

Traitement d'images par filtre bilatéral

0.1

Généré par Doxygen 1.8.9.1

Mardi 21 Avril 2015 09 :41 :16

Table des matières

1	Index hiérarchique	1
1.1	Hiérarchie des classes	1
2	Index des classes	3
2.1	Liste des classes	3
3	Index des fichiers	5
3.1	Liste des fichiers	5
4	Documentation des classes	7
4.1	Référence de la classe Controller	7
4.1.1	Description détaillée	8
4.1.2	Documentation des constructeurs et destructeur	8
4.1.2.1	Controller	8
4.1.2.2	Controller	9
4.1.2.3	~Controller	9
4.1.3	Documentation des fonctions membres	9
4.1.3.1	bilateralFilter	9
4.1.3.2	changeDetailsUser	9
4.1.3.3	constructNewImage	10
4.1.3.4	getAlpha	10
4.1.3.5	getBeta	10
4.1.3.6	getDetail	10
4.1.3.7	getFileNameInput	10
4.1.3.8	getNbIteration	10
4.1.3.9	getSigmaR	10
4.1.3.10	getSigmaS	11
4.1.3.11	setAlpha	11
4.1.3.12	setBeta	11
4.1.3.13	setDetail	11
4.1.3.14	setFileNameInput	11
4.1.3.15	setNbIteration	12

4.1.3.16	setSigmaR	12
4.1.3.17	setSigmaS	12
4.1.4	Documentation des données membres	12
4.1.4.1	alpha	12
4.1.4.2	alphaLower	12
4.1.4.3	alphaRaise	12
4.1.4.4	beta	12
4.1.4.5	betaLower	13
4.1.4.6	betaRaise	13
4.1.4.7	detail	13
4.1.4.8	fileNameInput	13
4.1.4.9	fsigmaR	13
4.1.4.10	fsigmaS	13
4.1.4.11	iteration	13
4.1.4.12	nbliteration	13
4.1.4.13	sigmaR	13
4.1.4.14	sigmaS	13
4.2	Référence de la classe ControllerThread	13
4.2.1	Description détaillée	16
4.2.2	Documentation des constructeurs et destructeur	16
4.2.2.1	ControllerThread	16
4.2.2.2	~ControllerThread	16
4.2.3	Documentation des fonctions membres	16
4.2.3.1	complete	16
4.2.3.2	run	16
4.2.4	Documentation des données membres	17
4.2.4.1	controller	17
4.3	Référence de la classe FilterBilateral	17
4.3.1	Description détaillée	19
4.3.2	Documentation des constructeurs et destructeur	19
4.3.2.1	FilterBilateral	19
4.3.2.2	FilterBilateral	19
4.3.2.3	FilterBilateral	20
4.3.2.4	FilterBilateral	20
4.3.2.5	~FilterBilateral	20
4.3.3	Documentation des fonctions membres	20
4.3.3.1	applyFilter	20
4.3.3.2	applyFilterNdG	21
4.3.3.3	applyFilterRGB	21
4.3.3.4	distanceEuclidienne	21

4.3.3.5	get_fSigmaR	21
4.3.3.6	get_fSigmaS	21
4.3.3.7	get_img	22
4.3.3.8	loiGaussienne	22
4.3.3.9	set_fSigmaR	22
4.3.3.10	set_fSigmaS	22
4.3.3.11	set_img	22
4.3.4	Documentation des données membres	22
4.3.4.1	fSigmaR	22
4.3.4.2	fSigmaS	23
4.3.4.3	height	23
4.3.4.4	img	23
4.3.4.5	width	23
4.4	Référence de la classe MainWindow	23
4.4.1	Description détaillée	26
4.4.2	Documentation des constructeurs et destructeur	26
4.4.2.1	MainWindow	26
4.4.2.2	~MainWindow	26
4.4.3	Documentation des données membres	26
4.4.3.1	mainWidget	26
4.5	Référence de la classe WidgetImage	26
4.5.1	Description détaillée	29
4.5.2	Documentation des constructeurs et destructeur	30
4.5.2.1	WidgetImage	30
4.5.2.2	~WidgetImage	30
4.5.3	Documentation des fonctions membres	30
4.5.3.1	calculTermine	30
4.5.3.2	createConnection	31
4.5.3.3	createDetailsGroupBox	31
4.5.3.4	createInfoGroupBox	32
4.5.3.5	createInputGroupBox	32
4.5.3.6	createLoadGroupBox	32
4.5.3.7	createOutputGroupBox	33
4.5.3.8	createSaveGroupBox	33
4.5.3.9	launchButtonClicked	33
4.5.3.10	navLoadButtonClicked	34
4.5.3.11	navSaveButtonClicked	35
4.5.4	Documentation des données membres	35
4.5.4.1	alphaLineEdit	35
4.5.4.2	betaLineEdit	35

4.5.4.3	cThread	35
4.5.4.4	detailsGroupBox	35
4.5.4.5	fileNameInput	35
4.5.4.6	infoGroupBox	35
4.5.4.7	inputGroupBox	35
4.5.4.8	inputImage	35
4.5.4.9	inputLabel	36
4.5.4.10	launchButton	36
4.5.4.11	loadGroupBox	36
4.5.4.12	loadLineEdit	36
4.5.4.13	lowCheckBox	36
4.5.4.14	navLoadButton	36
4.5.4.15	navSaveButton	36
4.5.4.16	nblterationLineEdit	36
4.5.4.17	outputGroupBox	36
4.5.4.18	outputImage	36
4.5.4.19	outputLabel	36
4.5.4.20	raiseCheckBox	36
4.5.4.21	saveGroupBox	37
4.5.4.22	saveLineEdit	37
4.5.4.23	sigmaRLineEdit	37
4.5.4.24	sigmaSLineEdit	37
5	Documentation des fichiers	39
5.1	Référence du fichier InterfaceQt/controller.cpp	39
5.2	Référence du fichier InterfaceQt/controller.h	39
5.3	Référence du fichier InterfaceQt/controllerthread.cpp	40
5.4	Référence du fichier InterfaceQt/controllerthread.h	40
5.5	Référence du fichier InterfaceQt/filterbilateral.cpp	41
5.6	Référence du fichier InterfaceQt/filterbilateral.h	42
5.6.1	Documentation des macros	43
5.6.1.1	_USE_MATH_DEFINES	43
5.7	Référence du fichier InterfaceQt/main.cpp	43
5.7.1	Documentation des fonctions	44
5.7.1.1	main	44
5.8	Référence du fichier InterfaceQt/mainwindow.cpp	44
5.9	Référence du fichier InterfaceQt/mainwindow.h	44
5.10	Référence du fichier InterfaceQt/widgetImage.cpp	45
5.11	Référence du fichier InterfaceQt/widgetImage.h	45
Index		47

Chapitre 1

Index hiérarchique

1.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

Controller	7
FilterBilateral	17
QMainWindow	
MainWindow	23
QThread	
ControllerThread	13
QWidget	
WidgetImage	26

Chapitre 2

Index des classes

2.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

Controller	7
ControllerThread	
Classe permettant d'exécuter la fonction changeDetailsUser du Controller dans un thread . .	13
FilterBilateral	
Classe permettant de manipuler des images avec un filtre bilatéral Cette classe permet d'appliquer un filtre bilatéral sur une image en niveau de gris ou RGB	17
MainWindow	
Classe de l'interface de la fenêtre principale	23
WidgetImage	
Classe du widget principal Ce widget est contenu dans la fenêtre principale	26

Chapitre 3

Index des fichiers

3.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

InterfaceQt/controller.cpp	39
InterfaceQt/controller.h	39
InterfaceQt/controllerthread.cpp	40
InterfaceQt/controllerthread.h	40
InterfaceQt/filterbilateral.cpp	41
InterfaceQt/filterbilateral.h	42
InterfaceQt/main.cpp	43
InterfaceQt/mainwindow.cpp	44
InterfaceQt/mainwindow.h	44
InterfaceQt/widgetImage.cpp	45
InterfaceQt/widgetImage.h	45

Chapitre 4

Documentation des classes

4.1 Référence de la classe Controller

```
#include <controller.h>
```

Graphe de collaboration de Controller :

Controller
<ul style="list-style-type: none">- nbIteration- alpha- beta- sigmaS- sigmaR- fileNameInput- detail- iteration- fsigmaS- fsigmaR- alphaRaise- betaRaise- alphaLower- betaLower
<ul style="list-style-type: none">+ Controller()+ Controller()+ ~Controller()+ changeDetailsUser()+ getNbIteration()+ setNbIteration()+ getAlpha()+ setAlpha()+ getBeta()+ setBeta()et 8 de plus...- constructNewImage()- bilateralFilter()

Fonctions membres publiques

- `Controller` ()
Constructeur par défaut.
- `Controller` (const `Controller` &c)
Constructeur par recopie.
- `~Controller` ()
Destructeur.
- `QImage * changeDetailsUser` ()
Modifie l'image itérant un filtre bilatéral Cette fonction permet de manipuler les détails d'une image en itérant un filtre bilatéral dessus.
- `int getNbIteration` ()
- `void setNbIteration` (int value)
- `float getAlpha` ()
- `void setAlpha` (float value)
- `float getBeta` ()
- `void setBeta` (float value)
- `float getSigmaS` ()
- `void setSigmaS` (float value)
- `float getSigmaR` ()
- `void setSigmaR` (float value)
- `QString getFileNameInput` ()
- `void setFileNameInput` (const `QString` &value)
- `bool getDetail` ()
- `void setDetail` (bool value)

Fonctions membres privées

- `cimg_library : :Clmg< double > constructNewImage` (`cimg_library : :Clmg< double > &imgInput`)
Construit la nouvelle image en itérant le filtre bilatéral dessus.
- `cimg_library : :Clmg< double > bilateralFilter` (`cimg_library : :Clmg< double > img`, float sigma)
*Applique le filtre bilatéral à l'image **img***

Attributs privés

- `int nbIteration`
- `float alpha`
- `float beta`
- `float sigmaS`
- `float sigmaR`
- `QString fileNameInput`
- `bool detail`

Attributs privés statiques

- `static int iteration` = 3
- `static float fsigmaS` = 60.0
- `static float fsigmaR` = 15.0
- `static float alphaRaise` = 0.8
- `static float betaRaise` = 3
- `static float alphaLower` = 1
- `static float betaLower` = 0.5

4.1.1 Description détaillée

Définition à la ligne 13 du fichier controller.h.

4.1.2 Documentation des constructeurs et destructeur

4.1.2.1 `Controller : :Controller () [inline]`

Constructeur par défaut.

Définition à la ligne 19 du fichier controller.h.

4.1.2.2 Controller::Controller (const Controller & c) [inline]

Constructeur par recopie.

Paramètres

<i>c</i>	Controlleur a copié
----------	---------------------

Définition à la ligne 24 du fichier controller.h.

4.1.2.3 Controller::~~Controller () [inline]

Destructeur.

Définition à la ligne 28 du fichier controller.h.

4.1.3 Documentation des fonctions membres

4.1.3.1 CImg< double > Controller::bilateralFilter (cimg_library::CImg< double > *img*, float *sigma*) [private]

Applique le filtre bilatéral à l'image **img**

Paramètres

<i>img</i>	L'image sur laquelle le filtre bilatéral est exécuté
<i>sigma</i>	Valeur de sigma R (Sigma S étant fixe).

Renvoie

Retourne la résultante du filtre bilatéral.

Définition à la ligne 74 du fichier controller.cpp.

Voici le graphe d'appel pour cette fonction :



4.1.3.2 QImage * Controller::changeDetailsUser ()

Modifie l'image itérant un filtre bilatéral Cette fonction permet de manipuler les détails d'une image en itérant un filtre bilatéral dessus.

Renvoie

Retourne la nouvelle image reconstruite.

Définition à la ligne 14 du fichier controller.cpp.

Voici le graphe des appelants de cette fonction :



4.1.3.3 `CImg< double > Controller : :constructNewImage (cimg_library : :CImg< double > & imgInput) [private]`

Construit la nouvelle image en itérant le filtre bilatéral dessus.

Paramètres

<i>imgInput</i>	L'image à manipuler
-----------------	---------------------

Renvoie

Retourne la nouvelle image reconstruite.

Définition à la ligne 48 du fichier controller.cpp.

4.1.3.4 `float Controller : :getAlpha () [inline]`

Définition à la ligne 39 du fichier controller.h.

4.1.3.5 `float Controller : :getBeta () [inline]`

Définition à la ligne 41 du fichier controller.h.

4.1.3.6 `bool Controller : :getDetail () [inline]`

Définition à la ligne 51 du fichier controller.h.

4.1.3.7 `QString Controller : :getFileNameInput () [inline]`

Définition à la ligne 48 du fichier controller.h.

4.1.3.8 `int Controller : :getNbIteration () [inline]`

Définition à la ligne 37 du fichier controller.h.

4.1.3.9 `float Controller : :getSigmaR () [inline]`

Définition à la ligne 45 du fichier controller.h.

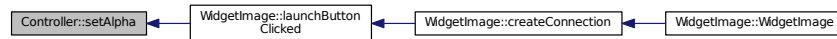
4.1.3.10 float Controller : :getSigmaS () [inline]

Définition à la ligne 43 du fichier controller.h.

4.1.3.11 void Controller : :setAlpha (float value) [inline]

Définition à la ligne 40 du fichier controller.h.

Voici le graphe des appelants de cette fonction :



4.1.3.12 void Controller : :setBeta (float value) [inline]

Définition à la ligne 42 du fichier controller.h.

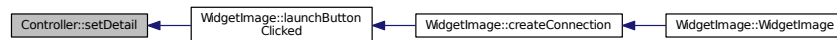
Voici le graphe des appelants de cette fonction :



4.1.3.13 void Controller : :setDetail (bool value) [inline]

Définition à la ligne 52 du fichier controller.h.

Voici le graphe des appelants de cette fonction :



4.1.3.14 void Controller : :setFileNameInput (const QString & value) [inline]

Définition à la ligne 49 du fichier controller.h.

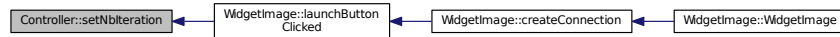
Voici le graphe des appelants de cette fonction :



4.1.3.15 void Controller : :setNbIteration (int *value*) [inline]

Définition à la ligne 38 du fichier controller.h.

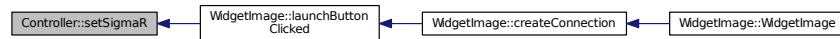
Voici le graphe des appelants de cette fonction :



4.1.3.16 void Controller : :setSigmaR (float *value*) [inline]

Définition à la ligne 46 du fichier controller.h.

Voici le graphe des appelants de cette fonction :



4.1.3.17 void Controller : :setSigmaS (float *value*) [inline]

Définition à la ligne 44 du fichier controller.h.

Voici le graphe des appelants de cette fonction :



4.1.4 Documentation des données membres

4.1.4.1 float Controller : :alpha [private]

Définition à la ligne 78 du fichier controller.h.

4.1.4.2 float Controller : :alphaLower = 1 [static], [private]

Définition à la ligne 60 du fichier controller.h.

4.1.4.3 float Controller : :alphaRaise = 0.8 [static], [private]

Définition à la ligne 58 du fichier controller.h.

4.1.4.4 float Controller : :beta [private]

Définition à la ligne 79 du fichier controller.h.

4.1.4.5 `float Controller : :betaLower = 0.5 [static], [private]`

Définition à la ligne 61 du fichier controller.h.

4.1.4.6 `float Controller : :betaRaise = 3 [static], [private]`

Définition à la ligne 59 du fichier controller.h.

4.1.4.7 `bool Controller : :detail [private]`

Définition à la ligne 83 du fichier controller.h.

4.1.4.8 `QString Controller : :fileNameInput [private]`

Définition à la ligne 82 du fichier controller.h.

4.1.4.9 `float Controller : :fsigmaR = 15.0 [static], [private]`

Définition à la ligne 57 du fichier controller.h.

4.1.4.10 `float Controller : :fsigmaS = 60.0 [static], [private]`

Définition à la ligne 56 du fichier controller.h.

4.1.4.11 `int Controller : :iteration = 3 [static], [private]`

Définition à la ligne 55 du fichier controller.h.

4.1.4.12 `int Controller : :nbIteration [private]`

Définition à la ligne 77 du fichier controller.h.

4.1.4.13 `float Controller : :sigmaR [private]`

Définition à la ligne 81 du fichier controller.h.

4.1.4.14 `float Controller : :sigmaS [private]`

Définition à la ligne 80 du fichier controller.h.

La documentation de cette classe a été générée à partir des fichiers suivants :

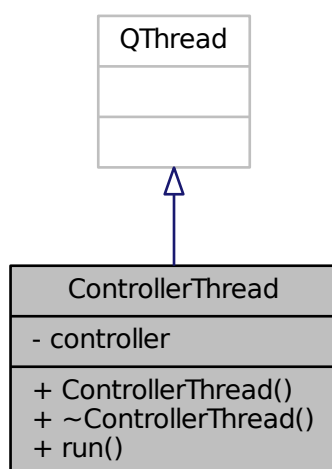
- InterfaceQt/[controller.h](#)
- InterfaceQt/[controller.cpp](#)

4.2 Référence de la classe ControllerThread

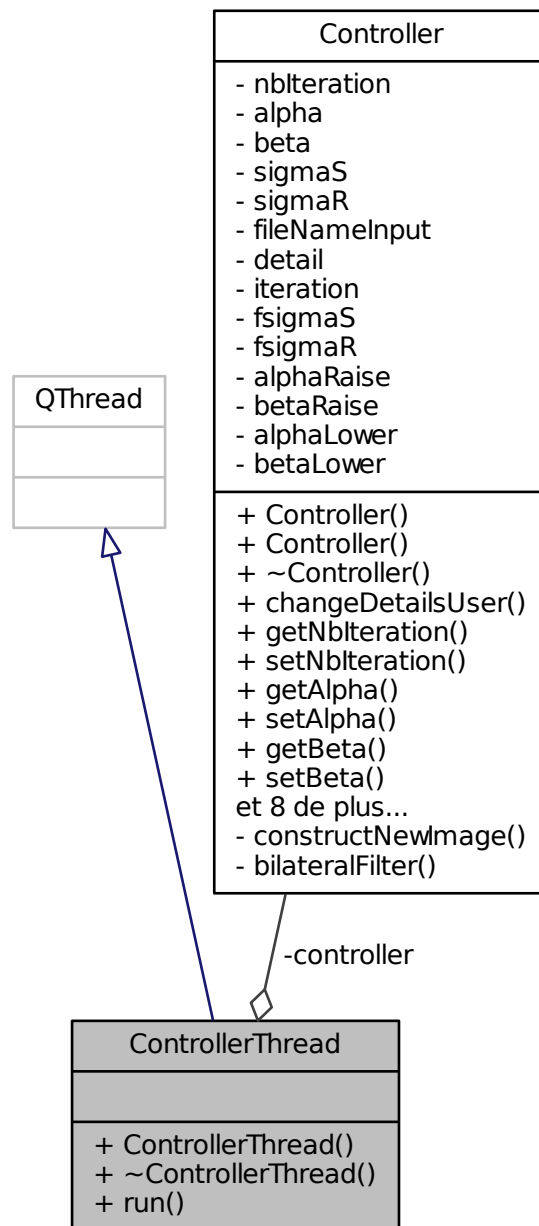
Classe permettant d'exécuter la fonction **changeDetailsUser** du [Controller](#) dans un thread.

```
#include <controllerthread.h>
```

Graphe d'héritage de ControllerThread :



Graphe de collaboration de ControllerThread :



Signaux

— void [complete](#) (QImage *)

Signal qui signifie la fin de l'exécution du thread Ce signal permet de communiquer avec le thread de l'application principale (GUI) en lui envoyant un objet QImage.

Fonctions membres publiques

— [ControllerThread](#) ([Controller](#) c)

- `~ControllerThread ()`
- `void run ()`

*Lancement de l'exécution du thread Le thread va appeler la fonction permettant de manipuler une image et lorsqu'il a fini il émet un signal (**complete()**) pour signifier qu'il a fini.*

Attributs privés

- `Controller controller`

4.2.1 Description détaillée

Classe permettant d'exécuter la fonction **changeDetailsUser** du `Controller` dans un thread.

Définition à la ligne 9 du fichier `controllerthread.h`.

4.2.2 Documentation des constructeurs et destructeur

4.2.2.1 `ControllerThread : :ControllerThread (Controller c) [inline]`

Définition à la ligne 14 du fichier `controllerthread.h`.

4.2.2.2 `ControllerThread : :~ControllerThread ()`

Définition à la ligne 3 du fichier `controllerthread.cpp`.

4.2.3 Documentation des fonctions membres

4.2.3.1 `void ControllerThread : :complete (QImage *) [signal]`

Signal qui signifie la fin de l'exécution du thread Ce signal permet de communiquer avec le thread de l'application principale (GUI) en lui envoyant un objet `QImage`.

Voici le graphe des appelants de cette fonction :



4.2.3.2 `void ControllerThread : :run ()`

Lancement de l'exécution du thread Le thread va appeler la fonction permettant de manipuler une image et lorsqu'il a fini il émet un signal (**complete()**) pour signifier qu'il a fini.

Définition à la ligne 9 du fichier `controllerthread.cpp`.

Voici le graphe d'appel pour cette fonction :



4.2.4 Documentation des données membres

4.2.4.1 Controller ControllerThread : :controller [private]

Définition à la ligne 31 du fichier controllerthread.h.

La documentation de cette classe a été générée à partir des fichiers suivants :

- InterfaceQt/[controllerthread.h](#)
- InterfaceQt/[controllerthread.cpp](#)

4.3 Référence de la classe FilterBilateral

Classe permettant de manipuler des images avec un filtre bilateral Cette classe permet d'appliquer un filtre bilateral sur une image en niveau de gris ou RGB.

```
#include <filterbilateral.h>
```

Graphe de collaboration de FilterBilateral :

FilterBilateral
<ul style="list-style-type: none"> - img - fSigmaS - fSigmaR - width - height
<ul style="list-style-type: none"> + FilterBilateral() + FilterBilateral() + FilterBilateral() + FilterBilateral() + ~FilterBilateral() + applyFilter() + applyFilterNdG() + applyFilterRGB() + get_fSigmaS() + get_fSigmaR() + get_img() + set_fSigmaS() + set_fSigmaR() + set_img() - loiGaussienne() - distanceEuclidienne()

Fonctions membres publiques

- [FilterBilateral](#) ()
Constructeur par défaut.
- [FilterBilateral](#) (const [FilterBilateral](#) &other)
Constructeur par copie.
- [FilterBilateral](#) (float fsigmaS, float fsigmaR, const cimg_library : :CImg< double > &input)
Constructeur.
- [FilterBilateral](#) (const cimg_library : :CImg< double > &input)
Constructeur.
- [~FilterBilateral](#) ()
Destructeur.
- cimg_library : :CImg< double > [applyFilter](#) ()
Applique le filtre bilateral à l'image initiale Appelle [applyFilterNdG\(\)](#) si l'image est en niveau de gris sinon [applyFilterRGB\(\)](#)
- cimg_library : :CImg< double > [applyFilterNdG](#) ()
Applique le filtre bilateral à une image en niveau de gris.
- cimg_library : :CImg< double > [applyFilterRGB](#) ()
Applique le filtre bilateral à une image RGB Le filtre bilateral est appliquée à la luminance.
- float [get_fSigmaS](#) ()
Getter pour sigma S.
- float [get_fSigmaR](#) ()
Getter pour sigma R.
- cimg_library : :CImg< double > [get_img](#) ()
Getter de l'image initiale.
- void [set_fSigmaS](#) (float sigmaS)
Change la valeur de sigma S.
- void [set_fSigmaR](#) (float sigmaR)
Change la valeur de sigma R.

- void **set_img** (cimg_library : :CImg< double > &newImg)
Change l'image initiale.

Fonctions membres privées

- double **loiGaussienne** (double value, float sigma)
calcul d'une gaussienne Calcul d'une gaussienne
- double **distanceEuclidienne** (int xP, int yP, int xQ, int yQ)
Calcul de la distance euclidienne entre deux points Calcul de la distance euclidienne entre deux points.

Attributs privés

- cimg_library : :CImg< double > **img**
Image initiale.
- float **fSigmaS**
Valeur de sigma S.
- float **fSigmaR**
Valeur de sigma R.
- int **width**
Nombre de colonne de l'image initiale.
- int **height**
Nombre de ligne de l'image initiale.

4.3.1 Description détaillée

Classe permettant de manipuler des images avec un filtre bilateral Cette classe permet d'appliquer un filtre bilateral sur une image en niveau de gris ou RGB.

Auteur

Natacha Marlio-Marette

Version

1.0

Date

11/03/2015

Définition à la ligne 16 du fichier filterbilateral.h.

4.3.2 Documentation des constructeurs et destructeur

4.3.2.1 FilterBilateral : :FilterBilateral () [inline]

Constructeur par défaut.

Définition à la ligne 22 du fichier filterbilateral.h.

4.3.2.2 FilterBilateral : :FilterBilateral (const FilterBilateral & other) [inline]

Constructeur par copie.

Paramètres

<i>other</i>	Objet FilterBilateral a recopié
--------------	---

Définition à la ligne 27 du fichier filterbilateral.h.

4.3.2.3 `FilterBilateral : :FilterBilateral (float fsigmaS, float fsigmaR, const cimg_library : :CImg< double > & input)`
`[inline]`

Constructeur.

Paramètres

<i>fsigmaS</i>	Valeur de sigma S
<i>fsigmaR</i>	Valeur de sigma R
<i>input</i>	Image initiale

Définition à la ligne 34 du fichier filterbilateral.h.

4.3.2.4 `FilterBilateral : :FilterBilateral (const cimg_library : :CImg< double > & input)` `[inline]`

Constructeur.

Paramètres

<i>input</i>	Image initiale
--------------	----------------

Définition à la ligne 39 du fichier filterbilateral.h.

4.3.2.5 `FilterBilateral : :~FilterBilateral ()` `[inline]`

Destructeur.

Définition à la ligne 43 du fichier filterbilateral.h.

4.3.3 Documentation des fonctions membres

4.3.3.1 `CImg< double > FilterBilateral : :applyFilter ()`

Applique le filtre bilateral à l'image initiale Appelle [applyFilterNdG\(\)](#) si l'image est en niveau de gris sinon [apply↔FilterRGB\(\)](#)

Renvoie

Retourne une nouvelle image instanciée sur laquelle le filtre bilatéral a été appliquée

Définition à la ligne 20 du fichier filterbilateral.cpp.

Voici le graphe des appelants de cette fonction :



4.3.3.2 `CImg< double > FilterBilateral : :applyFilterNdG ()`

Applique le filtre bilateral à une image en niveau de gris.

Renvoie

Retourne une nouvelle image instanciée

Définition à la ligne 35 du fichier filterbilateral.cpp.

4.3.3.3 `CImg< double > FilterBilateral : :applyFilterRGB ()`

Applique le filtre bilateral à une image RGB Le filtre bilateral est appliquée à la luminance.

Renvoie

Retourne une nouvelle image instanciée

Définition à la ligne 104 du fichier filterbilateral.cpp.

4.3.3.4 `double FilterBilateral : :distanceEuclidienne (int xP, int yP, int xQ, int yQ) [private]`

Calcul de la distance euclidienne entre deux points Calcul de la distance euclidienne entre deux points.

Paramètres

<code>xP</code>	Coordonnée x du premier point
<code>yP</code>	Coordonnée y du premier point
<code>xQ</code>	Coordonnée x du second point
<code>yQ</code>	Coordonnée y du second point

Renvoie

Résultat de la distance euclidienne entre les deux points

Définition à la ligne 13 du fichier filterbilateral.cpp.

4.3.3.5 `float FilterBilateral : :get_fSigmaR () [inline]`

Getter pour sigma R.

Renvoie

Retourne sigma R

Définition à la ligne 72 du fichier filterbilateral.h.

4.3.3.6 `float FilterBilateral : :get_fSigmaS () [inline]`

Getter pour sigma S.

Renvoie

Retourne sigma S

Définition à la ligne 67 du fichier filterbilateral.h.

4.3.3.7 `cimg_library : :CImg<double> FilterBilateral : :get_img () [inline]`

Getter de l'image initiale.

Renvoie

Retourne l'iamge initiale

Définition à la ligne 77 du fichier filterbilateral.h.

4.3.3.8 `double FilterBilateral : :loiGaussienne (double value, float sigma) [private]`

calcul d'une gaussienne Calcul d'une gaussienne

Paramètres

<i>value</i>	Paramètre de la gaussienne
<i>sigma</i>	Valeur du sigma de la gaussienne

Renvoie

Résultat du calcul de la gaussienne

Définition à la ligne 6 du fichier filterbilateral.cpp.

4.3.3.9 `void FilterBilateral : :set_fSigmaR (float sigmaR) [inline]`

Change la valeur de sigma R.

Paramètres

<i>sigmaS</i>	Valeur de sigma R
---------------	-------------------

Définition à la ligne 88 du fichier filterbilateral.h.

4.3.3.10 `void FilterBilateral : :set_fSigmaS (float sigmaS) [inline]`

Change la valeur de sigma S.

Paramètres

<i>sigmaS</i>	Valeur de sigma S
---------------	-------------------

Définition à la ligne 83 du fichier filterbilateral.h.

4.3.3.11 `void FilterBilateral : :set_img (cimg_library : :CImg< double > & newImg) [inline]`

Change l'image initiale.

Paramètres

<i>newImg</i>	Nouvelle image
---------------	----------------

Définition à la ligne 93 du fichier filterbilateral.h.

4.3.4 Documentation des données membres

4.3.4.1 `float FilterBilateral : :fSigmaR [private]`

Valeur de sigma R.

Définition à la ligne 126 du fichier filterbilateral.h.

4.3.4.2 `float FilterBilateral : :fSigmaS [private]`

Valeur de sigma S.

Définition à la ligne 122 du fichier filterbilateral.h.

4.3.4.3 `int FilterBilateral : :height [private]`

Nombre de ligne de l'image initiale.

Définition à la ligne 134 du fichier filterbilateral.h.

4.3.4.4 `cimg_library : :CImg<double> FilterBilateral : :img [private]`

Image initiale.

Définition à la ligne 118 du fichier filterbilateral.h.

4.3.4.5 `int FilterBilateral : :width [private]`

Nombre de colonne de l'image initiale.

Définition à la ligne 130 du fichier filterbilateral.h.

La documentation de cette classe a été générée à partir des fichiers suivants :

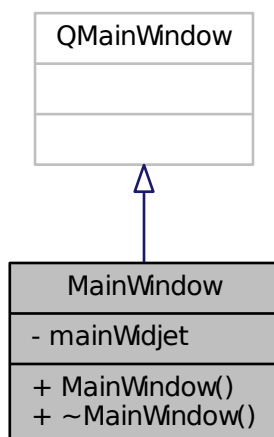
- InterfaceQt/[filterbilateral.h](#)
- InterfaceQt/[filterbilateral.cpp](#)

4.4 Référence de la classe MainWindow

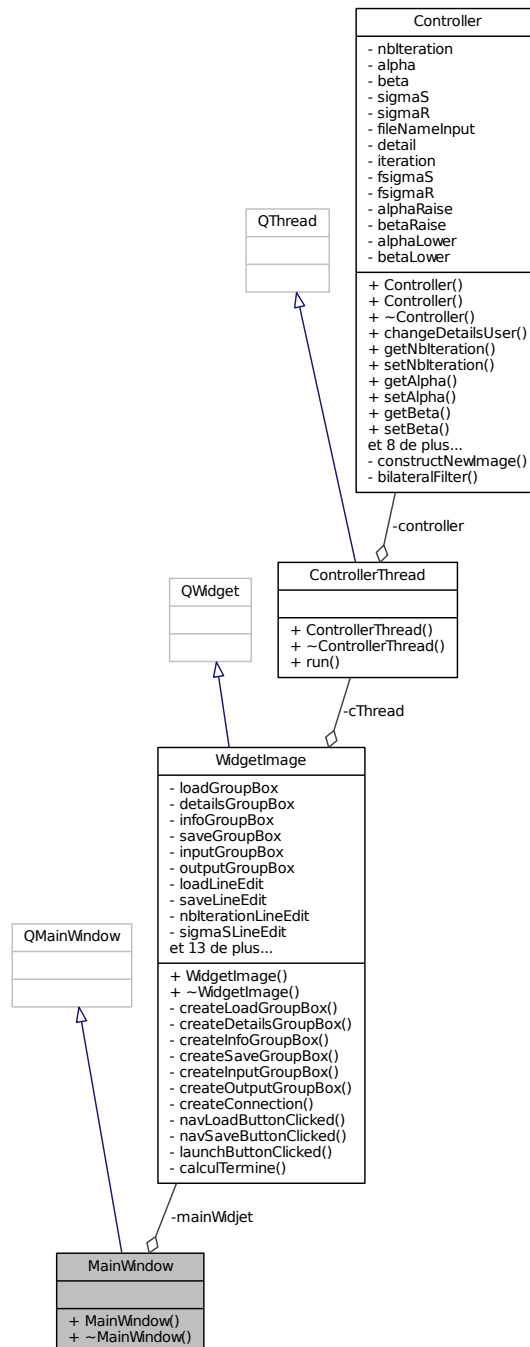
Classe de l'interface de la fenêtre principale.

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

Graphe d'héritage de MainWindow :



Graphe de collaboration de MainWindow :



Fonctions membres publiques

- [MainWindow](#) (QWidget *parent=0)
- [~MainWindow](#) ()

Attributs privés

- [WidgetImage](#) * [mainWdjet](#)

Widget principal de la fenêtre.

4.4.1 Description détaillée

Classe de l'interface de la fenêtre principale.

Auteur

Natacha Marlio-Marette

Version

0.1

Date

10/03/2015

Définition à la ligne 13 du fichier mainwindow.h.

4.4.2 Documentation des constructeurs et destructeur

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

Définition à la ligne 3 du fichier mainwindow.cpp.

4.4.2.2 MainWindow : :~MainWindow ()

Définition à la ligne 10 du fichier mainwindow.cpp.

4.4.3 Documentation des données membres

4.4.3.1 WidgetImage* MainWindow : :mainWidget [private]

Widget principal de la fenêtre.

Définition à la ligne 25 du fichier mainwindow.h.

La documentation de cette classe a été générée à partir des fichiers suivants :

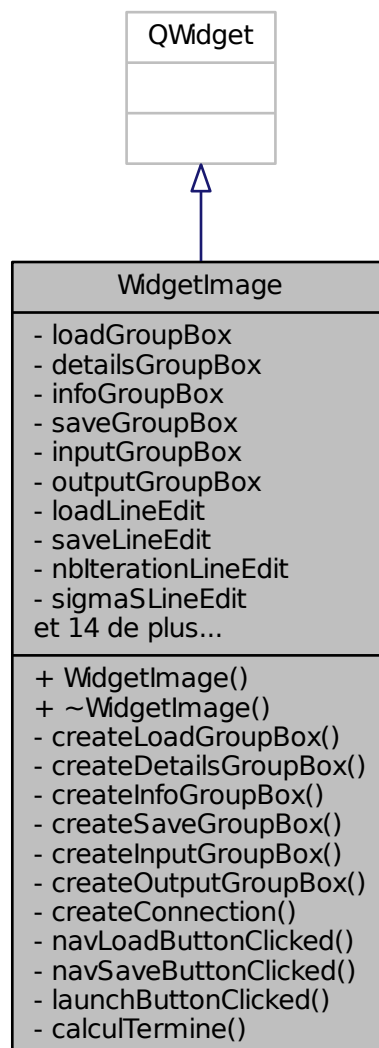
- InterfaceQt/[mainwindow.h](#)
- InterfaceQt/[mainwindow.cpp](#)

4.5 Référence de la classe WidgetImage

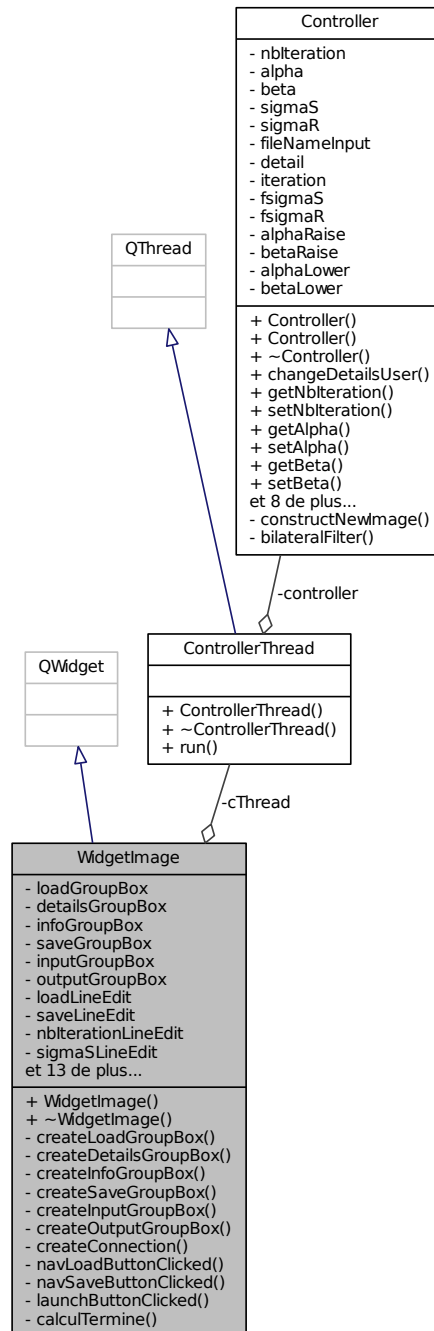
Classe du widget principal Ce widget est contenu dans la fenêtre principale.

```
#include <widgetImage.h>
```


Graphe d'héritage de WidgetImage :



Graphe de collaboration de WidgetImage :



Fonctions membres publiques

- [WidgetImage](#) (QWidget *parent=0)
Constructeur du widget.
- [~WidgetImage](#) ()
Destructeur.

Connecteurs privés

- void [navLoadButtonClicked](#) ()
Gestion de l'évènement click sur le bouton de navigation de chargement de l'image d'entrée Ouvre une fenêtre de navigation et charge l'image choisie.
- void [navSaveButtonClicked](#) ()
Gestion de l'évènement click sur le bouton de navigation de chargement de l'image de sortie Ouvre une fenêtre de navigation et sauvegarde l'image de sortie.
- void [launchButtonClicked](#) ()
Gestion de l'évènement click sur le bouton de lancement du filtre Récupération s'il y a des informations saisies par l'utilisateur et vérification de leur validité. Un objet controleur est crée prenant en compte ces informations si elles sont présentes ainsi que l'image d'entrée et le niveau de détail demandé. Un objet [ControllerThread](#) est crée prenant en paramètre le controleur précédemment crée. Ce thread connecte sa sortie au GUI afin qu'ils puissent communiquer. Le thread est ensuite démarré ce qui lance le traitement sur l'image d'entrée.
- void [calculTermine](#) (QImage *img)
Slot reçu lors de la fin de l'exécution du [ControllerThread](#).

Fonctions membres privées

- void [createLoadGroupBox](#) ()
Crée le widget GroupBox du chargement d'une image ainsi que son contenu.
- void [createDetailsGroupBox](#) ()
Crée le widget GroupBox du choix du niveau de détail ainsi que son contenu.
- void [createInfoGroupBox](#) ()
Crée le widget GroupBox permettant à l'utilisateur de rentrer manuellement des informations ainsi que son contenu.
- void [createSaveGroupBox](#) ()
Crée le widget GroupBox de sauvegarde d'une image ainsi que son contenu.
- void [createInputGroupBox](#) ()
Crée le widget GroupBox d'affichage d'une image d'entrée ainsi que son contenu.
- void [createOutputGroupBox](#) ()
Crée le widget GroupBox de l'image de sortie ainsi que son contenu.
- void [createConnection](#) ()
Création des connections Cette méthode permet de créer les connections entre les boutons et les actions qu'ils engendrent.

Attributs privés

- QGroupBox * [loadGroupBox](#)
- QGroupBox * [detailsGroupBox](#)
- QGroupBox * [infoGroupBox](#)
- QGroupBox * [saveGroupBox](#)
- QGroupBox * [inputGroupBox](#)
- QGroupBox * [outputGroupBox](#)
- QLineEdit * [loadLineEdit](#)
- QLineEdit * [saveLineEdit](#)
- QLineEdit * [nbIterationLineEdit](#)
- QLineEdit * [sigmaSLineEdit](#)
- QLineEdit * [sigmaRLineEdit](#)
- QLineEdit * [alphaLineEdit](#)
- QLineEdit * [betaLineEdit](#)
- QPushButton * [navLoadButton](#)
- QPushButton * [launchButton](#)
- QPushButton * [navSaveButton](#)
- QCheckBox * [raiseCheckBox](#)
- QCheckBox * [lowCheckBox](#)
- QLabel * [inputLabel](#)
- QLabel * [outputLabel](#)
- QImage * [inputImage](#)
- QImage * [outputImage](#)
- QString [fileNameInput](#)
- [ControllerThread](#) * [cThread](#)
cThread permet d'exécuter le controller sur un thread à part L'exécution du controller sur un thread à part permet de ne pas bloquer l'interface lors des calculs.

4.5.1 Description détaillée

Classe du widget principal Ce widget est contenu dans la fenêtre principale.

Auteur

Natacha Marlio-Marette

Version

0.1

Date

10/03/2015

Définition à la ligne 26 du fichier widgetImage.h.

4.5.2 Documentation des constructeurs et destructeur**4.5.2.1 WidgetImage : :WidgetImage (QWidget * *parent* = 0) [explicit]**

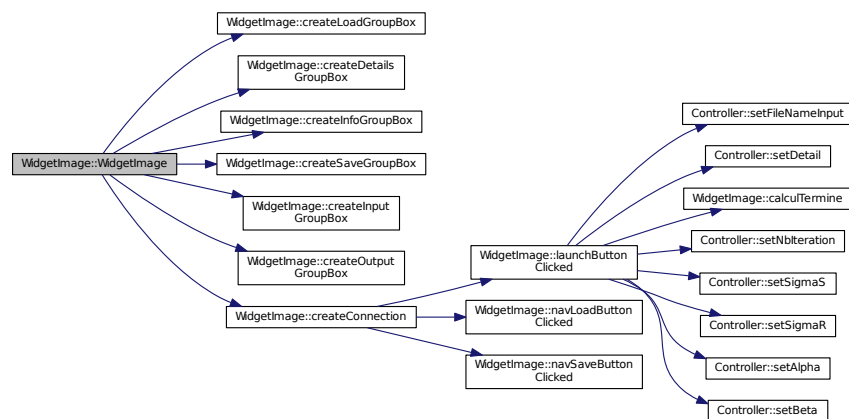
Constructeur du widget.

Paramètres

<i>parent</i>	
---------------	--

Définition à la ligne 3 du fichier widgetImage.cpp.

Voici le graphe d'appel pour cette fonction :

**4.5.2.2 WidgetImage : :~WidgetImage () [inline]**

Destructeur.

Définition à la ligne 39 du fichier widgetImage.h.

4.5.3 Documentation des fonctions membres**4.5.3.1 void WidgetImage : :calculTermine (QImage * *img*) [private],[slot]**Slot reçu lors de la fin de l'exécution du [ControllerThread](#).

Paramètres

<i>img</i>	Représente la QImage créée après manipulation de l'image d'entrée
------------	---

Définition à la ligne 272 du fichier widgetImage.cpp.

Voici le graphe des appelants de cette fonction :

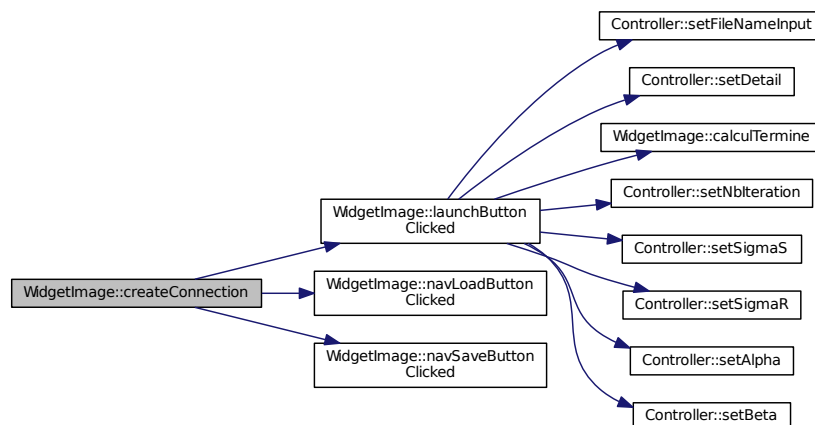


4.5.3.2 void WidgetImage : :createConnection () [private]

Création des connections Cette méthode permet de créer les connections entre les boutons et les actions qu'ils engendrent.

Définition à la ligne 156 du fichier widgetImage.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



4.5.3.3 void WidgetImage : :createDetailsGroupBox () [private]

Crée le widget GroupBox du choix du niveau de détail ainsi que son contenu.

Définition à la ligne 55 du fichier widgetImage.cpp.

Voici le graphe des appelants de cette fonction :



4.5.3.4 void WidgetImage : :createInfoGroupBox () [private]

Crée le widget GroupBox permettant à l'utilisateur de rentrer manuellement des informations ainsi que son contenu.

Définition à la ligne 74 du fichier widgetImage.cpp.

Voici le graphe des appelants de cette fonction :



4.5.3.5 void WidgetImage : :createInputGroupBox () [private]

Crée le widget GroupBox d'affichage d'une image d'entrée ainsi que son contenu.

Définition à la ligne 134 du fichier widgetImage.cpp.

Voici le graphe des appelants de cette fonction :

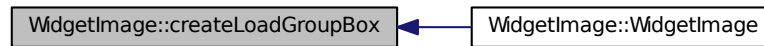


4.5.3.6 void WidgetImage : :createLoadGroupBox () [private]

Crée le widget GroupBox du chargement d'une image ainsi que son contenu.

Définition à la ligne 38 du fichier widgetImage.cpp.

Voici le graphe des appelants de cette fonction :



4.5.3.7 void WidgetImage : :createOutputGroupBox () [private]

Crée le widget GroupBox de l'image de sortie ainsi que son contenu.

Définition à la ligne 145 du fichier widgetImage.cpp.

Voici le graphe des appelants de cette fonction :



4.5.3.8 void WidgetImage : :createSaveGroupBox () [private]

Crée le widget GroupBox de sauvegarde d'une image ainsi que son contenu.

Définition à la ligne 117 du fichier widgetImage.cpp.

Voici le graphe des appelants de cette fonction :

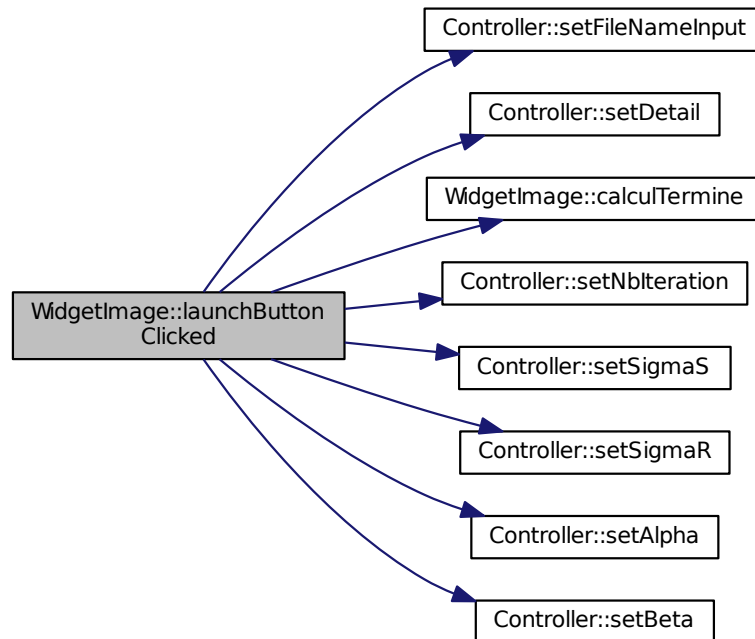


4.5.3.9 void WidgetImage : :launchButtonClicked () [private],[slot]

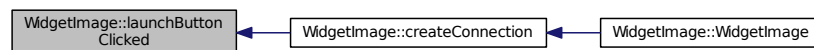
Gestion de l'évènement click sur le bouton de lancement du filtre Récupération s'il y en a des informations saisies par l'utilisateur et vérification de leur validité. Un objet contrôleur est créé prenant en compte ces informations si elles sont présentes ainsi que l'image d'entrée et le niveau de détail demandé. Un objet [ControllerThread](#) est créé prenant en paramètre le contrôleur précédemment créé. Ce thread connecte sa sortie au GUI afin qu'ils puissent communiquer. Le thread est ensuite démarré ce qui lance le traitement sur l'image d'entrée.

Définition à la ligne 204 du fichier widgetImage.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

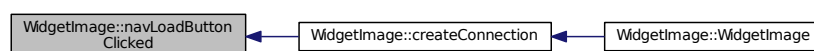


4.5.3.10 void WidgetImage : :navLoadButtonClicked () [private],[slot]

Gestion de l'évènement click sur le bouton de navigation de chargement de l'image d'entrée Ouvre une fenêtre de navigation et charge l'image choisie.

Définition à la ligne 164 du fichier widgetImage.cpp.

Voici le graphe des appelants de cette fonction :

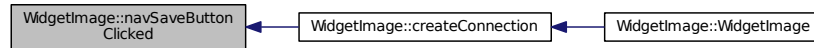


4.5.3.11 void WidgetImage : :navSaveButtonClicked () [private],[slot]

Gestion de l'évènement click sur le bouton de navigation de chargement de l'image de sortie Ouvre une fenêtre de navigation et sauvegarde l'image de sortie.

Définition à la ligne 186 du fichier widgetImage.cpp.

Voici le graphe des appelants de cette fonction :



4.5.4 Documentation des données membres

4.5.4.1 QLineEdit* WidgetImage : :alphaLineEdit [private]

Définition à la ligne 106 du fichier widgetImage.h.

4.5.4.2 QLineEdit* WidgetImage : :betaLineEdit [private]

Définition à la ligne 107 du fichier widgetImage.h.

4.5.4.3 ControllerThread* WidgetImage : :cThread [private]

cThread permet d'exécuter le controller sur un thread à part L'exécution du controller sur un thread à part permet de ne pas bloquer l'interface lors des calculs.

Définition à la ligne 127 du fichier widgetImage.h.

4.5.4.4 QGroupBox* WidgetImage : :detailsGroupBox [private]

Définition à la ligne 95 du fichier widgetImage.h.

4.5.4.5 QString WidgetImage : :fileNameInput [private]

Définition à la ligne 122 du fichier widgetImage.h.

4.5.4.6 QGroupBox* WidgetImage : :infoGroupBox [private]

Définition à la ligne 96 du fichier widgetImage.h.

4.5.4.7 QGroupBox* WidgetImage : :inputGroupBox [private]

Définition à la ligne 98 du fichier widgetImage.h.

4.5.4.8 QImage* WidgetImage : :inputImage [private]

Définition à la ligne 119 du fichier widgetImage.h.

4.5.4.9 QLabel* WidgetImage : :inputLabel [private]

Définition à la ligne 116 du fichier widgetImage.h.

4.5.4.10 QPushButton* WidgetImage : :launchButton [private]

Définition à la ligne 110 du fichier widgetImage.h.

4.5.4.11 QGroupBox* WidgetImage : :loadGroupBox [private]

Définition à la ligne 94 du fichier widgetImage.h.

4.5.4.12 QLineEdit* WidgetImage : :loadLineEdit [private]

Définition à la ligne 101 du fichier widgetImage.h.

4.5.4.13 QCheckBox* WidgetImage : :lowCheckBox [private]

Définition à la ligne 114 du fichier widgetImage.h.

4.5.4.14 QPushButton* WidgetImage : :navLoadButton [private]

Définition à la ligne 109 du fichier widgetImage.h.

4.5.4.15 QPushButton* WidgetImage : :navSaveButton [private]

Définition à la ligne 111 du fichier widgetImage.h.

4.5.4.16 QLineEdit* WidgetImage : :nbIterationLineEdit [private]

Définition à la ligne 103 du fichier widgetImage.h.

4.5.4.17 QGroupBox* WidgetImage : :outputGroupBox [private]

Définition à la ligne 99 du fichier widgetImage.h.

4.5.4.18 QImage* WidgetImage : :outputImage [private]

Définition à la ligne 120 du fichier widgetImage.h.

4.5.4.19 QLabel* WidgetImage : :outputLabel [private]

Définition à la ligne 117 du fichier widgetImage.h.

4.5.4.20 QCheckBox* WidgetImage : :raiseCheckBox [private]

Définition à la ligne 113 du fichier widgetImage.h.

4.5.4.21 `QGroupBox* WidgetImage : :saveGroupBox` [private]

Définition à la ligne 97 du fichier widgetImage.h.

4.5.4.22 `QLineEdit* WidgetImage : :saveLineEdit` [private]

Définition à la ligne 102 du fichier widgetImage.h.

4.5.4.23 `QLineEdit* WidgetImage : :sigmaRLineEdit` [private]

Définition à la ligne 105 du fichier widgetImage.h.

4.5.4.24 `QLineEdit* WidgetImage : :sigmaSLineEdit` [private]

Définition à la ligne 104 du fichier widgetImage.h.

La documentation de cette classe a été générée à partir des fichiers suivants :

- InterfaceQt/[widgetImage.h](#)
- InterfaceQt/[widgetImage.cpp](#)

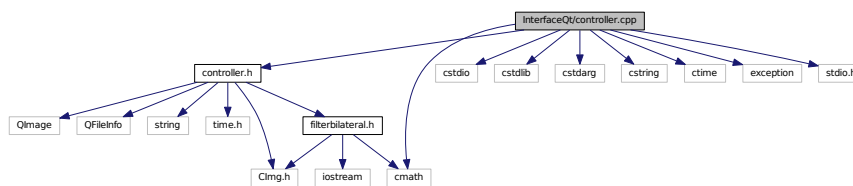
Chapitre 5

Documentation des fichiers

5.1 Référence du fichier InterfaceQt/controller.cpp

```
#include "controller.h"
```

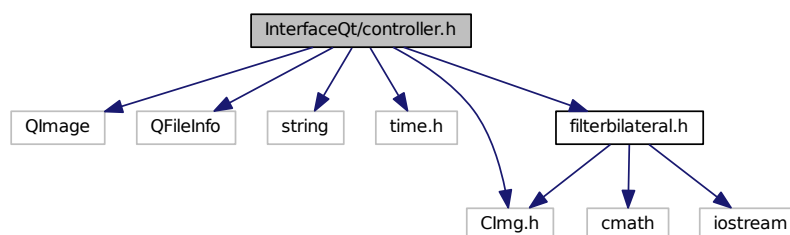
Graphe des dépendances par inclusion de controller.cpp :



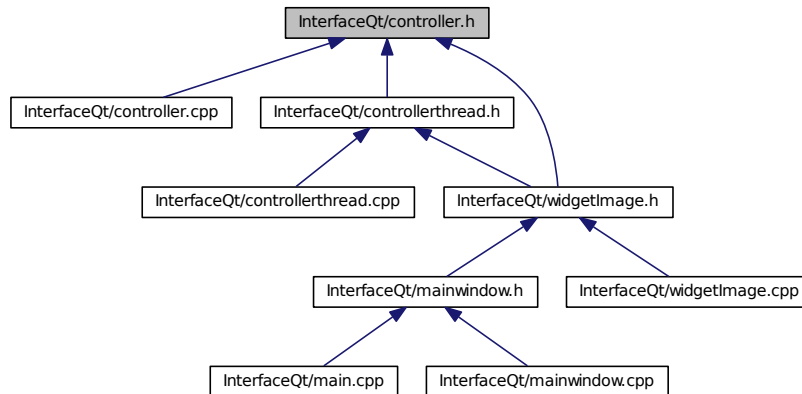
5.2 Référence du fichier InterfaceQt/controller.h

```
#include <QImage>
#include <QFileInfo>
#include <string>
#include <time.h>
#include "filterbilateral.h"
#include "CImg.h"
```

Graphe des dépendances par inclusion de controller.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



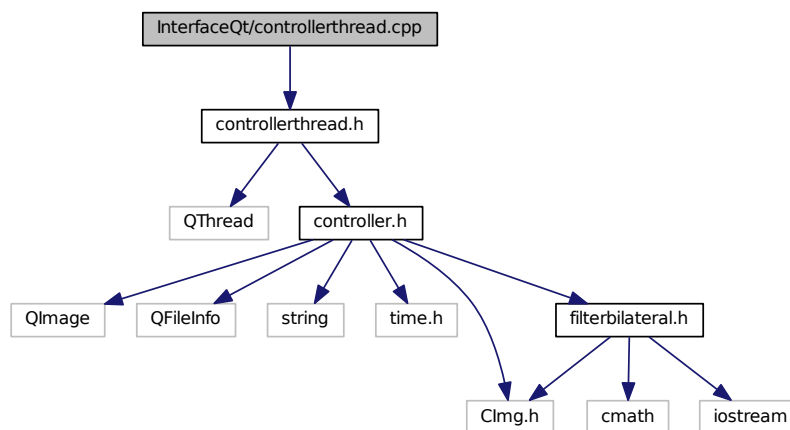
Classes

— class [Controller](#)

5.3 Référence du fichier InterfaceQt/controllerthread.cpp

```
#include "controllerthread.h"
```

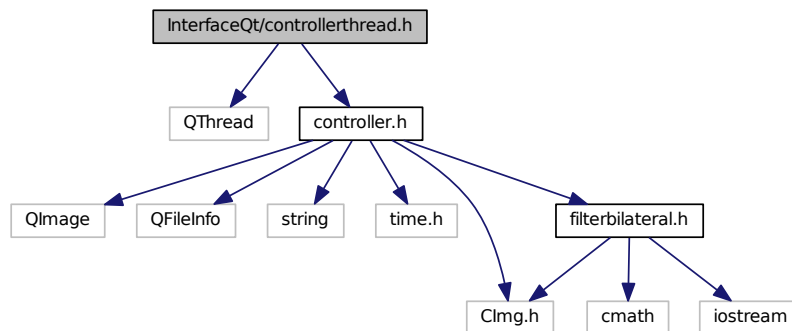
Grappe des dépendances par inclusion de controllerthread.cpp :



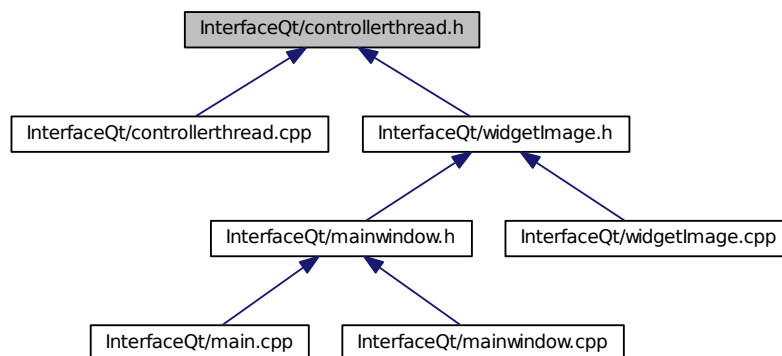
5.4 Référence du fichier InterfaceQt/controllerthread.h

```
#include <QThread>
#include "controller.h"
```

Graphe des dépendances par inclusion de controllerthread.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

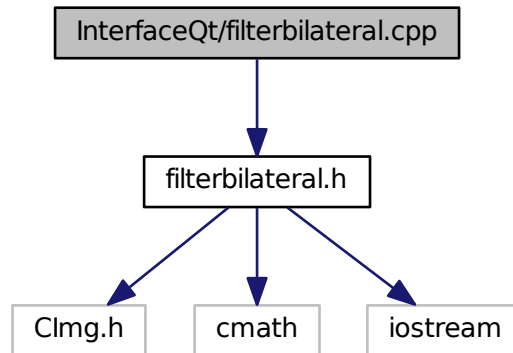
— class [ControllerThread](#)

Classe permettant d'exécuter la fonction **changeDetailsUser** du [Controller](#) dans un thread.

5.5 Référence du fichier InterfaceQt/filterbilateral.cpp

```
#include "filterbilateral.h"
```

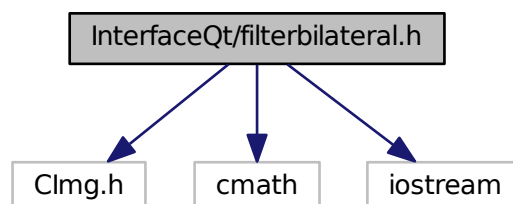
Graphe des dépendances par inclusion de filterbilateral.cpp :



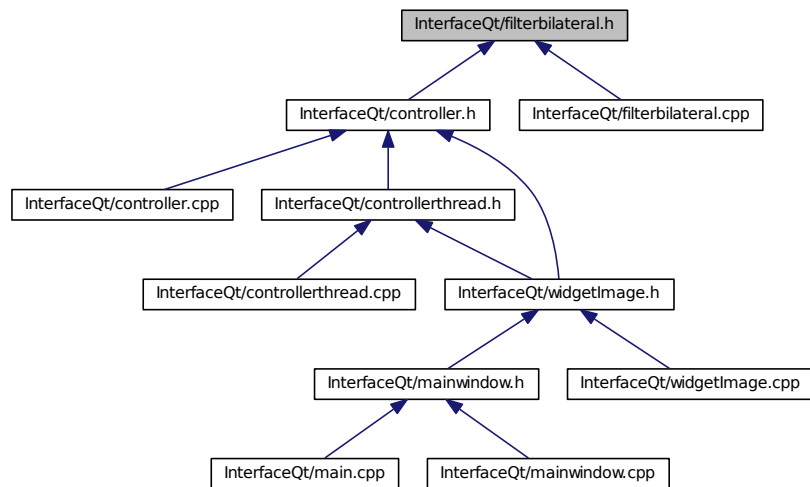
5.6 Référence du fichier InterfaceQt/filterbilateral.h

```
#include "CImg.h"  
#include <cmath>  
#include <iostream>
```

Graphe des dépendances par inclusion de filterbilateral.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

— class [FilterBilateral](#)

Classe permettant de manipuler des images avec un filtre bilatéral Cette classe permet d'appliquer un filtre bilatéral sur une image en niveau de gris ou RGB.

Macros

— #define [_USE_MATH_DEFINES](#)

5.6.1 Documentation des macros

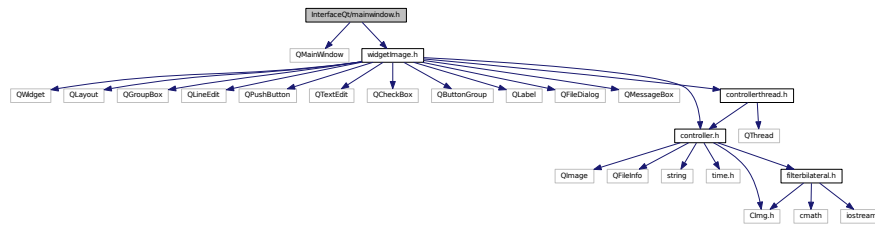
5.6.1.1 #define [_USE_MATH_DEFINES](#)

Définition à la ligne 4 du fichier filterbilateral.h.

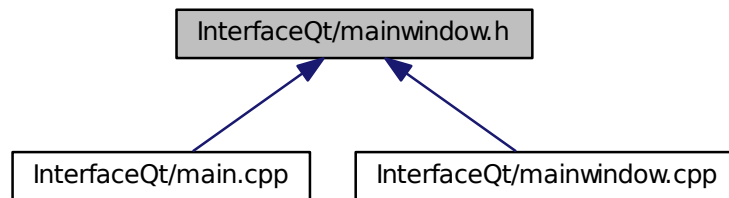
5.7 Référence du fichier InterfaceQt/main.cpp

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


Graphe des dépendances par inclusion de mainwindow.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



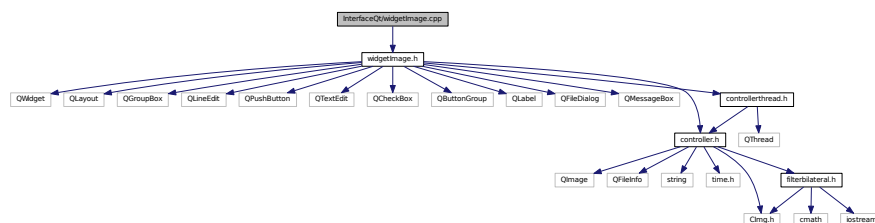
Classes

- class [MainWindow](#)
Classe de l'interface de la fenêtre principale.

5.10 Référence du fichier InterfaceQt/widgetImage.cpp

```
#include "widgetImage.h"
```

Graphe des dépendances par inclusion de widgetImage.cpp :



5.11 Référence du fichier InterfaceQt/widgetImage.h

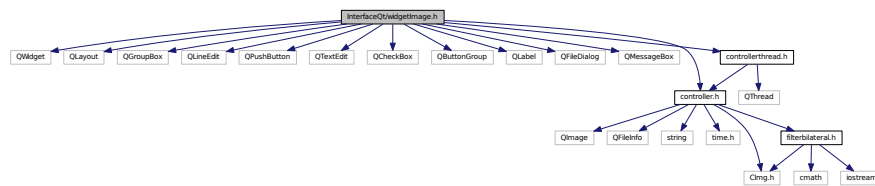
```
#include <QWidget>
```

```

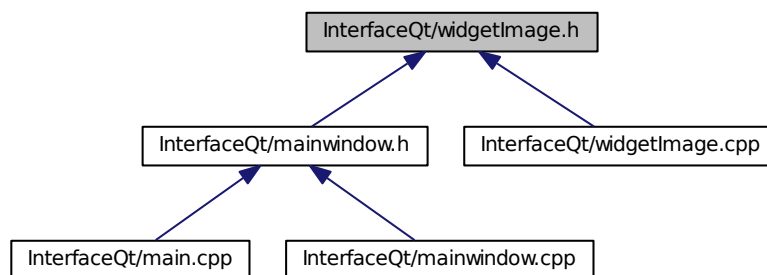
#include <QLayout>
#include <QGroupBox>
#include <QLineEdit>
#include <QPushButton>
#include <QTextEdit>
#include <QCheckBox>
#include <QButtonGroup>
#include <QLabel>
#include <QFileDialog>
#include <QMessageBox>
#include "controller.h"
#include "controllerthread.h"

```

Graphe des dépendances par inclusion de widgetImage.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

— class [WidgetImage](#)

Classe du widget principal Ce widget est contenu dans la fenêtre principale.

Index

`_USE_MATH_DEFINES`

`filterbilateral.h`, 43

`~Controller`

`Controller`, 9

`~ControllerThread`

`ControllerThread`, 16

`~FilterBilateral`

`FilterBilateral`, 20

`~MainWindow`

`MainWindow`, 26

`~WidgetImage`

`WidgetImage`, 30

`alpha`

`Controller`, 12

`alphaLineEdit`

`WidgetImage`, 35

`alphaLower`

`Controller`, 12

`alphaRaise`

`Controller`, 12

`applyFilter`

`FilterBilateral`, 20

`applyFilterNdG`

`FilterBilateral`, 20

`applyFilterRGB`

`FilterBilateral`, 21

`beta`

`Controller`, 12

`betaLineEdit`

`WidgetImage`, 35

`betaLower`

`Controller`, 12

`betaRaise`

`Controller`, 13

`bilateralFilter`

`Controller`, 9

`cThread`

`WidgetImage`, 35

`calculTermine`

`WidgetImage`, 30

`changeDetailsUser`

`Controller`, 9

`complete`

`ControllerThread`, 16

`constructNewImage`

`Controller`, 10

`Controller`, 7

`~Controller`, 9

`alpha`, 12

`alphaLower`, 12

`alphaRaise`, 12

`beta`, 12

`betaLower`, 12

`betaRaise`, 13

`bilateralFilter`, 9

`changeDetailsUser`, 9

`constructNewImage`, 10

`Controller`, 8

`detail`, 13

`fileNameInput`, 13

`fsigmaR`, 13

`fsigmaS`, 13

`getAlpha`, 10

`getBeta`, 10

`getDetail`, 10

`getFileNameInput`, 10

`getNbIteration`, 10

`getSigmaR`, 10

`getSigmaS`, 10

`iteration`, 13

`nbIteration`, 13

`setAlpha`, 11

`setBeta`, 11

`setDetail`, 11

`setFileNameInput`, 11

`setNbIteration`, 11

`setSigmaR`, 12

`setSigmaS`, 12

`sigmaR`, 13

`sigmaS`, 13

`controller`

`ControllerThread`, 17

`ControllerThread`, 13

`~ControllerThread`, 16

`complete`, 16

`controller`, 17

`ControllerThread`, 16

`run`, 16

`createConnection`

`WidgetImage`, 31

`createDetailsGroupBox`

`WidgetImage`, 31

`createInfoGroupBox`

`WidgetImage`, 32

`createInputGroupBox`

`WidgetImage`, 32

- createLoadGroupBox
 - WidgetImage, 32
- createOutputGroupBox
 - WidgetImage, 33
- createSaveGroupBox
 - WidgetImage, 33
- detail
 - Controller, 13
- detailsGroupBox
 - WidgetImage, 35
- distanceEuclidienne
 - FilterBilateral, 21
- fSigmaR
 - FilterBilateral, 22
- fSigmaS
 - FilterBilateral, 23
- fileNameInput
 - Controller, 13
 - WidgetImage, 35
- FilterBilateral, 17
 - ~FilterBilateral, 20
 - applyFilter, 20
 - applyFilterNdG, 20
 - applyFilterRGB, 21
 - distanceEuclidienne, 21
 - fSigmaR, 22
 - fSigmaS, 23
 - FilterBilateral, 19, 20
 - get_fSigmaR, 21
 - get_fSigmaS, 21
 - get_img, 21
 - height, 23
 - img, 23
 - loiGaussienne, 22
 - set_fSigmaR, 22
 - set_fSigmaS, 22
 - set_img, 22
 - width, 23
- filterbilateral.h
 - _USE_MATH_DEFINES, 43
- fsigmaR
 - Controller, 13
- fsigmaS
 - Controller, 13
- get_fSigmaR
 - FilterBilateral, 21
- get_fSigmaS
 - FilterBilateral, 21
- get_img
 - FilterBilateral, 21
- getAlpha
 - Controller, 10
- getBeta
 - Controller, 10
- getDetail
 - Controller, 10
- getFileNameInput
 - Controller, 10
- getNbIteration
 - Controller, 10
- getSigmaR
 - Controller, 10
- getSigmaS
 - Controller, 10
- height
 - FilterBilateral, 23
- img
 - FilterBilateral, 23
- infoGroupBox
 - WidgetImage, 35
- inputGroupBox
 - WidgetImage, 35
- inputImage
 - WidgetImage, 35
- inputLabel
 - WidgetImage, 35
- InterfaceQt/controller.cpp, 39
- InterfaceQt/controller.h, 39
- InterfaceQt/controllerthread.cpp, 40
- InterfaceQt/controllerthread.h, 40
- InterfaceQt/filterbilateral.cpp, 41
- InterfaceQt/filterbilateral.h, 42
- InterfaceQt/main.cpp, 43
- InterfaceQt/mainwindow.cpp, 44
- InterfaceQt/mainwindow.h, 44
- InterfaceQt/widgetImage.cpp, 45
- InterfaceQt/widgetImage.h, 45
- iteration
 - Controller, 13
- launchButton
 - WidgetImage, 36
- launchButtonClicked
 - WidgetImage, 33
- loadGroupBox
 - WidgetImage, 36
- loadLineEdit
 - WidgetImage, 36
- loiGaussienne
 - FilterBilateral, 22
- lowCheckBox
 - WidgetImage, 36
- main
 - main.cpp, 44
- main.cpp
 - main, 44
- mainWidget
 - MainWindow, 26
- MainWindow, 23
 - ~MainWindow, 26
- mainWidget, 26
- MainWindow, 26

- navLoadButton
 - WidgetImage, 36
- navLoadButtonClicked
 - WidgetImage, 34
- navSaveButton
 - WidgetImage, 36
- navSaveButtonClicked
 - WidgetImage, 34
- nbIteration
 - Controller, 13
- nbIterationLineEdit
 - WidgetImage, 36
- outputGroupBox
 - WidgetImage, 36
- outputImage
 - WidgetImage, 36
- outputLabel
 - WidgetImage, 36
- raiseCheckBox
 - WidgetImage, 36
- run
 - ControllerThread, 16
- saveGroupBox
 - WidgetImage, 36
- saveLineEdit
 - WidgetImage, 37
- set_fSigmaR
 - FilterBilateral, 22
- set_fSigmaS
 - FilterBilateral, 22
- set_img
 - FilterBilateral, 22
- setAlpha
 - Controller, 11
- setBeta
 - Controller, 11
- setDetail
 - Controller, 11
- setFileNameInput
 - Controller, 11
- setNbIteration
 - Controller, 11
- setSigmaR
 - Controller, 12
- setSigmaS
 - Controller, 12
- sigmaR
 - Controller, 13
- sigmaRLineEdit
 - WidgetImage, 37
- sigmaS
 - Controller, 13
- sigmaSLineEdit
 - WidgetImage, 37
- WidgetImage, 26
 - ~WidgetImage, 30
 - alphaLineEdit, 35
 - betaLineEdit, 35
 - cThread, 35
 - calculTermine, 30
 - createConnection, 31
 - createDetailsGroupBox, 31
 - createInfoGroupBox, 32
 - createInputGroupBox, 32
 - createLoadGroupBox, 32
 - createOutputGroupBox, 33
 - createSaveGroupBox, 33
 - detailsGroupBox, 35
 - fileNameInput, 35
 - infoGroupBox, 35
 - inputGroupBox, 35
 - inputImage, 35
 - inputLabel, 35
 - launchButton, 36
 - launchButtonClicked, 33
 - loadGroupBox, 36
 - loadLineEdit, 36
 - lowCheckBox, 36
 - navLoadButton, 36
 - navLoadButtonClicked, 34
 - navSaveButton, 36
 - navSaveButtonClicked, 34
 - nbIterationLineEdit, 36
 - outputGroupBox, 36
 - outputImage, 36
 - outputLabel, 36
 - raiseCheckBox, 36
 - saveGroupBox, 36
 - saveLineEdit, 37
 - sigmaRLineEdit, 37
 - sigmaSLineEdit, 37
 - WidgetImage, 30
- width
 - FilterBilateral, 23