- cornerAnnotation
 - QVTKImageWidget, [22](#)
- crop
 - ProbeCalibrationWidget, [15](#)
- cropProbeImage
 - ProbeCalibrationWidget, [15](#)
- data
 - Calibration, [10](#)
- DataType
 - Calibration, [10](#)
- displayImage
 - QVTKImageWidget, [19](#)
- displaySelectedImage
 - MainWindow, [12](#)
 - QVTKImageWidget, [19](#)
- displayVolume
 - QVTKImageWidget, [19](#)
 - VolumeReconstructionWidget, [31](#)
- displayVolumeImages
 - QVTKImageWidget, [19](#)
- displayWidget
 - MainWindow, [12](#)
- estimatedUSCalibrationParameters
 - Calibration, [10](#)
- Execute
 - QVTKImageWidgetCommand, [26](#)
- generate
 - VolumeReconstructionWidget, [31](#)
- generateVolume
 - VolumeReconstruction, [28](#)
- getCoordinates
 - ProbeCalibrationWidget, [15](#)
- getDisplayWidget
 - MainWindow, [12](#)
- getEstimatedUSCalibrationParameters
 - Calibration, [10](#)

- getImageDisplayedIndex
 - QVTKImageWidget, 20
- getImageHeigth
 - QVTKImageWidget, 20
- getImageSize
 - QVTKImageWidget, 20
- getImageStack
 - QVTKImageWidget, 20
- getImageType
 - QVTKImageWidget, 20
- getImageViewer
 - QVTKImageWidget, 20
- getImageWidth
 - QVTKImageWidget, 20
- getNumOfDimesions
 - QVTKImageWidget, 20
- getPixelType
 - QVTKImageWidget, 20
- getQVTKWidget
 - QVTKImageWidget, 20
- getTransformScale
 - QVTKImageWidget, 21
- getTransformStack
 - QVTKImageWidget, 21
- getVolumImageStack
 - QVTKImageWidget, 21
- getXPicked
 - QVTKImageWidget, 21
- getYPicked
 - QVTKImageWidget, 21
- image
 - ProbeCalibrationWidget, 16
- imageBoundsXStack
 - VolumeReconstruction, 28
 - VolumeReconstructionWidget, 32
- imageBoundsYStack
 - VolumeReconstruction, 28
 - VolumeReconstructionWidget, 32
- imageBoundsZStack
 - VolumeReconstruction, 29
 - VolumeReconstructionWidget, 32
- imageCoordsXStack
 - VolumeReconstructionWidget, 32
- imageCoordsYStack
 - VolumeReconstructionWidget, 33
- imageCoordsZStack
 - VolumeReconstructionWidget, 33
- imageDisplayedIndex
 - QVTKImageWidget, 22
- imageHeight
 - QVTKImageWidget, 22
- imagePlaneStack
 - VolumeReconstruction, 29
- imagePoints
 - Calibration, 10
- imageStack
 - ProbeCalibrationWidget, 16
 - QVTKImageWidget, 23
- ImageType
 - QVTKImageWidget.h, 38
- imageType
 - QVTKImageWidget, 23
- imageViewer
 - QVTKImageWidget, 23
- ImageWidget
 - QVTKImageWidgetCommand, 26
- imageWidth
 - QVTKImageWidget, 23
- imagesFileNames
 - MainWindow, 13
- InsertImagePoints
 - Calibration, 10
- InsertTransformations
 - Calibration, 10
- invert
 - VolumeReconstructionWidget, 31
- invertColors
 - VolumeReconstructionWidget, 33
- isImageStackLoaded
 - QVTKImageWidget, 23
- isVolumImageStackLoaded
 - QVTKImageWidget, 23
- itkImage
 - QVTKImageWidget, 23
- loadRotationsFile
 - ProbeCalibrationWidget, 15
- loadTranslationsFile
 - ProbeCalibrationWidget, 15
- main
 - main.cpp, 35
- main.cpp, 35
 - main, 35
- MainWindow, 11
 - ~MainWindow, 12
 - addImages, 12
 - addLogText, 12
 - Connections, 12
 - displaySelectedImage, 12
 - displayWidget, 12
 - getDisplayWidget, 12
 - imagesFileNames, 13
 - MainWindow, 12
 - MainWindow, 12
 - openVolumeData, 12
 - print, 12
 - probeCalibration, 12
 - textOnTextArea, 13
 - ui, 13
 - volumeCalibrationData, 13
 - volumImagesFileNames, 13
 - volumeReconstruction, 12
 - volumeRotationData, 13
 - volumeTranslationData, 13
- mainWindow
 - ProbeCalibrationWidget, 16

- VolumeReconstructionWidget, 33
- mainwindow.cpp, 35
- mainwindow.h, 36
- maxDistance
 - VolumeReconstruction, 29
- New
 - Calibration, 10
 - QVTKImageWidgetCommand, 26
 - VolumeReconstruction, 28
- numDimensions
 - QVTKImageWidget, 23
- openVolumeData
 - MainWindow, 12
- Picker
 - QVTKImageWidgetCommand, 26
- pixelType
 - QVTKImageWidget, 23
- print
 - MainWindow, 12
- probeCalibration
 - MainWindow, 12
- ProbeCalibrationWidget, 13
 - ~ProbeCalibrationWidget, 15
 - calibrate, 15
 - calibrationParameters, 16
 - coords, 16
 - crop, 15
 - cropProbelImage, 15
 - getCoordinates, 15
 - image, 16
 - imageStack, 16
 - loadRotationsFile, 15
 - loadTranslationsFile, 15
 - mainWindow, 16
 - ProbeCalibrationWidget, 15
 - ProbeCalibrationWidget, 15
 - rotations, 16
 - rotations_2, 16
 - saveCalibration, 15
 - setImage, 15
 - setImageStack, 15
 - setMainWindow, 16
 - translations, 16
 - workWithStack, 16
- ProbeCalibrationWidget.cpp, 36
 - setCoordsSize, 36
- ProbeCalibrationWidget.h, 36
- QVTKImageWidget, 17
 - ~QVTKImageWidget, 19
 - computeTransformation, 19
 - cornerAnnotation, 22
 - displayImage, 19
 - displaySelectedImage, 19
 - displayVolume, 19
 - displayVolumelImages, 19
 - getImageDisplayedIndex, 20
 - getImageHeigth, 20
 - getImageSize, 20
 - getImageStack, 20
 - getImageType, 20
 - getImageViewer, 20
 - getImageWidth, 20
 - getNumOfDimesions, 20
 - getPixelType, 20
 - getQVTKWidget, 20
 - getTransformScale, 21
 - getTransformStack, 21
 - getVolumelImageStack, 21
 - getXPicked, 21
 - getYPicked, 21
 - imageDisplayedIndex, 22
 - imageHeight, 22
 - imageStack, 23
 - imageType, 23
 - imageView, 23
 - imageWidth, 23
 - isImageStackLoaded, 23
 - isVolumelImageStackLoaded, 23
 - itkImage, 23
 - numDimensions, 23
 - pixelType, 23
 - QVTKImageWidget, 19
 - qvtkWidget, 23
 - QVTKImageWidget, 19
 - renderer, 23
 - renwin, 23
 - rgbltkImage, 24
 - scale, 24
 - setAndDisplayImage, 21
 - setAndDisplayMultipleImages, 21
 - setAndDisplayVolumelImages, 22
 - setImageProperties, 22
 - setXPicked, 22
 - setYPicked, 22
 - transformStack, 24
 - volumeDataCalibration, 24
 - volumeDataRotations, 24
 - volumeDataTranslations, 24
 - volumelImageActorStack, 24
 - volumelImageStack, 24
 - vtkImage, 24
 - xPicked, 24
 - xPosition, 24
 - yPicked, 24
 - yPosition, 25
- QVTKImageWidget.cpp, 37
- QVTKImageWidget.h, 37
 - ImageType, 38
 - RGBImageType, 38
 - RGBPixelType, 38
- QVTKImageWidgetCommand, 25
 - ~QVTKImageWidgetCommand, 25
 - Annotation, 26

- Execute, 26
- ImageWidget, 26
- New, 26
- Picker, 26
- QVTKImageWidgetCommand, 25
- QVTKImageWidgetCommand, 25
- SetAnnotation, 26
- SetImageWidget, 26
- SetPicker, 26
- QVTKImageWidgetCommand.cpp, 38
- QVTKImageWidgetCommand.h, 39
- qvtkWidget
 - QVTKImageWidget, 23
- RGBImageType
 - QVTKImageWidget.h, 38
- RGBPixelFormat
 - QVTKImageWidget.h, 38
- renderer
 - QVTKImageWidget, 23
- renwin
 - QVTKImageWidget, 23
- rgbvtkImage
 - QVTKImageWidget, 24
- rotations
 - ProbeCalibrationWidget, 16
- rotations_2
 - ProbeCalibrationWidget, 16
- save
 - VolumeReconstructionWidget, 31
- saveCalibration
 - ProbeCalibrationWidget, 15
- scale
 - QVTKImageWidget, 24
 - VolumeReconstruction, 29
 - VolumeReconstructionWidget, 33
- setAndDisplayImage
 - QVTKImageWidget, 21
- setAndDisplayMultipleImages
 - QVTKImageWidget, 21
- setAndDisplayVolumeImages
 - QVTKImageWidget, 22
- SetAnnotation
 - QVTKImageWidgetCommand, 26
- setCoordsSize
 - ProbeCalibrationWidget.cpp, 36
- setDisplayProperties
 - VolumeReconstructionWidget, 32
- setImage
 - ProbeCalibrationWidget, 15
- setImageBoundsStack
 - VolumeReconstruction, 28
- setImageProperties
 - QVTKImageWidget, 22
- setImageStack
 - ProbeCalibrationWidget, 15
- SetImageWidget
 - QVTKImageWidgetCommand, 26
- setMainWindow
 - ProbeCalibrationWidget, 16
 - VolumeReconstructionWidget, 32
- SetPicker
 - QVTKImageWidgetCommand, 26
- setScale
 - VolumeReconstruction, 28
- setTransformStack
 - VolumeReconstruction, 28
 - VolumeReconstructionWidget, 32
- setVolumeColorMap
 - VolumeReconstructionWidget, 32
- setVolumeImageStack
 - VolumeReconstruction, 28
 - VolumeReconstructionWidget, 32
- setVolumeOpacity
 - VolumeReconstructionWidget, 32
- setVolumeOrigin
 - VolumeReconstruction, 28
- setVolumeSize
 - VolumeReconstruction, 28
- setXPicked
 - QVTKImageWidget, 22
- setYPicked
 - QVTKImageWidget, 22
- textOnTextArea
 - MainWindow, 13
- transformStack
 - QVTKImageWidget, 24
 - VolumeReconstruction, 29
 - VolumeReconstructionWidget, 33
- transformations
 - Calibration, 10
- translations
 - ProbeCalibrationWidget, 16
- transparency
 - VolumeReconstructionWidget, 32
- Ui, 7
- ui
 - MainWindow, 13
 - VolumeReconstructionWidget, 33
- volume
 - VolumeReconstructionWidget, 33
- volumeCalibrationData
 - MainWindow, 13
- volumeData
 - VolumeReconstructionWidget, 33
- volumeDataCalibration
 - QVTKImageWidget, 24
- volumeDataRotations
 - QVTKImageWidget, 24
- volumeDataTranslations
 - QVTKImageWidget, 24
- volumeFinal
 - VolumeReconstructionWidget, 33
- volumeImageActorStack

- QVTKImageWidget, 24
- volumeImageStack
 - QVTKImageWidget, 24
 - VolumeReconstruction, 29
 - VolumeReconstructionWidget, 33
- volumeImagesFileNames
 - MainWindow, 13
- volumeOrigin
 - VolumeReconstruction, 29
 - VolumeReconstructionWidget, 33
- volumeProperty
 - VolumeReconstructionWidget, 33
- VolumeReconstruction, 26
 - calcImagePlane, 27
 - calcMaxDistance, 27
 - calcVoxelValue, 28
 - generateVolume, 28
 - imageBoundsXStack, 28
 - imageBoundsYStack, 28
 - imageBoundsZStack, 29
 - imagePlaneStack, 29
 - maxDistance, 29
 - New, 28
 - scale, 29
 - setImageBoundsStack, 28
 - setScale, 28
 - setTransformStack, 28
 - setVolumeImageStack, 28
 - setVolumeOrigin, 28
 - setVolumeSize, 28
 - transformStack, 29
 - volumeImageStack, 29
 - volumeOrigin, 29
 - volumeSize, 29
- volumeReconstruction
 - MainWindow, 12
- VolumeReconstruction.cpp, 39
- VolumeReconstruction.h, 39
- VolumeReconstructionWidget, 29
 - ~VolumeReconstructionWidget, 31
 - calcImageBounds, 31
 - calcImageCoords, 31
 - calcVolumeSize, 31
 - displayVolume, 31
 - generate, 31
 - imageBoundsXStack, 32
 - imageBoundsYStack, 32
 - imageBoundsZStack, 32
 - imageCoordsXStack, 32
 - imageCoordsYStack, 33
 - imageCoordsZStack, 33
 - invert, 31
 - invertColors, 33
 - mainWindow, 33
 - save, 31
 - scale, 33
 - setDisplayProperties, 32
 - setMainWindow, 32
 - setTransformStack, 32
 - setVolumeColorMap, 32
 - setVolumeImageStack, 32
 - setVolumeOpacity, 32
 - transformStack, 33
 - transparency, 32
 - ui, 33
 - volume, 33
 - volumeData, 33
 - volumeFinal, 33
 - volumeImageStack, 33
 - volumeOrigin, 33
 - volumeProperty, 33
 - VolumeReconstructionWidget, 31
 - volumeSize, 34
 - VolumeReconstructionWidget, 31
- VolumeReconstructionWidget.cpp, 39
- VolumeReconstructionWidget.h, 39
- volumeRotationData
 - MainWindow, 13
- volumeSize
 - VolumeReconstruction, 29
 - VolumeReconstructionWidget, 34
- volumeTranslationData
 - MainWindow, 13
- vtkImage
 - QVTKImageWidget, 24
- workWithStack
 - ProbeCalibrationWidget, 16
- xPicked
 - QVTKImageWidget, 24
- xPosition
 - QVTKImageWidget, 24
- yPicked
 - QVTKImageWidget, 24
- yPosition
 - QVTKImageWidget, 25