

Volume Reconstruction

1.0

Generated by Doxygen 1.8.0

Mon Jul 23 2012 15:08:35

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 | CroplImagesWidget Class Reference | 11 |
| 5.2.1 | Detailed Description | 12 |
| 5.2.2 | Constructor & Destructor Documentation | 12 |

| | | |
|----------|--|----|
| 5.2.2.1 | CropImagesWidget | 12 |
| 5.2.2.2 | ~CropImagesWidget | 12 |
| 5.2.3 | Member Function Documentation | 12 |
| 5.2.3.1 | crop | 12 |
| 5.2.3.2 | cropProbImage | 12 |
| 5.2.3.3 | save | 12 |
| 5.2.3.4 | setImage | 12 |
| 5.2.3.5 | setImageStack | 12 |
| 5.2.3.6 | setMainWindow | 12 |
| 5.2.4 | Member Data Documentation | 12 |
| 5.2.4.1 | cropImage | 13 |
| 5.2.4.2 | cropStack | 13 |
| 5.2.4.3 | image | 13 |
| 5.2.4.4 | imageStack | 13 |
| 5.2.4.5 | mainWindow | 13 |
| 5.2.4.6 | ui | 13 |
| 5.2.4.7 | workWithStack | 13 |
| 5.3 | MainWindow Class Reference | 13 |
| 5.3.1 | Detailed Description | 14 |
| 5.3.2 | Constructor & Destructor Documentation | 14 |
| 5.3.2.1 | MainWindow | 14 |
| 5.3.2.2 | ~MainWindow | 14 |
| 5.3.3 | Member Function Documentation | 14 |
| 5.3.3.1 | addImages | 14 |
| 5.3.3.2 | addLogText | 15 |
| 5.3.3.3 | cropImages | 15 |
| 5.3.3.4 | displaySelectedImage | 15 |
| 5.3.3.5 | getDisplayWidget | 15 |
| 5.3.3.6 | openVolume | 15 |
| 5.3.3.7 | openVolumeData | 15 |
| 5.3.3.8 | print | 15 |
| 5.3.3.9 | probeCalibration | 15 |
| 5.3.3.10 | setSelectedOpacity | 15 |
| 5.3.3.11 | volumeReconstruction | 15 |
| 5.3.4 | Member Data Documentation | 15 |
| 5.3.4.1 | Connections | 15 |
| 5.3.4.2 | displayWidget | 15 |
| 5.3.4.3 | imagesFileNames | 15 |
| 5.3.4.4 | textOnTextArea | 16 |
| 5.3.4.5 | ui | 16 |

| | | |
|----------|--|----|
| 5.3.4.6 | volumeCalibrationData | 16 |
| 5.3.4.7 | volumeFilename | 16 |
| 5.3.4.8 | volumeImagesFileNames | 16 |
| 5.3.4.9 | volumeRotationData | 16 |
| 5.3.4.10 | volumeTranslationData | 16 |
| 5.4 | ProbeCalibrationWidget Class Reference | 16 |
| 5.4.1 | Detailed Description | 17 |
| 5.4.2 | Constructor & Destructor Documentation | 17 |
| 5.4.2.1 | ProbeCalibrationWidget | 17 |
| 5.4.2.2 | ~ProbeCalibrationWidget | 18 |
| 5.4.3 | Member Function Documentation | 18 |
| 5.4.3.1 | calibrate | 18 |
| 5.4.3.2 | crop | 18 |
| 5.4.3.3 | cropProbeImage | 18 |
| 5.4.3.4 | getCoordinates | 18 |
| 5.4.3.5 | loadRotationsFile | 18 |
| 5.4.3.6 | loadTranslationsFile | 18 |
| 5.4.3.7 | saveCalibration | 18 |
| 5.4.3.8 | setImage | 18 |
| 5.4.3.9 | setImageStack | 18 |
| 5.4.3.10 | setMainWindow | 19 |
| 5.4.4 | Member Data Documentation | 19 |
| 5.4.4.1 | calibrationParameters | 19 |
| 5.4.4.2 | coords | 19 |
| 5.4.4.3 | image | 19 |
| 5.4.4.4 | imageStack | 19 |
| 5.4.4.5 | mainWindow | 19 |
| 5.4.4.6 | rotations | 19 |
| 5.4.4.7 | rotations_2 | 19 |
| 5.4.4.8 | translations | 19 |
| 5.4.4.9 | workWithStack | 19 |
| 5.5 | QVTKImageWidget Class Reference | 19 |
| 5.5.1 | Detailed Description | 22 |
| 5.5.2 | Constructor & Destructor Documentation | 22 |
| 5.5.2.1 | QVTKImageWidget | 22 |
| 5.5.2.2 | ~QVTKImageWidget | 22 |
| 5.5.3 | Member Function Documentation | 22 |
| 5.5.3.1 | computeTransformation | 22 |
| 5.5.3.2 | displayImage | 22 |
| 5.5.3.3 | displaySelectedImage | 23 |

| | | |
|----------|---|----|
| 5.5.3.4 | displayVolume | 23 |
| 5.5.3.5 | displayVolumeImages | 23 |
| 5.5.3.6 | getImageDisplayedIndex | 23 |
| 5.5.3.7 | getImageHeight | 23 |
| 5.5.3.8 | getImageSize | 23 |
| 5.5.3.9 | getImageStack | 23 |
| 5.5.3.10 | getImageType | 23 |
| 5.5.3.11 | getImageViewer | 23 |
| 5.5.3.12 | getImageWidth | 23 |
| 5.5.3.13 | getNumOfDimensions | 24 |
| 5.5.3.14 | getPixelType | 24 |
| 5.5.3.15 | getQVTKWidget | 24 |
| 5.5.3.16 | getTransformScale | 24 |
| 5.5.3.17 | getTransformStack | 24 |
| 5.5.3.18 | getVolumeImageStack | 24 |
| 5.5.3.19 | getXPicked | 24 |
| 5.5.3.20 | getYPicked | 24 |
| 5.5.3.21 | setAndDisplayImage | 24 |
| 5.5.3.22 | setAndDisplayImage | 24 |
| 5.5.3.23 | setAndDisplayMultipleImages | 25 |
| 5.5.3.24 | setAndDisplayMultipleImages | 25 |
| 5.5.3.25 | setAndDisplayVolume | 25 |
| 5.5.3.26 | setAndDisplayVolume | 25 |
| 5.5.3.27 | setAndDisplayVolumeImages | 25 |
| 5.5.3.28 | setImageProperties | 26 |
| 5.5.3.29 | setVolumeData | 26 |
| 5.5.3.30 | setVolumeOpacity | 26 |
| 5.5.3.31 | setVolumeOrigin | 26 |
| 5.5.3.32 | setXPicked | 26 |
| 5.5.3.33 | setYPicked | 26 |
| 5.5.4 | Member Data Documentation | 26 |
| 5.5.4.1 | cornerAnnotation | 26 |
| 5.5.4.2 | imageDisplayedIndex | 26 |
| 5.5.4.3 | imageHeight | 26 |
| 5.5.4.4 | imageStack | 26 |
| 5.5.4.5 | imageType | 26 |
| 5.5.4.6 | imageView | 26 |
| 5.5.4.7 | imageWidth | 27 |
| 5.5.4.8 | isImageStackLoaded | 27 |
| 5.5.4.9 | isVolumeImageStackLoaded | 27 |

| | | |
|----------|--|----|
| 5.5.4.10 | itkImage | 27 |
| 5.5.4.11 | numDimensions | 27 |
| 5.5.4.12 | opacityPoint | 27 |
| 5.5.4.13 | pixelType | 27 |
| 5.5.4.14 | qvtkWidget | 27 |
| 5.5.4.15 | renderer | 27 |
| 5.5.4.16 | renwin | 27 |
| 5.5.4.17 | rgbItkImage | 27 |
| 5.5.4.18 | scale | 27 |
| 5.5.4.19 | transformStack | 27 |
| 5.5.4.20 | volume | 28 |
| 5.5.4.21 | volumeData | 28 |
| 5.5.4.22 | volumeDataCalibration | 28 |
| 5.5.4.23 | volumeDataRotations | 28 |
| 5.5.4.24 | volumeDataTranslations | 28 |
| 5.5.4.25 | volumeImageActorStack | 28 |
| 5.5.4.26 | volumeImageStack | 28 |
| 5.5.4.27 | volumeProperty | 28 |
| 5.5.4.28 | volumeScalarOpacity | 28 |
| 5.5.4.29 | vtkImage | 28 |
| 5.5.4.30 | xPicked | 28 |
| 5.5.4.31 | xPosition | 28 |
| 5.5.4.32 | yPicked | 29 |
| 5.5.4.33 | yPosition | 29 |
| 5.6 | QVTKImageWidgetCommand Class Reference | 29 |
| 5.6.1 | Detailed Description | 29 |
| 5.6.2 | Constructor & Destructor Documentation | 29 |
| 5.6.2.1 | QVTKImageWidgetCommand | 29 |
| 5.6.2.2 | ~QVTKImageWidgetCommand | 29 |
| 5.6.3 | Member Function Documentation | 30 |
| 5.6.3.1 | Execute | 30 |
| 5.6.3.2 | New | 30 |
| 5.6.3.3 | SetAnnotation | 30 |
| 5.6.3.4 | SetImageWidget | 30 |
| 5.6.3.5 | SetPicker | 30 |
| 5.6.4 | Member Data Documentation | 30 |
| 5.6.4.1 | Annotation | 30 |
| 5.6.4.2 | ImageWidget | 30 |
| 5.6.4.3 | Picker | 30 |
| 5.7 | VolumeReconstruction Class Reference | 30 |

| | | |
|----------|--|----|
| 5.7.1 | Detailed Description | 31 |
| 5.7.2 | Member Function Documentation | 31 |
| 5.7.2.1 | calcImagePlane | 31 |
| 5.7.2.2 | calcMaxDistance | 32 |
| 5.7.2.3 | calcVoxelValue | 32 |
| 5.7.2.4 | generateVolume | 32 |
| 5.7.2.5 | New | 32 |
| 5.7.2.6 | setImageBoundsStack | 32 |
| 5.7.2.7 | setScale | 32 |
| 5.7.2.8 | setTransformStack | 32 |
| 5.7.2.9 | setVolumeImageStack | 32 |
| 5.7.2.10 | setVolumeOrigin | 32 |
| 5.7.2.11 | setVolumeSize | 32 |
| 5.7.3 | Member Data Documentation | 32 |
| 5.7.3.1 | imageBoundsXStack | 32 |
| 5.7.3.2 | imageBoundsYStack | 33 |
| 5.7.3.3 | imageBoundsZStack | 33 |
| 5.7.3.4 | imagePlaneStack | 33 |
| 5.7.3.5 | maxDistance | 33 |
| 5.7.3.6 | scale | 33 |
| 5.7.3.7 | transformStack | 33 |
| 5.7.3.8 | volumeImageStack | 33 |
| 5.7.3.9 | volumeOrigin | 33 |
| 5.7.3.10 | volumeSize | 33 |
| 5.8 | VolumeReconstructionWidget Class Reference | 33 |
| 5.8.1 | Detailed Description | 35 |
| 5.8.2 | Constructor & Destructor Documentation | 35 |
| 5.8.2.1 | VolumeReconstructionWidget | 35 |
| 5.8.2.2 | ~VolumeReconstructionWidget | 35 |
| 5.8.3 | Member Function Documentation | 35 |
| 5.8.3.1 | calcImageBounds | 35 |
| 5.8.3.2 | calcImageCoords | 35 |
| 5.8.3.3 | calcVolumeSize | 35 |
| 5.8.3.4 | displayVolume | 35 |
| 5.8.3.5 | generate | 35 |
| 5.8.3.6 | save | 35 |
| 5.8.3.7 | setDisplayProperties | 35 |
| 5.8.3.8 | setMainWindow | 35 |
| 5.8.3.9 | setTransformStack | 36 |
| 5.8.3.10 | setVolumeColorMap | 36 |

| | | |
|----------|---|-----------|
| 5.8.3.11 | setVolumeImageStack | 36 |
| 5.8.3.12 | setVolumeOpacity | 36 |
| 5.8.4 | Member Data Documentation | 36 |
| 5.8.4.1 | imageBoundsXStack | 36 |
| 5.8.4.2 | imageBoundsYStack | 36 |
| 5.8.4.3 | imageBoundsZStack | 36 |
| 5.8.4.4 | imageCoordsXStack | 36 |
| 5.8.4.5 | imageCoordsYStack | 36 |
| 5.8.4.6 | imageCoordsZStack | 36 |
| 5.8.4.7 | mainWindow | 36 |
| 5.8.4.8 | scale | 37 |
| 5.8.4.9 | transformStack | 37 |
| 5.8.4.10 | ui | 37 |
| 5.8.4.11 | volume | 37 |
| 5.8.4.12 | volumeData | 37 |
| 5.8.4.13 | volumeFinal | 37 |
| 5.8.4.14 | volumeImageStack | 37 |
| 5.8.4.15 | volumeOrigin | 37 |
| 5.8.4.16 | volumeProperty | 37 |
| 5.8.4.17 | volumeSize | 37 |
| 6 | File Documentation | 39 |
| 6.1 | Calibration.cpp File Reference | 39 |
| 6.2 | Calibration.h File Reference | 39 |
| 6.3 | CropImagesWidget.cpp File Reference | 39 |
| 6.4 | CropImagesWidget.h File Reference | 39 |
| 6.5 | main.cpp File Reference | 40 |
| 6.5.1 | Function Documentation | 40 |
| 6.5.1.1 | main | 40 |
| 6.6 | mainwindow.cpp File Reference | 40 |
| 6.7 | mainwindow.h File Reference | 40 |
| 6.8 | ProbeCalibrationWidget.cpp File Reference | 41 |
| 6.8.1 | Variable Documentation | 41 |
| 6.8.1.1 | setCoordsSize | 41 |
| 6.9 | ProbeCalibrationWidget.h File Reference | 41 |
| 6.10 | QVTKImageWidget.cpp File Reference | 41 |
| 6.11 | QVTKImageWidget.h File Reference | 42 |
| 6.11.1 | Typedef Documentation | 43 |
| 6.11.1.1 | ImageType | 43 |
| 6.11.1.2 | RGBImageType | 43 |

| | |
|--|----|
| 6.11.1.3 RGBPixelFormat | 43 |
| 6.12 QVTKImageWidgetCommand.cpp File Reference | 43 |
| 6.13 QVTKImageWidgetCommand.h File Reference | 43 |
| 6.14 VolumeReconstruction.cpp File Reference | 43 |
| 6.15 VolumeReconstruction.h File Reference | 44 |
| 6.16 VolumeReconstructionWidget.cpp File Reference | 44 |
| 6.17 VolumeReconstructionWidget.h File Reference | 44 |

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 |
| CropImagesWidget | |
| Crop images | 11 |
| MainWindow | |
| Main window for user interaction | 13 |
| ProbeCalibrationWidget | |
| Obtain data for calibration process | 16 |
| QVTKImageWidget | |
| Display VTK images | 19 |
| QVTKImageWidgetCommand | |
| Interaction with mouse | 29 |
| VolumeReconstruction | |
| Generate a new volume | 30 |
| VolumeReconstructionWidget | |
| Has the interaction methods for the user to generate a new volue | 33 |

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

| | |
|--------------------------------|----|
| Calibration.cpp | 39 |
| Calibration.h | 39 |
| CropImagesWidget.cpp | 39 |
| CropImagesWidget.h | 39 |
| main.cpp | 40 |
| mainwindow.cpp | 40 |
| mainwindow.h | 40 |
| ProbeCalibrationWidget.cpp | 41 |
| ProbeCalibrationWidget.h | 41 |
| QVTKImageWidget.cpp | 41 |
| QVTKImageWidget.h | 42 |
| QVTKImageWidgetCommand.cpp | 43 |
| QVTKImageWidgetCommand.h | 43 |
| VolumeReconstruction.cpp | 43 |
| VolumeReconstruction.h | 44 |
| VolumeReconstructionWidget.cpp | 44 |
| VolumeReconstructionWidget.h | 44 |

Chapter 4

Namespace Documentation

4.1 Ui Namespace Reference

Chapter 5

Class Documentation

5.1 Calibration Class Reference

Implements LSQRRecipes 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 lsqrRecipes::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<lsqrRecipes::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 CropImagesWidget Class Reference

Crop images.

```
#include <CropImagesWidget.h>
```

Public Member Functions

- [CropImagesWidget](#) (QWidget *parent=0)
- [~CropImagesWidget](#) ()
- void [setImageStack](#) (std::vector< vtkSmartPointer< vtkImageData > > [imageStack](#))
Set this stack of vtkImageData.
- void [setImage](#) (vtkSmartPointer< vtkImageData > [image](#))
Set this vtkImageData.
- void [setMainWindow](#) (MainWindow *mainwindow)
Set the window to display the crop images.

Private Slots

- void [crop](#) ()
calls the crop method when the crop button is clicked
- void [save](#) ()
Save the cropped images in a folder.

Private Member Functions

- vtkSmartPointer< vtkImageData > [cropProbeImage](#) (vtkSmartPointer< vtkImageData > [image](#), int depth-Type)
Crop ultrasound image depending of the depth type.

Private Attributes

- Ui::CropImagesWidget * [ui](#)
- bool [workWithStack](#)
if there are multiple images to work with
- std::vector< vtkSmartPointer
< vtkImageData > > [imageStack](#)
an Array of vtkImageData to work
- MainWindow * [mainWindow](#)
- vtkSmartPointer< vtkImageData > [image](#)
the vtkImageData to work
- vtkSmartPointer< vtkImageData > [croppedImage](#)
the cropped image
- std::vector< vtkSmartPointer
< vtkImageData > > [cropStack](#)
the cropped imageStack

5.2.1 Detailed Description

Crop images.

This class crop one or multiple images deppendng on which range of the ultrasound machine was used. It can crop images in ranges 4, 5, 6 and 8. Images can be saved in a folder or used to calibrate the US probe.

5.2.2 Constructor & Destructor Documentation

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

5.2.2.2 `CroplImagesWidget::~~CroplImagesWidget ()`

5.2.3 Member Function Documentation

5.2.3.1 `void CroplImagesWidget::crop ()` `[private, slot]`

calls the crop method when the crop button is clicked

5.2.3.2 `vtkSmartPointer< vtkImageData > CroplImagesWidget::cropProbelImage (vtkSmartPointer< vtkImageData > image, int depthType)` `[private]`

Crop ultrasound image deppndng of the depth type.

5.2.3.3 `void CroplImagesWidget::save ()` `[private, slot]`

Save the cropped images in a folder.

5.2.3.4 `void CroplImagesWidget::setImage (vtkSmartPointer< vtkImageData > image)`

Set this vtkImageData.

Parameters

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

5.2.3.5 `void CroplImagesWidget::setImageStack (std::vector< vtkSmartPointer< vtkImageData > > imageStack)`

Set this stack of vtkImageData.

Parameters

| | | |
|-----------|----------|----------------------------|
| <i>in</i> | <i>a</i> | std Vector of vtkImageData |
|-----------|----------|----------------------------|

5.2.3.6 `void CroplImagesWidget::setMainWindow (MainWindow * mainwindow)`

Set the window to display the crop images.

5.2.4 Member Data Documentation

5.2.4.1 `vtkSmartPointer<vtkImageData> CroplImagesWidget::croplImage` [private]

the cropped image

5.2.4.2 `std::vector< vtkSmartPointer<vtkImageData> > CroplImagesWidget::cropStack` [private]

the cropped imageStack

5.2.4.3 `vtkSmartPointer<vtkImageData> CroplImagesWidget::image` [private]

the vtkImageData to work

5.2.4.4 `std::vector< vtkSmartPointer<vtkImageData> > CroplImagesWidget::imageStack` [private]

an Array of vtkImageData to work

5.2.4.5 `MainWindow* CroplImagesWidget::mainWindow` [private]

the main window to call it

5.2.4.6 `Ui::CroplImagesWidget* CroplImagesWidget::ui` [private]

5.2.4.7 `bool CroplImagesWidget::workWithStack` [private]

if there are multiple images to work with

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

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

5.3 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 [setSelectedOpacity](#) (int idx)
Set selected opacity for the displayed volume.
- 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.*
- void [openVolume](#) ()
Set a volume filename.
- void [croplImages](#) ()

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 [volumeFilename](#)
The filename of the selected volume.
- QString [textOnTextArea](#)
- [QVTKImageWidget](#) * [displayWidget](#)
- [vtkSmartPointer](#)
< [vtkEventQtSlotConnect](#) > [Connections](#)

5.3.1 Detailed Description

Main window for user interaction.

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

5.3.2 Constructor & Destructor Documentation

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

5.3.2.2 **MainWindow::~~MainWindow ()**

5.3.3 Member Function Documentation

5.3.3.1 **void MainWindow::addImages ()** [private, slot]

Add image folder to application.

5.3.3.2 void MainWindow::addLogText (QString *str*)

5.3.3.3 void MainWindow::cropImages () [private, slot]

5.3.3.4 void MainWindow::displaySelectedImage (int *idx*) [private, slot]

Display selected image with the image slider.

5.3.3.5 QVTKImageWidget * MainWindow::getDisplayWidget ()

return this display widget

Parameters

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

5.3.3.6 void MainWindow::openVolume () [private, slot]

Set a volume filename.

5.3.3.7 void MainWindow::openVolumeData () [private, slot]

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

5.3.3.8 void MainWindow::print () [private, slot]

Print message in logger.

5.3.3.9 void MainWindow::probeCalibration () [private, slot]

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

5.3.3.10 void MainWindow::setSelectedOpacity (int *idx*) [private, slot]

Set selected opacity for the displayed volume.

5.3.3.11 void MainWindow::volumeReconstruction () [private, slot]

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

5.3.4 Member Data Documentation

5.3.4.1 vtkSmartPointer<vtkEventQtSlotConnect> MainWindow::Connections [private]

5.3.4.2 QVTKImageWidget* MainWindow::displayWidget [private]

Central widget for display image purposes

5.3.4.3 QStringList MainWindow::imagesFileNames [private]

The filename of each selected image.

5.3.4.4 **QString MainWindow::textOnTextArea** [private]

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

5.3.4.6 **QString MainWindow::volumeCalibrationData** [private]

The filename of the estimated parameters in the calibration.

5.3.4.7 **QString MainWindow::volumeFilename** [private]

The filename of the selected volume.

5.3.4.8 **QStringList MainWindow::volumeImagesFileNames** [private]

The filenames of each selected volume image.

5.3.4.9 **QString MainWindow::volumeRotationData** [private]

The filename of the rotation data for each image.

5.3.4.10 **QString MainWindow::volumeTranslationData** [private]

The filename of the translation data for each image.

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

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

5.4 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 > [cropProbeImage](#) (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](#)

5.4.1 Detailed Description

Obtain data for calibration process.

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

5.4.2 Constructor & Destructor Documentation

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

Constructor

5.4.2.2 `ProbeCalibrationWidget::~~ProbeCalibrationWidget ()` [virtual]

5.4.3 Member Function Documentation

5.4.3.1 `void ProbeCalibrationWidget::calibrate ()` [private, slot]

Calls Calibrate.h to estimate the calibration parameters.

5.4.3.2 `void ProbeCalibrationWidget::crop ()` [private, slot]

crop the images to delete the extra information

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

Crop ultrasound image depending of the depth type

5.4.3.4 `void ProbeCalibrationWidget::getCoordinates ()` [private, slot]

get the cross point coordinates

5.4.3.5 `void ProbeCalibrationWidget::loadRotationsFile ()` [private, slot]

Load the rotations file.

5.4.3.6 `void ProbeCalibrationWidget::loadTranslationsFile ()` [private, slot]

Load the translation file.

5.4.3.7 `void ProbeCalibrationWidget::saveCalibration ()` [private, slot]

Save the Estimated Parameters in a .txt file.

5.4.3.8 `void ProbeCalibrationWidget::setImage (vtkSmartPointer< vtkImageData > image)`

Set this vtkImageData.

Parameters

| | | |
|----|---|-------------------------------|
| in | a | smart Pointer of vtkImageData |
|----|---|-------------------------------|

5.4.3.9 `void ProbeCalibrationWidget::setImageStack (std::vector< vtkSmartPointer< vtkImageData > > imageStack)`

Set this stack of vtkImageData.

Parameters

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

5.4.3.10 `void ProbeCalibrationWidget::setMainWindow (MainWindow * mainwindow)`

Set the window to display the crop images

5.4.4 Member Data Documentation

5.4.4.1 `std::vector<double> ProbeCalibrationWidget::calibrationParameters` [private]

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

5.4.4.2 `vnl_matrix<double> ProbeCalibrationWidget::coords` [private]

a vnl_matrix to store the selected coordinates on each image

5.4.4.3 `vtkSmartPointer<vtkImageData> ProbeCalibrationWidget::image` [private]

the vtkImageData to work

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

an Array of vtkImageData to work

5.4.4.5 `MainWindow* ProbeCalibrationWidget::mainWindow` [private]

the main window to call it

5.4.4.6 `vnl_matrix<double> ProbeCalibrationWidget::rotations` [private]

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

5.4.4.7 `float ProbeCalibrationWidget::rotations_2[7][4]` [private]

5.4.4.8 `vnl_matrix<double> ProbeCalibrationWidget::translations` [private]

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

5.4.4.9 `bool ProbeCalibrationWidget::workWithStack` [private]

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

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

5.5 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 translateFilename, 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 [setAndDisplayVolume](#) (QString volumeFilename)
Set and display volume data.
- void [setAndDisplayVolume](#) (vtkSmartPointer< vtkImageData > [volumeData](#))
Set and display volume data.
- 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.
- 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.
- void [setVolumeData](#) (vtkSmartPointer< vtkImageData > [volumeData](#))
Set the data of the display volume.
- void [setVolumeOpacity](#) (int opacity)
Set the opacity for the displayed volume.

- void [setVolumeOrigin](#) (vnl_vector< double > volumeOrigin)
- 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](#))
- void [displayVolume](#) (vtkSmartPointer< vtkVolume > [volume](#))
- 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](#)
- vtkSmartPointer< vtkVolume > [volume](#)
- vtkSmartPointer< vtkImageData > [volumeData](#)
- vtkSmartPointer
< vtkPiecewiseFunction > [volumeScalarOpacity](#)
- vtkSmartPointer
< vtkVolumeProperty > [volumeProperty](#)
- int [opacityPoint](#)
- 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.

5.5.1 Detailed Description

Display VTK images.

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

5.5.2 Constructor & Destructor Documentation

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

Constructor for this ImageWidget

5.5.2.2 QVTKImageWidget::~~QVTKImageWidget () [virtual]

Destructor

5.5.3 Member Function Documentation

5.5.3.1 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

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

Display the given vtkImage

5.5.3.3 void QVTKImageWidget::displaySelectedImage (int *idx*)

display an image stored in this imageStack.

Parameters

| | | |
|-----------|------------|--|
| <i>in</i> | <i>the</i> | index in the stack position of the image |
|-----------|------------|--|

5.5.3.4 void QVTKImageWidget::displayVolume (vtkSmartPointer< vtkVolume > *volume*) [private]

Display the given volume

5.5.3.5 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

5.5.3.6 int QVTKImageWidget::getImageDisplayedIndex ()

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

5.5.3.7 int QVTKImageWidget::getImageHeigth ()

returns this image heigth

5.5.3.8 int* QVTKImageWidget::getImageSize ()

returns an array with the width and height of the image

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

return this image stack

5.5.3.10 QString QVTKImageWidget::getImageType ()

Returns the type of image displayed.

5.5.3.11 vtkSmartPointer< vtkImageViewer2 > QVTKImageWidget::getImageViewer ()

Return this widget image viewer.

Parameters

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

5.5.3.12 int QVTKImageWidget::getImageWidth ()

returns this image width

5.5.3.13 QString QVTKImageWidget::getNumOfDimesions ()

Return the numer of dimensions of the image.

5.5.3.14 QString QVTKImageWidget::getPixelType ()

Returns the pixel type in loaded images.

5.5.3.15 QVTKWidget * QVTKImageWidget::getQVTKWidget ()

Return this qvtkWidget.

Parameters

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

5.5.3.16 vnl_vector< double > QVTKImageWidget::getTransformScale ()**5.5.3.17 std::vector< vnl_matrix< double > > QVTKImageWidget::getTransformStack ()**

return this transform stack

5.5.3.18 std::vector< vtkSmartPointer< vtkImageData > > QVTKImageWidget::getVolumeImageStack ()

return this volume image stack

5.5.3.19 int QVTKImageWidget::getXPicked ()

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

Parameters

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

5.5.3.20 int QVTKImageWidget::getYPicked ()

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

Parameters

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

5.5.3.21 void QVTKImageWidget::setAndDisplayImage (QString *imageFilename*)

Sets and display an image from a given image path

5.5.3.22 void QVTKImageWidget::setAndDisplayImage (vtkSmartPointer< vtkImageData > *image*)

Sets and display the given vtkImageData.

Parameters

| | | |
|----|---|---------------------------------|
| in | a | vtkImageData to set and display |
|----|---|---------------------------------|

5.5.3.23 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 | a | QStringList that contain the filename of each image |
|----|---|---|

5.5.3.24 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 |
|----|---|-----------------------------|

5.5.3.25 void QVTKImageWidget::setAndDisplayVolume (QString *volumeFilename*)

Set and display volume data.

Parameters

| | | |
|----|---|---|
| in | a | QString that contain the filename of the volume |
|----|---|---|

5.5.3.26 void QVTKImageWidget::setAndDisplayVolume (vtkSmartPointer< vtkImageData > *volumeData*)

Set and display volume data.

Parameters

| | | |
|----|---|--|
| in | a | vtkImageData generated by VolumeReconstruction.h |
|----|---|--|

5.5.3.27 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. |
|----|---|---|

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

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

5.5.3.29 void QVTKImageWidget::setVolumeData (vtkSmartPointer< vtkImageData > *volumeData*)

Set the data of the display volume.

5.5.3.30 void QVTKImageWidget::setVolumeOpacity (int *opacity*)

Set the opacity for the displayed volume.

5.5.3.31 void QVTKImageWidget::setVolumeOrigin (vnl_vector< double > *volumeOrigin*)

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

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

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

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

5.5.4 Member Data Documentation

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

Object for display information in the corners of the vtkImageViewer2.

5.5.4.2 int QVTKImageWidget::imageDisplayedIndex [private]

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

5.5.4.3 int QVTKImageWidget::imageHeight [private]

Height of the image

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

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

5.5.4.5 int QVTKImageWidget::imageType [private]

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

5.5.4.6 vtkSmartPointer<vtkImageViewer2> QVTKImageWidget::imageViewer [private]

the image viewer for display images

5.5.4.7 int QVTKImageWidget::imageWidth [private]

Width of the image

5.5.4.8 bool QVTKImageWidget::isImageStackLoaded

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

5.5.4.9 bool QVTKImageWidget::isVolumeImageStackLoaded

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

5.5.4.10 ImageType::Pointer QVTKImageWidget::itkImage [private]

The grayscale image displayed in this widget

5.5.4.11 size_t QVTKImageWidget::numDimensions [private]

The number of the image dimensions

5.5.4.12 int QVTKImageWidget::opacityPoint [private]**5.5.4.13 std::string QVTKImageWidget::pixelType [private]**

The type of the image pixels

5.5.4.14 QVTKWidget* QVTKImageWidget::qvtkWidget [private]

The QVTKWidget for display and interact with the images

5.5.4.15 vtkSmartPointer<vtkRenderer> QVTKImageWidget::renderer [private]

The VTK renderer

5.5.4.16 vtkSmartPointer<vtkRenderWindow> QVTKImageWidget::renwin [private]

The VTK render window

5.5.4.17 RGBImageType::Pointer QVTKImageWidget::rgbItkImage [private]

The RGB image displayed for this widget

5.5.4.18 vnl_vector<double> QVTKImageWidget::scale [private]

Estimated scale of the images

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

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

5.5.4.20 `vtkSmartPointer<vtkVolume> QVTKImageWidget::volume` [private]

The displayed volume data

5.5.4.21 `vtkSmartPointer<vtkImageData> QVTKImageWidget::volumeData` [private]

The displayed volume data

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

The angles and translation estimated

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

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

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

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

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

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

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

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

5.5.4.27 `vtkSmartPointer<vtkVolumeProperty> QVTKImageWidget::volumeProperty` [private]

The displayed volume properties

5.5.4.28 `vtkSmartPointer<vtkPiecewiseFunction> QVTKImageWidget::volumeScalarOpacity` [private]

The displayed volume opacity

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

The VTK image to display i this window

5.5.4.30 `int QVTKImageWidget::xPicked` [private]

The x coordinate of the picked position over the image

5.5.4.31 `int QVTKImageWidget::xPosition` [private]

current x coordinate of mouse position over the image

5.5.4.32 int QVTKImageWidget::yPicked [private]

current y coordinate of picked position over the image

5.5.4.33 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](#)

5.6 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](#)

5.6.1 Detailed Description

Interaction with mouse.

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

5.6.2 Constructor & Destructor Documentation

5.6.2.1 QVTKImageWidgetCommand::QVTKImageWidgetCommand ()

5.6.2.2 QVTKImageWidgetCommand::~~QVTKImageWidgetCommand ()

5.6.3 Member Function Documentation

5.6.3.1 void QVTKImageWidgetCommand::Execute (vtkObject * , unsigned long *vtkNotUsed*event, void *)
[virtual]

5.6.3.2 QVTKImageWidgetCommand * QVTKImageWidgetCommand::New () [static]

5.6.3.3 void QVTKImageWidgetCommand::SetAnnotation (vtkSmartPointer< vtkCornerAnnotation > *annotation*)

5.6.3.4 void QVTKImageWidgetCommand::SetImageWidget (QVTKImageWidget * *imageWidget*)

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

Parameters

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

5.6.3.5 void QVTKImageWidgetCommand::SetPicker (vtkSmartPointer< vtkPropPicker > *picker*)

5.6.4 Member Data Documentation

5.6.4.1 vtkSmartPointer<vtkCornerAnnotation> QVTKImageWidgetCommand::Annotation [private]

Pointer to the annotation

5.6.4.2 QVTKImageWidget* QVTKImageWidgetCommand::ImageWidget [private]

The widget related to the mouse events

5.6.4.3 vtkSmartPointer<vtkPropPicker> QVTKImageWidgetCommand::Picker [private]

Pointer to the picker

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

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

5.7 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](#)

5.7.1 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.

5.7.2 Member Function Documentation

5.7.2.1 void VolumeReconstruction::calcImagePlane () [private]

Compute the plane equation for each image.

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

Computes the maximum distance in the volume.

5.7.2.3 `double VolumeReconstruction::calcVoxelValue (std::vector< vnl_vector< double > > crossPoints,
vnl_vector< double > distancePlane, vnl_vector< double > distance) [private]`

Computes the voxel value using three lineal interpolation.

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

Returns the new volume data with the voxel based method.

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

Constructor.

5.7.2.6 `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.

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

Set the scale of the images.

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

Set the transformation for each image used in the reconstruction.

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

Set image data stack to generate the volume.

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

Set the volume data origin in the 3D scene.

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

Set the size of the volume data.

5.7.3 Member Data Documentation

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

Stacks for the image Bounds in x

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

Stacks for the image Bounds in Y

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

Stacks for the image Bounds in Z

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

The plane equation for each image

5.7.3.5 `double VolumeReconstruction::maxDistance` [private]

the maximun distance found in the volume

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

scale of the images

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

Contains the transformation for each image

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

The stack of images data

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

Where the volume data begins in the 3D scene

5.7.3.10 `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](#)

5.8 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.

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 in the 3D space.
- void [calcVolumeSize](#) (bool)
Computes the volume size.
- void [setVolumeOpacity](#) ()
Set the volume opacity.
- void [setVolumeColorMap](#) ()
Set the volume color 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](#)

5.8.1 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

5.8.2 Constructor & Destructor Documentation

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

5.8.2.2 `VolumeReconstructionWidget::~~VolumeReconstructionWidget ()`

5.8.3 Member Function Documentation

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

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

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

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

5.8.3.3 `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 |
|-----------|-----------|---|

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

Display the volume in the 3D scene.

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

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

5.8.3.6 `void VolumeReconstructionWidget::save ()` `[private, slot]`

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

5.8.3.7 `void VolumeReconstructionWidget::setDisplayProperties (vtkSmartPointer< vtkImageData >)`
`[private]`

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

5.8.3.8 `void VolumeReconstructionWidget::setMainWindow (MainWindow * mainwindow)`

Set the window to display the volume

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

Set the transformation stack for the volume image

5.8.3.10 void **VolumeReconstructionWidget::setVolumeColorMap** () [private]

Set the volume colo transfer function.

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

Set the image data stack

5.8.3.12 void **VolumeReconstructionWidget::setVolumeOpacity** () [private]

Set the volume opacity.

5.8.4 Member Data Documentation

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

Contains the transformed bounds in x of each image pixel

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

Contains the transformed bounds in y of each image pixel

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

Contains the transformed bounds in z of each image pixel

5.8.4.4 std::vector< vnl_matrix<double> > **VolumeReconstructionWidget::imageCoordsXStack** [private]

Contains the transformed coords in x of each image pixel

5.8.4.5 std::vector< vnl_matrix<double> > **VolumeReconstructionWidget::imageCoordsYStack** [private]

Contains the transformed coords in y of each image pixel

5.8.4.6 std::vector< vnl_matrix<double> > **VolumeReconstructionWidget::imageCoordsZStack** [private]

Contains the transformed coords in z of each image pixel

5.8.4.7 **MainWindow*** **VolumeReconstructionWidget::mainWindow** [private]

the main window to call it

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

Scale of the images

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

Contains the transformation of each image

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

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

the volume to display

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

Data of the volume

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

End of the volume data in the 3D space

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

The data of each image

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

Start of the volume data in the 3D space

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

Main volume properties

5.8.4.17 `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 CroplImagesWidget.cpp File Reference

```
#include "CropImagesWidget.h"  
#include "vtkExtractVOI.h"  
#include "vtkBMPWriter.h"
```

6.4 CroplImagesWidget.h File Reference

```
#include "ui_CropImagesWidget.h"  
#include "mainwindow.h"  
#include <QWidget>  
#include <vtkSmartPointer.h>  
#include <vtkImageData.h>
```

Classes

- class [CroplImagesWidget](#)

Crop images.

Namespaces

- namespace [Ui](#)

6.5 main.cpp File Reference

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

Functions

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

6.5.1 Function Documentation

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

6.6 mainwindow.cpp File Reference

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

6.7 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](#)

6.8 ProbeCalibrationWidget.cpp File Reference

```
#include "ProbeCalibrationWidget.h"
#include "Calibration.h"
#include <QErrorMessage>
#include <QString>
#include <QFile>
#include <QTextStream>
#include <vtkExtractVOI.h>
#include <vn1/vnl_quaternion.h>
#include <vn1/vnl_vector_fixed.h>
#include <vn1/algo/vnl_levenberg_marquardt.h>
#include <vn1/vnl_double_2.h>
```

Variables

- bool [setCoordsSize](#) = false

6.8.1 Variable Documentation

6.8.1.1 bool [setCoordsSize](#) = false

6.9 ProbeCalibrationWidget.h File Reference

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

Classes

- class [ProbeCalibrationWidget](#)

Obtain data for calibration process.

6.10 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 <vtkVolumeRayCastMapper.h>
#include <vtkVolumeRayCastCompositeFunction.h>
#include <vtkColorTransferFunction.h>
#include <vtkMath.h>
#include <vtkMetaImageReader.h>
```

6.11 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 <vtkVolume.h>
#include <vtkVolumeProperty.h>
#include <vtkPiecewiseFunction.h>
#include <vtkTransform.h>
#include <vn1/vnl_quaternion.h>
#include <vn1/vnl_matrix.h>
#include <vn1/vnl_vector.h>
```

Classes

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

Typedefs

- typedef itk::RGBPixel
< unsigned char > [RGBPixelType](#)

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

6.11.1 Typedef Documentation

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

6.11.1.2 typedef itk::Image< [RGBPixelType](#), 2> [RGBImageType](#)

6.11.1.3 typedef itk::RGBPixel< unsigned char > [RGBPixelType](#)

6.12 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>
```

6.13 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.

6.14 VolumeReconstruction.cpp File Reference

```
#include "VolumeReconstruction.h"
#include <vtkMath.h>
#include <vtkMetaImageWriter.h>
#include <vn1/vnl_inverse.h>
#include <exception>
#include <time.h>
```

6.15 VolumeReconstruction.h File Reference

```
#include <vtkSmartPointer.h>
#include <vtkImageData.h>
#include <vtkPlane.h>
#include <vn1/vnl_matrix.h>
#include <vn1/vnl_vector.h>
#include <math.h>
#include <vector>
```

Classes

- class [VolumeReconstruction](#)
Generate a new volume.

6.16 VolumeReconstructionWidget.cpp File Reference

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

6.17 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/vnl_matrix.h>
#include <vn1/vnl_vector.h>
```

Classes

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

Namespaces

- namespace [Ui](#)

Index

- ~CropImagesWidget
 - CropImagesWidget, [12](#)
- ~MainWindow
 - MainWindow, [14](#)
- ~ProbeCalibrationWidget
 - ProbeCalibrationWidget, [17](#)
- ~QVTKImageWidget
 - QVTKImageWidget, [22](#)
- ~QVTKImageWidgetCommand
 - QVTKImageWidgetCommand, [29](#)
- ~VolumeReconstructionWidget
 - VolumeReconstructionWidget, [35](#)
- addImages
 - MainWindow, [14](#)
- addLogText
 - MainWindow, [14](#)
- Annotation
 - QVTKImageWidgetCommand, [30](#)
- calcImageBounds
 - VolumeReconstructionWidget, [35](#)
- calcImageCoords
 - VolumeReconstructionWidget, [35](#)
- calcImagePlane
 - VolumeReconstruction, [31](#)
- calcMaxDistance
 - VolumeReconstruction, [31](#)
- calcVolumeSize
 - VolumeReconstructionWidget, [35](#)
- calcVoxelValue
 - VolumeReconstruction, [32](#)
- Calibrate
 - Calibration, [10](#)
- calibrate
 - ProbeCalibrationWidget, [18](#)
- 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, [39](#)
- Calibration.h, [39](#)
- calibrationParameters
 - ProbeCalibrationWidget, [19](#)
- ClearImagePoints
 - Calibration, [10](#)
- ClearTransformations
 - Calibration, [10](#)
- computeTransformation
 - QVTKImageWidget, [22](#)
- Connections
 - MainWindow, [15](#)
- coords
 - ProbeCalibrationWidget, [19](#)
- cornerAnnotation
 - QVTKImageWidget, [26](#)
- crop
 - CropImagesWidget, [12](#)
 - ProbeCalibrationWidget, [18](#)
- cropImage
 - CropImagesWidget, [12](#)
- cropImages
 - MainWindow, [15](#)
- CropImagesWidget, [11](#)
 - ~CropImagesWidget, [12](#)
 - crop, [12](#)
 - cropImage, [12](#)
 - CropImagesWidget, [12](#)
 - cropProbeImage, [12](#)
 - cropStack, [13](#)
 - CropImagesWidget, [12](#)
 - image, [13](#)
 - imageStack, [13](#)
 - mainWindow, [13](#)
 - save, [12](#)
 - setImage, [12](#)
 - setImageStack, [12](#)
 - setMainWindow, [12](#)
 - ui, [13](#)
 - workWithStack, [13](#)
- CropImagesWidget.cpp, [39](#)
- CropImagesWidget.h, [39](#)
- cropProbeImage
 - CropImagesWidget, [12](#)
 - ProbeCalibrationWidget, [18](#)
- cropStack
 - CropImagesWidget, [13](#)
- data
 - Calibration, [10](#)
- DataType

- Calibration, 10
- displayImage
 - QVTKImageWidget, 22
- displaySelectedImage
 - MainWindow, 15
 - QVTKImageWidget, 22
- displayVolume
 - QVTKImageWidget, 23
 - VolumeReconstructionWidget, 35
- displayVolumImages
 - QVTKImageWidget, 23
- displayWidget
 - MainWindow, 15
- estimatedUSCalibrationParameters
 - Calibration, 10
- Execute
 - QVTKImageWidgetCommand, 30
- generate
 - VolumeReconstructionWidget, 35
- generateVolume
 - VolumeReconstruction, 32
- getCoordinates
 - ProbeCalibrationWidget, 18
- getDisplayWidget
 - MainWindow, 15
- getEstimatedUSCalibrationParameters
 - Calibration, 10
- getImageDisplayedIndex
 - QVTKImageWidget, 23
- getImageHeight
 - QVTKImageWidget, 23
- getImageSize
 - QVTKImageWidget, 23
- getImageStack
 - QVTKImageWidget, 23
- getImageType
 - QVTKImageWidget, 23
- getImageViewer
 - QVTKImageWidget, 23
- getImageWidth
 - QVTKImageWidget, 23
- getNumOfDimesions
 - QVTKImageWidget, 23
- getPixelType
 - QVTKImageWidget, 24
- getQVTKWidget
 - QVTKImageWidget, 24
- getTransformScale
 - QVTKImageWidget, 24
- getTransformStack
 - QVTKImageWidget, 24
- getVolumImageStack
 - QVTKImageWidget, 24
- getXPicked
 - QVTKImageWidget, 24
- getYPicked
 - QVTKImageWidget, 24
- image
 - CropImagesWidget, 13
 - ProbeCalibrationWidget, 19
- imageBoundsXStack
 - VolumeReconstruction, 32
 - VolumeReconstructionWidget, 36
- imageBoundsYStack
 - VolumeReconstruction, 32
 - VolumeReconstructionWidget, 36
- imageBoundsZStack
 - VolumeReconstruction, 33
 - VolumeReconstructionWidget, 36
- imageCoordsXStack
 - VolumeReconstructionWidget, 36
- imageCoordsYStack
 - VolumeReconstructionWidget, 36
- imageCoordsZStack
 - VolumeReconstructionWidget, 36
- imageDisplayedIndex
 - QVTKImageWidget, 26
- imageHeight
 - QVTKImageWidget, 26
- imagePlaneStack
 - VolumeReconstruction, 33
- imagePoints
 - Calibration, 10
- imageStack
 - CropImagesWidget, 13
 - ProbeCalibrationWidget, 19
 - QVTKImageWidget, 26
- ImageType
 - QVTKImageWidget.h, 43
- imageType
 - QVTKImageWidget, 26
- imageViewer
 - QVTKImageWidget, 26
- ImageWidget
 - QVTKImageWidgetCommand, 30
- imageWidth
 - QVTKImageWidget, 26
- imagesFileNames
 - MainWindow, 15
- InsertImagePoints
 - Calibration, 10
- InsertTransformations
 - Calibration, 10
- isImageStackLoaded
 - QVTKImageWidget, 27
- isVolumImageStackLoaded
 - QVTKImageWidget, 27
- itkImage
 - QVTKImageWidget, 27
- loadRotationsFile
 - ProbeCalibrationWidget, 18
- loadTranslationsFile
 - ProbeCalibrationWidget, 18
- main

- main.cpp, 40
- main.cpp, 40
 - main, 40
- MainWindow, 13
 - ~MainWindow, 14
 - addImages, 14
 - addLogText, 14
 - Connections, 15
 - cropImages, 15
 - displaySelectedImage, 15
 - displayWidget, 15
 - getDisplayWidget, 15
 - imagesFilenames, 15
 - MainWindow, 14
 - MainWindow, 14
 - openVolume, 15
 - openVolumeData, 15
 - print, 15
 - probeCalibration, 15
 - setSelectedOpacity, 15
 - textOnTextArea, 15
 - ui, 16
 - volumeCalibrationData, 16
 - volumeFilename, 16
 - volumeImagesFilenames, 16
 - volumeReconstruction, 15
 - volumeRotationData, 16
 - volumeTranslationData, 16
- mainWindow
 - CropImagesWidget, 13
 - ProbeCalibrationWidget, 19
 - VolumeReconstructionWidget, 36
- mainwindow.cpp, 40
- mainwindow.h, 40
- maxDistance
 - VolumeReconstruction, 33
- New
 - Calibration, 10
 - QVTKImageWidgetCommand, 30
 - VolumeReconstruction, 32
- numDimensions
 - QVTKImageWidget, 27
- opacityPoint
 - QVTKImageWidget, 27
- openVolume
 - MainWindow, 15
- openVolumeData
 - MainWindow, 15
- Picker
 - QVTKImageWidgetCommand, 30
- pixelType
 - QVTKImageWidget, 27
- print
 - MainWindow, 15
- probeCalibration
 - MainWindow, 15
- ProbeCalibrationWidget, 16
 - ~ProbeCalibrationWidget, 17
 - calibrate, 18
 - calibrationParameters, 19
 - coords, 19
 - crop, 18
 - cropProbelImage, 18
 - getCoordinates, 18
 - image, 19
 - imageStack, 19
 - loadRotationsFile, 18
 - loadTranslationsFile, 18
 - mainWindow, 19
 - ProbeCalibrationWidget, 17
 - ProbeCalibrationWidget, 17
 - rotations, 19
 - rotations_2, 19
 - saveCalibration, 18
 - setImage, 18
 - setImageStack, 18
 - setMainWindow, 18
 - translations, 19
 - workWithStack, 19
- ProbeCalibrationWidget.cpp, 41
 - setCoordsSize, 41
- ProbeCalibrationWidget.h, 41
- QVTKImageWidget, 19
 - ~QVTKImageWidget, 22
 - computeTransformation, 22
 - cornerAnnotation, 26
 - displayImage, 22
 - displaySelectedImage, 22
 - displayVolume, 23
 - displayVolumeImages, 23
 - getImageDisplayedIndex, 23
 - getImageHeigh, 23
 - getImageSize, 23
 - getImageStack, 23
 - getImageType, 23
 - getImageViewer, 23
 - getImageWidth, 23
 - getNumOfDimesions, 23
 - getPixelType, 24
 - getQVTKWidget, 24
 - getTransformScale, 24
 - getTransformStack, 24
 - getVolumeImageStack, 24
 - getXPicked, 24
 - getYPicked, 24
 - imageDisplayedIndex, 26
 - imageHeight, 26
 - imageStack, 26
 - imageType, 26
 - imageView, 26
 - imageWidth, 26
 - isImageStackLoaded, 27
 - isVolumeImageStackLoaded, 27
 - itkImage, 27

- numDimensions, [27](#)
- opacityPoint, [27](#)
- pixelType, [27](#)
- QVTKImageWidget, [22](#)
- qvtkWidget, [27](#)
- QVTKImageWidget, [22](#)
- renderer, [27](#)
- renwin, [27](#)
- rgbItrImage, [27](#)
- scale, [27](#)
- setAndDisplayImage, [24](#)
- setAndDisplayMultipleImages, [25](#)
- setAndDisplayVolume, [25](#)
- setAndDisplayVolumeImages, [25](#)
- setImageProperties, [25](#)
- setVolumeData, [26](#)
- setVolumeOpacity, [26](#)
- setVolumeOrigin, [26](#)
- setXPicked, [26](#)
- setYPicked, [26](#)
- transformStack, [27](#)
- volume, [27](#)
- volumeData, [28](#)
- volumeDataCalibration, [28](#)
- volumeDataRotations, [28](#)
- volumeDataTranslations, [28](#)
- volumeImageActorStack, [28](#)
- volumeImageStack, [28](#)
- volumeProperty, [28](#)
- volumeScalarOpacity, [28](#)
- vtkImage, [28](#)
- xPicked, [28](#)
- xPosition, [28](#)
- yPicked, [28](#)
- yPosition, [29](#)
- QVTKImageWidget.cpp, [41](#)
- QVTKImageWidget.h, [42](#)
 - ImageType, [43](#)
 - RGBImageType, [43](#)
 - RGBPixelFormat, [43](#)
- QVTKImageWidgetCommand, [29](#)
 - ~QVTKImageWidgetCommand, [29](#)
 - Annotation, [30](#)
 - Execute, [30](#)
 - ImageWidget, [30](#)
 - New, [30](#)
 - Picker, [30](#)
 - QVTKImageWidgetCommand, [29](#)
 - QVTKImageWidgetCommand, [29](#)
 - SetAnnotation, [30](#)
 - SetImageWidget, [30](#)
 - SetPicker, [30](#)
- QVTKImageWidgetCommand.cpp, [43](#)
- QVTKImageWidgetCommand.h, [43](#)
- qvtkWidget
 - QVTKImageWidget, [27](#)
- RGBImageType
 - QVTKImageWidget.h, [43](#)
- RGBPixelFormat
 - QVTKImageWidget.h, [43](#)
- renderer
 - QVTKImageWidget, [27](#)
- renwin
 - QVTKImageWidget, [27](#)
- rgbItrImage
 - QVTKImageWidget, [27](#)
- rotations
 - ProbeCalibrationWidget, [19](#)
- rotations_2
 - ProbeCalibrationWidget, [19](#)
- save
 - CropImagesWidget, [12](#)
 - VolumeReconstructionWidget, [35](#)
- saveCalibration
 - ProbeCalibrationWidget, [18](#)
- scale
 - QVTKImageWidget, [27](#)
 - VolumeReconstruction, [33](#)
 - VolumeReconstructionWidget, [36](#)
- setAndDisplayImage
 - QVTKImageWidget, [24](#)
- setAndDisplayMultipleImages
 - QVTKImageWidget, [25](#)
- setAndDisplayVolume
 - QVTKImageWidget, [25](#)
- setAndDisplayVolumeImages
 - QVTKImageWidget, [25](#)
- SetAnnotation
 - QVTKImageWidgetCommand, [30](#)
- setCoordsSize
 - ProbeCalibrationWidget.cpp, [41](#)
- setDisplayProperties
 - VolumeReconstructionWidget, [35](#)
- setImage
 - CropImagesWidget, [12](#)
 - ProbeCalibrationWidget, [18](#)
- setImageBoundsStack
 - VolumeReconstruction, [32](#)
- setImageProperties
 - QVTKImageWidget, [25](#)
- setImageStack
 - CropImagesWidget, [12](#)
 - ProbeCalibrationWidget, [18](#)
- SetImageWidget
 - QVTKImageWidgetCommand, [30](#)
- setMainWindow
 - CropImagesWidget, [12](#)
 - ProbeCalibrationWidget, [18](#)
 - VolumeReconstructionWidget, [35](#)
- SetPicker
 - QVTKImageWidgetCommand, [30](#)
- setScale
 - VolumeReconstruction, [32](#)
- setSelectedOpacity
 - MainWindow, [15](#)
- setTransformStack

- VolumeReconstruction, [32](#)
- VolumeReconstructionWidget, [35](#)
- setVolumeColorMap
 - VolumeReconstructionWidget, [36](#)
- setVolumeData
 - QVTKImageWidget, [26](#)
- setVolumedImageStack
 - VolumeReconstruction, [32](#)
 - VolumeReconstructionWidget, [36](#)
- setVolumeOpacity
 - QVTKImageWidget, [26](#)
 - VolumeReconstructionWidget, [36](#)
- setVolumeOrigin
 - QVTKImageWidget, [26](#)
 - VolumeReconstruction, [32](#)
- setVolumeSize
 - VolumeReconstruction, [32](#)
- setXPicked
 - QVTKImageWidget, [26](#)
- setYPicked
 - QVTKImageWidget, [26](#)
- textOnTextArea
 - MainWindow, [15](#)
- transformStack
 - QVTKImageWidget, [27](#)
 - VolumeReconstruction, [33](#)
 - VolumeReconstructionWidget, [37](#)
- transformations
 - Calibration, [10](#)
- translations
 - ProbeCalibrationWidget, [19](#)
- Ui, [7](#)
- ui
 - CropImagesWidget, [13](#)
 - MainWindow, [16](#)
 - VolumeReconstructionWidget, [37](#)
- volume
 - QVTKImageWidget, [27](#)
 - VolumeReconstructionWidget, [37](#)
- volumeCalibrationData
 - MainWindow, [16](#)
- volumeData
 - QVTKImageWidget, [28](#)
 - VolumeReconstructionWidget, [37](#)
- volumeDataCalibration
 - QVTKImageWidget, [28](#)
- volumeDataRotations
 - QVTKImageWidget, [28](#)
- volumeDataTranslations
 - QVTKImageWidget, [28](#)
- volumeFilename
 - MainWindow, [16](#)
- volumeFinal
 - VolumeReconstructionWidget, [37](#)
- volumedImageActorStack
 - QVTKImageWidget, [28](#)
- volumedImageStack
 - QVTKImageWidget, [28](#)
 - VolumeReconstruction, [33](#)
 - VolumeReconstructionWidget, [37](#)
- volumedImageNames
 - MainWindow, [16](#)
- volumeOrigin
 - VolumeReconstruction, [33](#)
 - VolumeReconstructionWidget, [37](#)
- volumeProperty
 - QVTKImageWidget, [28](#)
 - VolumeReconstructionWidget, [37](#)
- VolumeReconstruction, [30](#)
 - calcImagePlane, [31](#)
 - calcMaxDistance, [31](#)
 - calcVoxelValue, [32](#)
 - generateVolume, [32](#)
 - imageBoundsXStack, [32](#)
 - imageBoundsYStack, [32](#)
 - imageBoundsZStack, [33](#)
 - imagePlaneStack, [33](#)
 - maxDistance, [33](#)
 - New, [32](#)
 - scale, [33](#)
 - setImageBoundsStack, [32](#)
 - setScale, [32](#)
 - setTransformStack, [32](#)
 - setVolumedImageStack, [32](#)
 - setVolumeOrigin, [32](#)
 - setVolumeSize, [32](#)
 - transformStack, [33](#)
 - volumedImageStack, [33](#)
 - volumeOrigin, [33](#)
 - volumeSize, [33](#)
- volumeReconstruction
 - MainWindow, [15](#)
- VolumeReconstruction.cpp, [43](#)
- VolumeReconstruction.h, [44](#)
- VolumeReconstructionWidget, [33](#)
 - ~VolumeReconstructionWidget, [35](#)
 - calcImageBounds, [35](#)
 - calcImageCoords, [35](#)
 - calcVolumeSize, [35](#)
 - displayVolume, [35](#)
 - generate, [35](#)
 - imageBoundsXStack, [36](#)
 - imageBoundsYStack, [36](#)
 - imageBoundsZStack, [36](#)
 - imageCoordsXStack, [36](#)
 - imageCoordsYStack, [36](#)
 - imageCoordsZStack, [36](#)
 - mainWindow, [36](#)
 - save, [35](#)
 - scale, [36](#)
 - setDisplayProperties, [35](#)
 - setMainWindow, [35](#)
 - setTransformStack, [35](#)
 - setVolumeColorMap, [36](#)

- setVolumeImageStack, [36](#)
- setVolumeOpacity, [36](#)
- transformStack, [37](#)
- ui, [37](#)
- volume, [37](#)
- volumeData, [37](#)
- volumeFinal, [37](#)
- volumeImageStack, [37](#)
- volumeOrigin, [37](#)
- volumeProperty, [37](#)
- VolumeReconstructionWidget, [35](#)
- volumeSize, [37](#)
- VolumeReconstructionWidget, [35](#)
- VolumeReconstructionWidget.cpp, [44](#)
- VolumeReconstructionWidget.h, [44](#)
- volumeRotationData
 - MainWindow, [16](#)
- volumeScalarOpacity
 - QVTKImageWidget, [28](#)
- volumeSize
 - VolumeReconstruction, [33](#)
 - VolumeReconstructionWidget, [37](#)
- volumeTranslationData
 - MainWindow, [16](#)
- vtkImage
 - QVTKImageWidget, [28](#)
- workWithStack
 - CropImagesWidget, [13](#)
 - ProbeCalibrationWidget, [19](#)
- xPicked
 - QVTKImageWidget, [28](#)
- xPosition
 - QVTKImageWidget, [28](#)
- yPicked
 - QVTKImageWidget, [28](#)
- yPosition
 - QVTKImageWidget, [29](#)