

iStuffTracking
0.2.5

Generated by Doxygen 1.8.4

Wed Jul 24 2013 22:02:46

Contents

1	Todo List	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	File Index	9
5.1	File List	9
6	Namespace Documentation	11
6.1	boost Namespace Reference	11
6.2	boost::serialization Namespace Reference	11
6.2.1	Function Documentation	11
6.2.1.1	load	11
6.2.1.2	save	11
6.3	IStuff Namespace Reference	11
6.3.1	Typedef Documentation	12
6.3.1.1	Features	12
7	Class Documentation	13
7.1	IStuff::Database Class Reference	13
7.1.1	Detailed Description	14
7.1.2	Constructor & Destructor Documentation	14
7.1.2.1	Database	14
7.1.2.2	~Database	14
7.1.3	Member Function Documentation	14
7.1.3.1	build	14
7.1.3.2	load	14
7.1.3.3	match	15

7.1.3.4	save	15
7.1.4	Member Data Documentation	15
7.1.4.1	dbName	15
7.1.4.2	dbPath	15
7.1.4.3	descriptorDB	15
7.1.4.4	keypointDB	15
7.1.4.5	labelDB	15
7.1.4.6	MATCH_THRESHOLD	15
7.1.4.7	matcher	15
7.1.4.8	MIN_INLIER_RATIO	16
7.1.4.9	NNDR_RATIO	16
7.2	ISuff::DBCException Class Reference	16
7.2.1	Detailed Description	16
7.2.2	Member Function Documentation	16
7.2.2.1	what	16
7.3	ISuff::DBLoadingException Class Reference	16
7.3.1	Detailed Description	17
7.3.2	Member Function Documentation	17
7.3.2.1	what	17
7.4	ISuff::DBSavingException Class Reference	17
7.4.1	Detailed Description	17
7.4.2	Member Function Documentation	17
7.4.2.1	what	17
7.5	ISuff::FakableQueue Class Reference	18
7.5.1	Detailed Description	18
7.5.2	Constructor & Destructor Documentation	19
7.5.2.1	FakableQueue	19
7.5.2.2	~FakableQueue	19
7.5.3	Member Function Documentation	19
7.5.3.1	dequeue	19
7.5.3.2	discard	19
7.5.3.3	enqueue	19
7.5.3.4	getStarter	20
7.5.3.5	start	20
7.5.4	Member Data Documentation	20
7.5.4.1	queue_mutex	20
7.5.4.2	real_queue	20
7.5.4.3	saved_queue	20
7.5.4.4	TAG	20
7.6	ISuff::Label Struct Reference	20

7.6.1	Detailed Description	21
7.6.2	Constructor & Destructor Documentation	21
7.6.2.1	Label	21
7.6.3	Member Function Documentation	21
7.6.3.1	operator==	21
7.6.4	Member Data Documentation	21
7.6.4.1	color	21
7.6.4.2	name	21
7.6.4.3	position	21
7.7	ISuff::Manager Class Reference	21
7.7.1	Detailed Description	22
7.7.2	Constructor & Destructor Documentation	23
7.7.2.1	Manager	23
7.7.2.2	~Manager	23
7.7.3	Member Function Documentation	23
7.7.3.1	elaborateFrame	23
7.7.3.2	getObject	23
7.7.3.3	paintObject	23
7.7.3.4	sendMessage	24
7.7.3.5	setDatabase	24
7.7.3.6	setObject	24
7.7.4	Member Data Documentation	24
7.7.4.1	actual_object	24
7.7.4.2	frames_tracked_count	25
7.7.4.3	MSG_RECOGNITION_END	25
7.7.4.4	MSG_RECOGNITION_START	25
7.7.4.5	MSG_TRACKING_END	25
7.7.4.6	MSG_TRACKING_START	25
7.7.4.7	object_update	25
7.7.4.8	RECOGNITION_PERIOD	25
7.7.4.9	recognizer	25
7.7.4.10	TAG	25
7.7.4.11	tracker	25
7.8	ISuff::Object Class Reference	25
7.8.1	Detailed Description	26
7.8.2	Constructor & Destructor Documentation	26
7.8.2.1	Object	26
7.8.2.2	~Object	26
7.8.3	Member Function Documentation	27
7.8.3.1	addLabel	27

7.8.3.2	empty	27
7.8.3.3	getLabels	27
7.8.3.4	paint	27
7.8.4	Member Data Documentation	27
7.8.4.1	labels	27
7.8.4.2	TAG	28
7.9	IStuff::Recognizer Class Reference	28
7.9.1	Detailed Description	28
7.9.2	Constructor & Destructor Documentation	29
7.9.2.1	Recognizer	29
7.9.2.2	~Recognizer	29
7.9.3	Member Function Documentation	29
7.9.3.1	backgroundRecognizeFrame	29
7.9.3.2	isRunning	29
7.9.3.3	recognizeFrame	29
7.9.3.4	sendMessage	30
7.9.3.5	setDatabase	30
7.9.3.6	setRunning	30
7.9.4	Member Data Documentation	30
7.9.4.1	m_matcher	30
7.9.4.2	m_running	30
7.9.4.3	m_thread	30
7.9.4.4	TAG	31
7.10	IStuff::Tracker Class Reference	31
7.10.1	Detailed Description	32
7.10.2	Constructor & Destructor Documentation	32
7.10.2.1	Tracker	32
7.10.2.2	~Tracker	32
7.10.3	Member Function Documentation	32
7.10.3.1	backgroundTrackFrame	32
7.10.3.2	calcFeatures	32
7.10.3.3	calcFeatures	33
7.10.3.4	isRunning	33
7.10.3.5	sendMessage	33
7.10.3.6	setRunning	34
7.10.3.7	trackFrame	34
7.10.3.8	updateObject	34
7.10.4	Member Data Documentation	34
7.10.4.1	IMG_RESIZE	34
7.10.4.2	LK_WINDOW	35

7.10.4.3	m_detector	35
7.10.4.4	m_features	35
7.10.4.5	m_frame	35
7.10.4.6	m_matcher	35
7.10.4.7	m_object	35
7.10.4.8	m_object_mutex	35
7.10.4.9	m_running	35
7.10.4.10	m_saved_features	35
7.10.4.11	m_thread	35
7.10.4.12	NEAREST_FEATURES_COUNT	35
7.10.4.13	TAG	35
8	File Documentation	37
8.1	src/IStuff/database.cpp File Reference	37
8.1.1	Detailed Description	37
8.2	src/IStuff/database.h File Reference	37
8.2.1	Detailed Description	38
8.2.2	Variable Documentation	38
8.2.2.1	debug	38
8.3	src/IStuff/fakable_queue.cpp File Reference	38
8.4	src/IStuff/fakable_queue.h File Reference	39
8.4.1	Detailed Description	39
8.4.2	Variable Documentation	39
8.4.2.1	debug	39
8.5	src/IStuff/manager.cpp File Reference	40
8.6	src/IStuff/manager.h File Reference	40
8.6.1	Detailed Description	40
8.6.2	Variable Documentation	41
8.6.2.1	debug	41
8.6.2.2	hl_debug	41
8.7	src/IStuff/object.cpp File Reference	41
8.8	src/IStuff/object.h File Reference	41
8.8.1	Detailed Description	41
8.8.2	Variable Documentation	42
8.8.2.1	debug	42
8.9	src/IStuff/recognizer.cpp File Reference	42
8.10	src/IStuff/recognizer.h File Reference	42
8.10.1	Detailed Description	43
8.10.2	Variable Documentation	43
8.10.2.1	debug	43

8.11	src/IStuff/serialize_opencv.h File Reference	43
8.11.1	Detailed Description	44
8.12	src/IStuff/tracker.cpp File Reference	44
8.13	src/IStuff/tracker.h File Reference	44
8.13.1	Detailed Description	45
8.13.2	Variable Documentation	45
8.13.2.1	debug	45
8.14	src/main.cpp File Reference	45
8.14.1	Detailed Description	45
8.14.2	Function Documentation	46
8.14.2.1	main	46
8.14.2.2	printHelp	46
8.15	src/main.h File Reference	46
8.15.1	Detailed Description	47
8.15.2	Function Documentation	47
8.15.2.1	main	47
8.15.2.2	printHelp	47
8.15.3	Variable Documentation	47
8.15.3.1	debug	47
8.15.3.2	hl_debug	47

Chapter 1

Todo List

Member `IStuff::FakableQueue::discard ()`

Find a more explicative name.

Member `IStuff::Object::addLabel (const Label)`

Update description.

Member `IStuff::Object::empty () const`

Describe.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

boost	11
boost::serialization	11
IStuff	11

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

IStuff::Database	13
exception	
IStuff::DBCreationException	16
IStuff::DBLoadingException	16
IStuff::DBSavingException	17
IStuff::FakableQueue	18
IStuff::Label	20
IStuff::Manager	21
IStuff::Object	25
IStuff::Recognizer	28
IStuff::Tracker	31

Chapter 4

Class Index

4.1 Class List

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

ISuff::Database	13
ISuff::DBCreationException	16
ISuff::DBLoadingException	16
ISuff::DBSavingException	17
ISuff::FakableQueue	
Class used to manage a synchronized double queue	18
ISuff::Label	
Label relative to a view of an Object	20
ISuff::Manager	
Class to manage the joint 3D Object recognition and tracking	21
ISuff::Object	
Class used to represent a three dimensional object	25
ISuff::Recognizer	
Class used to recognize objects in a video stream	28
ISuff::Tracker	
Class used to track objects in a video stream	31

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

src/main.cpp	
Main file	45
src/main.h	
Main file's header	46
src/IStuff/database.cpp	
Definition for Database class	37
src/IStuff/database.h	
Library for Database class	37
src/IStuff/fakable_queue.cpp	
src/IStuff/fakable_queue.h	
Header file for IStuff::FakableQueue	39
src/IStuff/manager.cpp	
src/IStuff/manager.h	
Header file relative to the class IStuff::Manager	40
src/IStuff/object.cpp	
src/IStuff/object.h	
Header file relative to the class IStuff::Object	41
src/IStuff/recognizer.cpp	
src/IStuff/recognizer.h	
Header file relative to the class IStuff::Recognizer	42
src/IStuff/serialize_opencv.h	
Serialization support for various opencv classes thought boost	43
src/IStuff/tracker.cpp	
src/IStuff/tracker.h	
Header file relative to the class IStuff::Tracker	44

Chapter 6

Namespace Documentation

6.1 boost Namespace Reference

Namespaces

- [serialization](#)

6.2 boost::serialization Namespace Reference

Functions

- `template<class Archive >`
`void save (Archive &ar, const cv::Mat &m, const unsigned int version)`
- `template<class Archive >`
`void load (Archive &ar, cv::Mat &m, const unsigned int version)`

6.2.1 Function Documentation

6.2.1.1 `template<class Archive > void boost::serialization::load (Archive &ar, cv::Mat &m, const unsigned int version)`

Definition at line 32 of file `serialize_opencv.h`.

6.2.1.2 `template<class Archive > void boost::serialization::save (Archive &ar, const cv::Mat &m, const unsigned int version)`

Definition at line 17 of file `serialize_opencv.h`.

6.3 IStuff Namespace Reference

Classes

- class [Database](#)
- class [DBCcreationException](#)
- class [DBLoadingException](#)
- class [DBSavingException](#)
- class [FakableQueue](#)

- Class used to manage a synchronized double queue.*
 - class [Manager](#)
 - Class to manage the joint 3D [Object](#) recognition and tracking.*
 - struct [Label](#)
 - [Label](#) relative to a view of an [Object](#).*
 - class [Object](#)
 - Class used to represent a three dimensional object.*
 - class [Recognizer](#)
 - Class used to recognize objects in a video stream.*
 - class [Tracker](#)
 - Class used to track objects in a video stream.*

Typedefs

- typedef std::vector< cv::Point2f > [Features](#)
 - An alias for a std::vector of cv::Point, used for tracking.*

6.3.1 Typedef Documentation

6.3.1.1 typedef std::vector<cv::Point2f> IStuff::Features

An alias for a std::vector of cv::Point, used for tracking.

Definition at line 29 of file tracker.h.

Chapter 7

Class Documentation

7.1 IStuff::Database Class Reference

```
#include <database.h>
```

Public Member Functions

- [Database](#) (std::string, std::string)
Constructor.
- virtual [~Database](#) ()
Destructor.
- [Object match](#) (cv::Mat)
Search for descriptors matching in passed frame.

Private Member Functions

- void [build](#) (std::string)
Creates the database from the sample images.
- void [load](#) ()
Load existing database and fill the structures.
- void [save](#) ()
Writes the database to a set of files in the default directory.

Private Attributes

- const float [NNDR_RATIO](#) = 0.6
- const float [MIN_INLIER_RATIO](#) = 0.5
- const int [MATCH_THRESHOLD](#) = 14
- std::string [dbPath](#)
- std::string [dbName](#)
- cv::FlannBasedMatcher [matcher](#)
- std::vector< std::vector< [Label](#) > > [labelDB](#)
- std::vector< std::vector
 < cv::KeyPoint > > [keypointDB](#)
- std::vector< cv::Mat > [descriptorDB](#)

7.1.1 Detailed Description

Author

Mattia Rizzini

Version

0.1.4

Date

2013-07-14

Definition at line 49 of file database.h.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 Database::Database (std::string , std::string)

Constructor.

If the name passed matches an existing DB it loads it, otherwise it creates it

Parameters

<i>in</i>	<i>_dbName</i>	The name of the DB to be loaded
<i>in</i>	<i>imagesPath</i>	The position of the sample images from which the descriptors are to be taken

Definition at line 26 of file database.cpp.

7.1.2.2 Database::~Database () [virtual]

Destructor.

Definition at line 55 of file database.cpp.

7.1.3 Member Function Documentation

7.1.3.1 void Database::build (std::string) [private]

Creates the database from the sample images.

Loaded the images contained in the argument path associate to every image sample its keypoints and descriptors. Also loads the label positions in the samples. Saves everything in three structures and to a set of files

Parameters

<i>in</i>	<i>imagesPath</i>	The path containing the source images
-----------	-------------------	---------------------------------------

Definition at line 205 of file database.cpp.

7.1.3.2 void Database::load () [private]

Load existing database and fill the structures.

Definition at line 324 of file database.cpp.

7.1.3.3 Object Database::match (cv::Mat)

Search for descriptors matching in passed frame.

Given an image, searches for descriptor matches in the database and returns an object containing the estimated label positions

Parameters

<i>in</i>	<i>frame</i>	The image to search into
-----------	--------------	--------------------------

Return values

<i>An</i>	Object containing an association between the labels and the positions in which every label is found
-----------	---

Definition at line 67 of file database.cpp.

7.1.3.4 void Database::save () [private]

Writes the database to a set of files in the default directory.

Definition at line 441 of file database.cpp.

7.1.4 Member Data Documentation

7.1.4.1 std::string IStuff::Database::dbName [private]

Definition at line 56 of file database.h.

7.1.4.2 std::string IStuff::Database::dbPath [private]

Definition at line 55 of file database.h.

7.1.4.3 std::vector< cv::Mat > IStuff::Database::descriptorDB [private]

Definition at line 61 of file database.h.

7.1.4.4 std::vector< std::vector< cv::KeyPoint > > IStuff::Database::keypointDB [private]

Definition at line 60 of file database.h.

7.1.4.5 std::vector< std::vector< Label > > IStuff::Database::labelDB [private]

Definition at line 59 of file database.h.

7.1.4.6 const int IStuff::Database::MATCH_THRESHOLD = 14 [private]

Definition at line 53 of file database.h.

7.1.4.7 cv::FlannBasedMatcher IStuff::Database::matcher [private]

Definition at line 58 of file database.h.

7.1.4.8 `const float IStuff::Database::MIN_INLIER_RATIO = 0.5` `[private]`

Definition at line 52 of file database.h.

7.1.4.9 `const float IStuff::Database::NNDR_RATIO = 0.6` `[private]`

Definition at line 51 of file database.h.

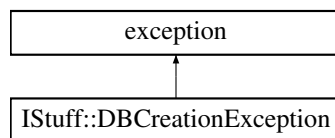
The documentation for this class was generated from the following files:

- [src/IStuff/database.h](#)
- [src/IStuff/database.cpp](#)

7.2 IStuff::DBCreationException Class Reference

```
#include <database.h>
```

Inheritance diagram for IStuff::DBCreationException:



Public Member Functions

- virtual const char * [what](#) () const throw ()

7.2.1 Detailed Description

Definition at line 75 of file database.h.

7.2.2 Member Function Documentation

7.2.2.1 `virtual const char* IStuff::DBCreationException::what () const throw ()` `[inline], [virtual]`

Definition at line 76 of file database.h.

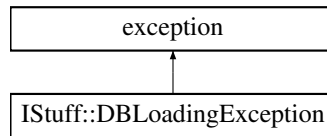
The documentation for this class was generated from the following file:

- [src/IStuff/database.h](#)

7.3 IStuff::DBLoadingException Class Reference

```
#include <database.h>
```

Inheritance diagram for IStuff::DBLoadingException:



Public Member Functions

- virtual const char * [what](#) () const throw ()

7.3.1 Detailed Description

Definition at line 81 of file database.h.

7.3.2 Member Function Documentation

7.3.2.1 virtual const char* IStuff::DBLoadingException::what () const throw) [inline], [virtual]

Definition at line 82 of file database.h.

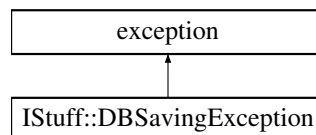
The documentation for this class was generated from the following file:

- src/IStuff/[database.h](#)

7.4 IStuff::DBSavingException Class Reference

```
#include <database.h>
```

Inheritance diagram for IStuff::DBSavingException:



Public Member Functions

- virtual const char * [what](#) () const throw ()

7.4.1 Detailed Description

Definition at line 87 of file database.h.

7.4.2 Member Function Documentation

7.4.2.1 virtual const char* IStuff::DBSavingException::what () const throw) [inline], [virtual]

Definition at line 88 of file database.h.

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

- [src/IStuff/database.h](#)

7.5 IStuff::FakableQueue Class Reference

Class used to manage a synchronized double queue.

```
#include <fakable_queue.h>
```

Public Member Functions

- [FakableQueue](#) ()
Constructor of this class.
- virtual [~FakableQueue](#) ()
- void [enqueue](#) (cv::Mat)
Adds a frame to the queue.
- void [start](#) (cv::Mat)
Starts the queue, enabling the enqueueement.
- void [discard](#) ()
Replaces the 'real' queue with the 'saved' one.
- cv::Mat [dequeue](#) ()
Returns and removes a frame from the 'real' queue.
- cv::Mat [getStarter](#) ()
Returns the frame that started the queue.

Private Attributes

- std::queue< cv::Mat > [real_queue](#)
- std::queue< cv::Mat > [saved_queue](#)
- boost::mutex [queue_mutex](#)

Static Private Attributes

- static const char [TAG](#) [] = "Fkq"

7.5.1 Detailed Description

Class used to manage a synchronized double queue.

This class is used by [IStuff::Tracker](#) and it allows to address problems caused by alternated recognizations:

This queue manages two queues: one "real", used normally, from where frames are enqueued and dequeued and one "fake", or "saved", used to store the frames regarding only the last recognition, where frames are just enqueued and which is substituted to the real queue when the last recognition ends.

Author

Maurizio Zucchelli

Version

0.1.2

Date

2013-07-18

Definition at line 23 of file fakable_queue.h.

7.5.2 Constructor & Destructor Documentation**7.5.2.1 FakableQueue::FakableQueue ()**

Constructor of this class.

Definition at line 30 of file fakable_queue.cpp.

7.5.2.2 FakableQueue::~~FakableQueue () [virtual]

Definition at line 33 of file fakable_queue.cpp.

7.5.3 Member Function Documentation**7.5.3.1 Mat FakableQueue::dequeue ()**

Returns and removes a frame from the 'real' queue.

Exceptions

<i>out_of_range</i>	If the queue is empty.
---------------------	------------------------

Returns

The frame in front of the 'real' queue.

Definition at line 103 of file fakable_queue.cpp.

7.5.3.2 void FakableQueue::discard ()

Replaces the 'real' queue with the 'saved' one.

Todo Find a more explicative name.

Definition at line 85 of file fakable_queue.cpp.

7.5.3.3 void FakableQueue::enqueue (cv::Mat)

Adds a frame to the queue.

If the queue has been started, the frame is added to both the 'real' and the 'saved' queue; otherwise, nothing happens.

Parameters

<i>in</i>	<i>frame</i>	The frame to be inserted.
-----------	--------------	---------------------------

Definition at line 45 of file fakable_queue.cpp.

7.5.3.4 Mat FakableQueue::getStarter ()

Returns the frame that started the queue.

Returns

The frame that started the queue.

Definition at line 126 of file fakable_queue.cpp.

7.5.3.5 void FakableQueue::start (cv::Mat)

Starts the queue, enabling the enqueueement.

This operation resets the 'saved' queue and then performs an enqueue with the given frame.

Parameters

<i>in</i>	<i>frame</i>	The frame starter of the queue.
-----------	--------------	---------------------------------

Definition at line 68 of file fakable_queue.cpp.

7.5.4 Member Data Documentation

7.5.4.1 boost::mutex IStuff::FakableQueue::queue_mutex [private]

Definition at line 31 of file fakable_queue.h.

7.5.4.2 std::queue<cv::Mat> IStuff::FakableQueue::real_queue [private]

Definition at line 29 of file fakable_queue.h.

7.5.4.3 std::queue<cv::Mat> IStuff::FakableQueue::saved_queue [private]

Definition at line 29 of file fakable_queue.h.

7.5.4.4 const char FakableQueue::TAG = "Fkq" [static], [private]

Definition at line 27 of file fakable_queue.h.

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

- src/IStuff/fakable_queue.h
- src/IStuff/fakable_queue.cpp

7.6 IStuff::Label Struct Reference

[Label](#) relative to a view of an [Object](#).

```
#include <object.h>
```

Public Member Functions

- [Label](#) (std::string _name, cv::Point2f _position, cv::Scalar _color)
- bool [operator==](#) (const [Label](#) &other) const

Public Attributes

- std::string [name](#)
- cv::Point2f [position](#)
- cv::Scalar [color](#)

7.6.1 Detailed Description

[Label](#) relative to a view of an [Object](#).

Definition at line 26 of file [object.h](#).

7.6.2 Constructor & Destructor Documentation

7.6.2.1 IStuff::Label::Label (std::string *_name*, cv::Point2f *_position*, cv::Scalar *_color*) [\[inline\]](#)

Definition at line 31 of file [object.h](#).

7.6.3 Member Function Documentation

7.6.3.1 bool IStuff::Label::operator== (const Label & *other*) const [\[inline\]](#)

Definition at line 35 of file [object.h](#).

7.6.4 Member Data Documentation

7.6.4.1 cv::Scalar IStuff::Label::color

Definition at line 29 of file [object.h](#).

7.6.4.2 std::string IStuff::Label::name

Definition at line 27 of file [object.h](#).

7.6.4.3 cv::Point2f IStuff::Label::position

Definition at line 28 of file [object.h](#).

The documentation for this struct was generated from the following file:

- [src/IStuff/object.h](#)

7.7 IStuff::Manager Class Reference

Class to manage the joint 3D [Object](#) recognition and tracking.

```
#include <manager.h>
```

Public Member Functions

- [Manager](#) ()
Constructs the class.
- virtual [~Manager](#) ()
- void [setDatabase](#) ([Database](#) *)
Changes the [IStuff::Database](#) used to identify the [IStuff::Object](#).
- [Object](#) [getObject](#) ()
Returns the current description of the [IStuff::Object](#).
- void [elaborateFrame](#) ([cv::Mat](#))
Elaborates a frame, searching for the [IStuff::Object](#).
- [cv::Mat](#) [paintObject](#) ([cv::Mat](#))
Paints the various masks of the [IStuff::Object](#) on the frame.
- void [sendMessage](#) (int, void *, void * = NULL)
Method to send messages to this [IStuff::Manager](#).

Static Public Attributes

- static const int [MSG_RECOGNITION_START](#) = 1
- static const int [MSG_RECOGNITION_END](#) = 2
- static const int [MSG_TRACKING_START](#) = 3
- static const int [MSG_TRACKING_END](#) = 4

Private Member Functions

- void [setObject](#) (const [Object](#))
Sets the [IStuff::Object](#) of this [IStuff::Manager](#).

Private Attributes

- int [frames_tracked_count](#)
When this reaches [RECOGNITION_PERIOD](#), a new recognition is done.
- boost::shared_mutex [object_update](#)
- [Object](#) [actual_object](#)
- [Recognizer](#) [recognizer](#)
- [Tracker](#) [tracker](#)

Static Private Attributes

- static const char [TAG](#) [] = "Mng"
- static const int [RECOGNITION_PERIOD](#) = 30

7.7.1 Detailed Description

Class to manage the joint 3D [Object](#) recognition and tracking.

Author

Maurizio Zucchelli

Version

0.3.0

Date

2013-07-16

Definition at line 28 of file manager.h.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 Manager::Manager ()

Constructs the class.

Definition at line 24 of file manager.cpp.

7.7.2.2 Manager::~Manager () [virtual]

Definition at line 29 of file manager.cpp.

7.7.3 Member Function Documentation

7.7.3.1 void Manager::elaborateFrame (cv::Mat)

Elaborates a frame, searching for the [IStuff::Object](#).

This function alternates the recognition to the tracking, making a new recognition every [IStuff::Manager::RECOGNITION_PERIOD](#) frames.

Parameters

<i>frame</i>	The frame to be analyzed.
--------------	---------------------------

Definition at line 82 of file manager.cpp.

7.7.3.2 Object Manager::getObject ()

Returns the current description of the [IStuff::Object](#).

Returns

The current description of IStuff::the [Object](#).

Definition at line 66 of file manager.cpp.

7.7.3.3 Mat Manager::paintObject (cv::Mat)

Paints the various masks of the [IStuff::Object](#) on the frame.

Parameters

<i>in</i>	<i>frame</i>	The frame on which the IStuff::Object must be painted.
-----------	--------------	--

Returns

A copy of the input frame, with the [IStuff::Object](#) painted on it.

Definition at line 111 of file manager.cpp.

7.7.3.4 void Manager::sendMessage (int *msg*, void * *data*, void * *reply_to* = NULL)

Method to send messages to this [IStuff::Manager](#).

Managed messages:

[IStuff::Manager::MSG_RECOGNITION_START](#) data: cv::Mat

This message is forwarded to both the [IStuff::Recognizer](#) (to make it start the recognition) and the [IStuff::Tracker](#) (to alert it).

This also resets the counter of frames tracked from last recognition.

[IStuff::Manager::MSG_RECOGNITION_END](#) data: [IStuff::Object](#)

This message is forwarded to the [IStuff::Tracker](#), to update its [IStuff::Object](#).

Parameters

<i>in</i>	<i>msg</i>	The message identifier.
<i>in</i>	<i>data</i>	The data related to the message.
<i>in</i>	<i>reply_to</i>	The sender of the message (optional).

Definition at line 135 of file manager.cpp.

7.7.3.5 void Manager::setDatabase ([Database](#) * *database*)

Changes the [IStuff::Database](#) used to identify the [IStuff::Object](#).

This means that with high probability a different [IStuff::Object](#) will be searched for: the next elaboration must be a recognition.

Parameters

<i>in</i>	<i>database</i>	The new IStuff::Database to be used.
-----------	-----------------	--

Definition at line 53 of file manager.cpp.

7.7.3.6 void Manager::setObject (const [Object](#) *object*) [private]

Sets the [IStuff::Object](#) of this [IStuff::Manager](#).

Parameters

<i>in</i>	<i>object</i>	The new IStuff::Object .
-----------	---------------	--

Definition at line 39 of file manager.cpp.

7.7.4 Member Data Documentation

7.7.4.1 [Object](#) [IStuff::Manager::actual_object](#) [private]

Definition at line 47 of file manager.h.

7.7.4.2 `int IStuff::Manager::frames_tracked_count` `[private]`

When this reaches RECOGNITION_PERIOD, a new recognition is done.

Definition at line 44 of file manager.h.

7.7.4.3 `const int IStuff::Manager::MSG_RECOGNITION_END = 2` `[static]`

Definition at line 33 of file manager.h.

7.7.4.4 `const int IStuff::Manager::MSG_RECOGNITION_START = 1` `[static]`

Definition at line 32 of file manager.h.

7.7.4.5 `const int IStuff::Manager::MSG_TRACKING_END = 4` `[static]`

Definition at line 35 of file manager.h.

7.7.4.6 `const int IStuff::Manager::MSG_TRACKING_START = 3` `[static]`

Definition at line 34 of file manager.h.

7.7.4.7 `boost::shared_mutex IStuff::Manager::object_update` `[private]`

Definition at line 45 of file manager.h.

7.7.4.8 `const int IStuff::Manager::RECOGNITION_PERIOD = 30` `[static], [private]`

Definition at line 39 of file manager.h.

7.7.4.9 `Recognizer IStuff::Manager::recognizer` `[private]`

Definition at line 48 of file manager.h.

7.7.4.10 `const char Manager::TAG = "Mng"` `[static], [private]`

Definition at line 38 of file manager.h.

7.7.4.11 `Tracker IStuff::Manager::tracker` `[private]`

Definition at line 49 of file manager.h.

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

- [src/IStuff/manager.h](#)
- [src/IStuff/manager.cpp](#)

7.8 IStuff::Object Class Reference

Class used to represent a three dimensional object.

```
#include <object.h>
```

Public Member Functions

- [Object](#) ()
Constructs a new object.
- virtual [~Object](#) ()
- void [addLabel](#) (const [Label](#))
Changes a [ISuff::Label](#) of this [ISuff::Object](#) to represent a different mask.
- bool [empty](#) () const
- std::vector< [Label](#) > [getLabels](#) () const
Returns a list of all the [ISuff::Label](#) associated to this object.
- cv::Mat [paint](#) (cv::Mat)
Paints the various [ISuff::Label](#) of the [ISuff::Object](#) on the frame.

Private Attributes

- std::vector< [Label](#) > [labels](#)

Static Private Attributes

- static const char [TAG](#) [] = "Obj"

7.8.1 Detailed Description

Class used to represent a three dimensional object.

Author

Maurizio Zucchelli

Version

0.1.0

Date

2013-07-16

Definition at line 41 of file object.h.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 [Object::Object](#) ()

Constructs a new object.

Definition at line 23 of file object.cpp.

7.8.2.2 [Object::~~Object](#) () [virtual]

Definition at line 26 of file object.cpp.

7.8.3 Member Function Documentation

7.8.3.1 void Object::addLabel (const Label *label*)

Changes a [IStuff::Label](#) of this [IStuff::Object](#) to represent a different mask.

Todo Update description.

If the [IStuff::Label](#) isn't currently part of the [IStuff::Object](#), it is added.

Parameters

in	<i>label</i>	the label name
in	<i>mask</i>	the label position
in	<i>color</i>	the label color

Definition at line 41 of file object.cpp.

7.8.3.2 bool Object::empty () const

Todo Describe.

Returns

Definition at line 54 of file object.cpp.

7.8.3.3 vector< Label > Object::getLabels () const

Returns a list of all the [IStuff::Label](#) associated to this object.

Returns

A std::vector containing the [IStuff::Label](#).

Definition at line 64 of file object.cpp.

7.8.3.4 Mat Object::paint (cv::Mat)

Paints the various [IStuff::Label](#) of the [IStuff::Object](#) on the frame.

Parameters

in	<i>frame</i>	The frame on which the IStuff::Object must be painted.
----	--------------	--

Returns

A copy of the input frame, with the [IStuff::Object](#) painted on it.

Definition at line 78 of file object.cpp.

7.8.4 Member Data Documentation

7.8.4.1 std::vector< Label > IStuff::Object::labels [private]

Definition at line 47 of file object.h.

7.8.4.2 `const char Object::TAG = "Obj" [static], [private]`

Definition at line 45 of file `object.h`.

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

- `src/IStuff/object.h`
- `src/IStuff/object.cpp`

7.9 IStuff::Recognizer Class Reference

Class used to recognize objects in a video stream.

```
#include <recognizer.h>
```

Public Member Functions

- [Recognizer](#) ()
Constructs a structure used to find 3D objects inside a video stream.
- virtual [~Recognizer](#) ()
- void [setDatabase](#) ([Database](#) *)
Associates a [IStuff::Database](#) to this [IStuff::Recognizer](#).
- bool [isRunning](#) () const
Checks whether this [IStuff::Recognizer](#) has a thread up and running.
- [Object](#) [recognizeFrame](#) (cv::Mat)
Recognizes an [IStuff::Object](#) into a frame.
- bool [backgroundRecognizeFrame](#) (cv::Mat, [Manager](#) *)
Method to do the recognition process in a separate thread.
- void [sendMessage](#) (int, void *, void *=NULL)
Method to send messages to this [IStuff::Recognizer](#).

Private Member Functions

- void [setRunning](#) (bool)

Private Attributes

- std::auto_ptr< boost::thread > [m_thread](#)
- bool [m_running](#)
- [Database](#) * [m_matcher](#)

Static Private Attributes

- static const char [TAG](#) [] = "Rec"

7.9.1 Detailed Description

Class used to recognize objects in a video stream.

This class manages a thread receiving a frame from an input stream, analyzing it to find some kind of 3D object and then updating the data used by the requester to track it.

Author

Maurizio Zucchelli

Version

0.2.1

Date

2013-07-14

Definition at line 28 of file recognizer.h.

7.9.2 Constructor & Destructor Documentation

7.9.2.1 Recognizer::Recognizer ()

Constructs a structure used to find 3D objects inside a video stream.

Definition at line 27 of file recognizer.cpp.

7.9.2.2 Recognizer::~Recognizer () [virtual]

Definition at line 36 of file recognizer.cpp.

7.9.3 Member Function Documentation

7.9.3.1 bool Recognizer::backgroundRecognizeFrame (cv::Mat , Manager *)

Method to do the recognition process in a separate thread.

Parameters

<i>in</i>	<i>frame</i>	The frame to be searched for an IStuff::Object .
<i>in</i>	<i>reference</i>	The reference to the IStuff::Manager to inform of the result.

Returns

`true` if the thread is started, `false` if it was already running.

Definition at line 98 of file recognizer.cpp.

7.9.3.2 bool Recognizer::isRunning () const

Checks whether this [IStuff::Recognizer](#) has a thread up and running.

Returns

`true` if recognizing, `false` otherwise.

Definition at line 63 of file recognizer.cpp.

7.9.3.3 Object Recognizer::recognizeFrame (cv::Mat)

Recognizes an [IStuff::Object](#) into a frame.

Parameters

<i>in</i>	<i>frame</i>	The frame to be searched for an IStuff::Object .
-----------	--------------	--

Returns

The [IStuff::Object](#) found inside the given frame.

Definition at line 77 of file recognizer.cpp.

7.9.3.4 `void Recognizer::sendMessage (int msg, void * data, void * reply_to = NULL)`

Method to send messages to this [IStuff::Recognizer](#).

Managed messages:

[IStuff::Manager::MSG_RECOGNITION_START](#) data: cv::Mat

This causes the recognition process to start.

Parameters

<i>in</i>	<i>msg</i>	The message identifier.
<i>in</i>	<i>data</i>	The data related to the message.
<i>in</i>	<i>reply_to</i>	The sender of the message (optional).

Definition at line 138 of file recognizer.cpp.

7.9.3.5 `void Recognizer::setDatabase (Database * matcher)`

Associates a [IStuff::Database](#) to this [IStuff::Recognizer](#).

Parameters

<i>in</i>	<i>matcher</i>	The new matcher to be used.
-----------	----------------	-----------------------------

Definition at line 46 of file recognizer.cpp.

7.9.3.6 `void Recognizer::setRunning (bool running)` [private]

Definition at line 51 of file recognizer.cpp.

7.9.4 Member Data Documentation

7.9.4.1 `Database* IStuff::Recognizer::m_matcher` [private]

Definition at line 37 of file recognizer.h.

7.9.4.2 `bool IStuff::Recognizer::m_running` [private]

Definition at line 35 of file recognizer.h.

7.9.4.3 `std::auto_ptr<boost::thread> IStuff::Recognizer::m_thread` [private]

Definition at line 34 of file recognizer.h.

7.9.4.4 `const char Recognizer::TAG = "Rec" [static], [private]`

Definition at line 32 of file recognizer.h.

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

- src/IStuff/recognizer.h
- src/IStuff/recognizer.cpp

7.10 IStuff::Tracker Class Reference

Class used to track objects in a video stream.

```
#include <tracker.h>
```

Public Member Functions

- [Tracker](#) ()
Constructs a structure used to track 3D objects inside a video stream.
- virtual [~Tracker](#) ()
- bool [isRunning](#) () const
Checks whether this [IStuff::Tracker](#) has a thread up and running.
- [Object](#) [trackFrame](#) (cv::Mat)
Tracks the current [IStuff::Object](#) between the last frame and this one.
- void [sendMessage](#) (int, void *, void *=NULL)
Method to send messages to this [IStuff::Tracker](#).

Private Member Functions

- void [setRunning](#) (bool)
Method used by this [IStuff::Tracker](#)'s thread to mark itself as running.
- [Features](#) [calcFeatures](#) (cv::Mat)
Method to calculate the [IStuff::Features](#) used to track [IStuff::Object](#) between frames.
- [Features](#) [calcFeatures](#) (cv::Mat, cv::Mat, [Features](#) *)
Method to track [IStuff::Features](#) between frames.
- [Object](#) [updateObject](#) ([Features](#), [Features](#), [Object](#))
Function to update an [IStuff::Object](#) from an old position to its new one.
- bool [backgroundTrackFrame](#) (cv::Mat, [Manager](#) *)
Method to do the tracking process in a separate thread.

Private Attributes

- std::auto_ptr< boost::thread > [m_thread](#)
- bool [m_running](#) = false
- boost::mutex [m_object_mutex](#)
- [Object](#) [m_object](#)
- cv::Mat [m_frame](#)
- [Features](#) [m_features](#)
- [Features](#) [m_saved_features](#)
- cv::Ptr< cv::FeatureDetector > [m_detector](#)
- cv::Ptr< cv::DescriptorMatcher > [m_matcher](#)

Static Private Attributes

- static const char [TAG](#) [] = "Trk"
- static const int [NEAREST_FEATURES_COUNT](#) = 10
- static const float constexpr [IMG_RESIZE](#) = .5
- static const cv::Size [LK_WINDOW](#) = cv::Size(15, 15)

7.10.1 Detailed Description

Class used to track objects in a video stream.

Author

Maurizio Zucchelli

Version

0.8.0

Date

2013-07-17

Definition at line 36 of file tracker.h.

7.10.2 Constructor & Destructor Documentation

7.10.2.1 Tracker::Tracker ()

Constructs a structure used to track 3D objects inside a video stream.

Definition at line 25 of file tracker.cpp.

7.10.2.2 Tracker::~Tracker () [virtual]

Definition at line 34 of file tracker.cpp.

7.10.3 Member Function Documentation

7.10.3.1 bool Tracker::backgroundTrackFrame (cv::Mat , Manager *) [private]

Method to do the tracking process in a separate thread.

Parameters

in	<i>frame</i>	The frame to be tracked for an IStuff::Object .
in	<i>reference</i>	The reference to the IStuff::Manager to inform of the result.

Returns

`true` if the thread is started, `false` if it was already running.

Definition at line 235 of file tracker.cpp.

7.10.3.2 Features Tracker::calcFeatures (cv::Mat frame) [private]

Method to calculate the [IStuff::Features](#) used to track [IStuff::Object](#) between frames.

Parameters

<i>in</i>	<i>frame</i>	The frame on which calculate the features.
-----------	--------------	--

Returns

The [IStuff::Features](#) detected on the given frame.

Definition at line 106 of file tracker.cpp.

7.10.3.3 Features Tracker::calcFeatures (*cv::Mat old_frame*, *cv::Mat new_frame*, *Features * old_features*) [private]

Method to track [IStuff::Features](#) between frames.

Parameters

<i>in</i>	<i>old_frame</i>	The frame relative to the given IStuff::Features .
<i>in</i>	<i>new_frame</i>	The frame where to track the IStuff::Features .
<i>in, out</i>	<i>old_features</i>	The old IStuff::Features , returned erased of the untracked features.

Returns

The [IStuff::Features](#) of the old frame relative to the new frame.

Definition at line 133 of file tracker.cpp.

7.10.3.4 bool Tracker::isRunning () const

Checks whether this [IStuff::Tracker](#) has a thread up and running.

Returns

`true` if tracking, `false` otherwise.

Definition at line 56 of file tracker.cpp.

7.10.3.5 void Tracker::sendMessage (*int msg*, *void * data*, *void * reply_to = NULL*)

Method to send messages to this [IStuff::Tracker](#).

Managed messages:

[IStuff::Manager::MSG_RECOGNITION_START](#) data: *cv::Mat*

This message's handling is synchronized.

The frame received is downscaled, then [IStuff::Features](#) are calculated and the actual [IStuff::Object](#) is updated according to this frame. The [IStuff::Features](#) are saved for use when the recognition ends.

[IStuff::Manager::MSG_RECOGNITION_END](#) data: *IStuff::Object*

This message's handling is synchronized.

This causes the [IStuff::Tracker](#) to actualize the new [IStuff::Object](#) by tracking it from the saved [IStuff::Features](#) and the current ones.

Parameters

in	<i>msg</i>	The message identifier.
in	<i>data</i>	The data related to the message.
in	<i>reply_to</i>	The sender of the message (optional).

Definition at line 283 of file tracker.cpp.

7.10.3.6 void Tracker::setRunning (bool *running*) [private]

Method used by this [IStuff::Tracker](#)'s thread to mark itself as running.

Parameters

in	<i>running</i>	The status to assign to the thread.
----	----------------	-------------------------------------

Definition at line 44 of file tracker.cpp.

7.10.3.7 Object Tracker::trackFrame (cv::Mat)

Tracks the current [IStuff::Object](#) between the last frame and this one.

This method is synchronized for its whole duration, this avoids other object updates to happen between the calculations and the internal data update.

Parameters

in	<i>new_frame</i>	The frame where to track the IStuff::Object .
----	------------------	---

Returns

The new [IStuff::Object](#) tracked in the new frame.

Definition at line 73 of file tracker.cpp.

7.10.3.8 Object Tracker::updateObject (Features *old_features*, Features *new_features*, Object *old_object*) [private]

Function to update an [IStuff::Object](#) from an old position to its new one.

This method calculates the new position by mediating the movement of the nearest [IStuff::Tracker::NEAREST_FEATURES_COUNT](#) features to every point of every [IStuff::Label](#) of the [IStuff::Object](#).

Parameters

in	<i>old_features</i>	The IStuff::Features relative to the IStuff::Object .
in	<i>new_features</i>	The IStuff::Features for the new IStuff::Object .
in	<i>old_object</i>	The IStuff::Object to be updated.

Returns

The new [IStuff::Object](#), moved according to the [IStuff::Features](#).

Definition at line 180 of file tracker.cpp.

7.10.4 Member Data Documentation

7.10.4.1 const float constexpr IStuff::Tracker::IMG_RESIZE = .5 [static], [private]

Definition at line 43 of file tracker.h.

7.10.4.2 `const Size Tracker::LK_WINDOW = cv::Size(15, 15)` [static], [private]

Definition at line 44 of file tracker.h.

7.10.4.3 `cv::Ptr<cv::FeatureDetector> IStuff::Tracker::m_detector` [private]

Definition at line 55 of file tracker.h.

7.10.4.4 **Features** `IStuff::Tracker::m_features` [private]

Definition at line 52 of file tracker.h.

7.10.4.5 `cv::Mat IStuff::Tracker::m_frame` [private]

Definition at line 51 of file tracker.h.

7.10.4.6 `cv::Ptr<cv::DescriptorMatcher> IStuff::Tracker::m_matcher` [private]

Definition at line 56 of file tracker.h.

7.10.4.7 **Object** `IStuff::Tracker::m_object` [private]

Definition at line 50 of file tracker.h.

7.10.4.8 `boost::mutex IStuff::Tracker::m_object_mutex` [private]

Definition at line 48 of file tracker.h.

7.10.4.9 `bool IStuff::Tracker::m_running = false` [private]

Definition at line 47 of file tracker.h.

7.10.4.10 **Features** `IStuff::Tracker::m_saved_features` [private]

Definition at line 52 of file tracker.h.

7.10.4.11 `std::auto_ptr<boost::thread> IStuff::Tracker::m_thread` [private]

Definition at line 46 of file tracker.h.

7.10.4.12 `const int IStuff::Tracker::NEAREST_FEATURES_COUNT = 10` [static], [private]

Definition at line 42 of file tracker.h.

7.10.4.13 `const char Tracker::TAG = "Trk"` [static], [private]

Definition at line 40 of file tracker.h.

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

- [src/IStuff/tracker.h](#)
- [src/IStuff/tracker.cpp](#)

Chapter 8

File Documentation

8.1 src/Stuff/database.cpp File Reference

Definition for Database class.

```
#include "database.h"
```

8.1.1 Detailed Description

Definition for Database class.

Definition in file [database.cpp](#).

8.2 src/Stuff/database.h File Reference

Library for Database class.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <numeric>
#include "object.h"
#include "opencv2/core/core.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/features2d/features2d.hpp"
#include "opencv2/calib3d/calib3d.hpp"
#include "opencv2/nonfree/nonfree.hpp"
#include "opencv2/video/video.hpp"
#include "boost/filesystem.hpp"
#include "boost/lambda/bind.hpp"
#include "boost/foreach.hpp"
#include "boost/archive/binary_oarchive.hpp"
#include "boost/archive/binary_iarchive.hpp"
#include "boost/random.hpp"
#include "boost/serialization/vector.hpp"
#include "serialize_opencv.h"
#include "boost/lexical_cast.hpp"
```

Classes

- class [IStuff::Database](#)
- class [IStuff::DBCreationException](#)
- class [IStuff::DBLoadingException](#)
- class [IStuff::DBSavingException](#)

Namespaces

- [IStuff](#)

Constant Groups

- [IStuff](#)

Variables

- bool [debug](#)

8.2.1 Detailed Description

Library for Database class.

Author

Mattia Rizzini

Version

0.1.3

Date

2013-07-14

Definition in file [database.h](#).

8.2.2 Variable Documentation

8.2.2.1 bool debug

Definition at line 19 of file main.h.

8.3 src/IStuff/fakable_queue.cpp File Reference

```
#include "fakable_queue.h"
```

8.4 src/ISuff/fakable_queue.h File Reference

Header file for [ISuff::FakableQueue](#).

```
#include <iostream>
#include <queue>
#include <boost/thread.hpp>
#include "opencv2/core/core.hpp"
```

Classes

- class [ISuff::FakableQueue](#)

Class used to manage a synchronized double queue.

Namespaces

- [ISuff](#)

Constant Groups

- [ISuff](#)

Variables

- bool [debug](#)

8.4.1 Detailed Description

Header file for [ISuff::FakableQueue](#).

Author

Maurizio Zucchelli

Version

0.1.0

Date

2013-07-18

Definition in file [fakable_queue.h](#).

8.4.2 Variable Documentation

8.4.2.1 bool debug

Definition at line 19 of file main.h.

8.5 src/IStuff/manager.cpp File Reference

```
#include "manager.h"
```

8.6 src/IStuff/manager.h File Reference

Header file relative to the class [IStuff::Manager](#).

```
#include <iostream>
#include "opencv2/core/core.hpp"
#include <boost/thread.hpp>
#include "object.h"
#include "database.h"
#include "recognizer.h"
#include "tracker.h"
```

Classes

- class [IStuff::Manager](#)
Class to manage the joint 3D [Object](#) recognition and tracking.

Namespaces

- [IStuff](#)

Constant Groups

- [IStuff](#)

Variables

- bool [debug](#)
- bool [hl_debug](#)

8.6.1 Detailed Description

Header file relative to the class [IStuff::Manager](#).

Author

Maurizio Zucchelli

Version

0.3.0

Date

2013-07-16

Definition in file [manager.h](#).

8.6.2 Variable Documentation

8.6.2.1 bool debug

Definition at line 19 of file main.h.

8.6.2.2 bool hl_debug

Definition at line 19 of file main.h.

8.7 src/IStuff/object.cpp File Reference

```
#include "object.h"
```

8.8 src/IStuff/object.h File Reference

Header file relative to the class [IStuff::Object](#).

```
#include <iostream>
#include <map>
#include <vector>
#include "opencv2/core/core.hpp"
```

Classes

- struct [IStuff::Label](#)
[Label](#) relative to a view of an [Object](#).
- class [IStuff::Object](#)
Class used to represent a three dimensional object.

Namespaces

- [IStuff](#)

Constant Groups

- [IStuff](#)

Variables

- bool [debug](#)

8.8.1 Detailed Description

Header file relative to the class [IStuff::Object](#).

Author

Maurizio Zucchelli
Mattia Rizzini

Version

0.1.1

Date

2013-07-16

Definition in file [object.h](#).

8.8.2 Variable Documentation

8.8.2.1 bool debug

Definition at line 19 of file main.h.

8.9 src/IStuff/recognizer.cpp File Reference

```
#include "recognizer.h"
```

8.10 src/IStuff/recognizer.h File Reference

Header file relative to the class [IStuff::Recognizer](#).

```
#include <iostream>
#include <boost/thread.hpp>
#include <boost/chrono.hpp>
#include "opencv2/imgproc/imgproc.hpp"
#include "object.h"
#include "database.h"
#include "manager.h"
```

Classes

- class [IStuff::Recognizer](#)
Class used to recognize objects in a video stream.

Namespaces

- [IStuff](#)

Constant Groups

- [IStuff](#)

Variables

- bool [debug](#)

8.10.1 Detailed Description

Header file relative to the class [IStuff::Recognizer](#).

Author

Maurizio Zucchelli

Version

0.2.1

Date

2013-07-14

Definition in file [recognizer.h](#).

8.10.2 Variable Documentation

8.10.2.1 bool debug

Definition at line 19 of file main.h.

8.11 src/IStuff/serialize_opencv.h File Reference

Serialization support for various opencv classes thought boost.

```
#include "boost/serialization/serialization.hpp"
#include "boost/serialization/split_free.hpp"
#include "boost/serialization/vector.hpp"
```

Namespaces

- [boost](#)
- [boost::serialization](#)

Constant Groups

- [boost](#)
- [boost::serialization](#)

Functions

- `template<class Archive >`
void [boost::serialization::save](#) (Archive &ar, const cv::Mat &m, const unsigned int version)
- `template<class Archive >`
void [boost::serialization::load](#) (Archive &ar, cv::Mat &m, const unsigned int version)

8.11.1 Detailed Description

Serialization support for various opencv classes through boost.

Author

Mattia Rizzini

Version

0.1.1

Date

2013-07-15

Definition in file [serialize_opencv.h](#).

8.12 src/IStuff/tracker.cpp File Reference

```
#include "tracker.h"
```

8.13 src/IStuff/tracker.h File Reference

Header file relative to the class [IStuff::Tracker](#).

```
#include <iostream>
#include <map>
#include <boost/thread.hpp>
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/video/video.hpp"
#include "opencv2/nonfree/nonfree.hpp"
#include "object.h"
#include "fakable_queue.h"
#include "manager.h"
```

Classes

- class [IStuff::Tracker](#)
Class used to track objects in a video stream.

Namespaces

- [IStuff](#)

Constant Groups

- [IStuff](#)

Typedefs

- typedef std::vector< cv::Point2f > [IStuff::Features](#)
An alias for a std::vector of cv::Point, used for tracking.

Variables

- bool [debug](#)

8.13.1 Detailed Description

Header file relative to the class [IStuff::Tracker](#).

Author

Maurizio Zucchelli

Version

0.8.0

Date

2013-07-17

Definition in file [tracker.h](#).

8.13.2 Variable Documentation

8.13.2.1 bool debug

Definition at line 19 of file main.h.

8.14 src/main.cpp File Reference

Main file.

```
#include "main.h"
```

Functions

- int [main](#) (int argc, char *argv[])
Main function.
- void [printHelp](#) ()
Function to display the help message.

8.14.1 Detailed Description

Main file.

Author

Maurizio Zucchelli
Mattia Rizzini

Version

0.1.2

Date

2013-07-13

Definition in file [main.cpp](#).

8.14.2 Function Documentation

8.14.2.1 `int main (int argc, char * argv[])`

Main function.

Parameters

<i>argc</i>	
<i>argv</i> []	

Returns

Definition at line 24 of file main.cpp.

8.14.2.2 `void printHelp ()`

Function to display the help message.

Definition at line 199 of file main.cpp.

8.15 `src/main.h` File Reference

Main file's header.

```
#include <iostream>
#include "opencv2/core/core.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "IStuff/manager.h"
```

Functions

- `int main (int, char **)`
- `void printHelp ()`

Function to display the help message.

Variables

- bool [debug](#)
- bool [hl_debug](#)

8.15.1 Detailed Description

Main file's header.

Author

Maurizio Zucchelli

Version

0.1.0

Date

2013-07-13

Definition in file [main.h](#).

8.15.2 Function Documentation

8.15.2.1 `int main (int , char **)`

8.15.2.2 `void printHelp ()`

Function to display the help message.

Definition at line 199 of file main.cpp.

8.15.3 Variable Documentation

8.15.3.1 `bool debug`

Definition at line 19 of file main.h.

8.15.3.2 `bool hl_debug`

Definition at line 19 of file main.h.

Index

- ~Database
 - IStuff::Database, [14](#)
- ~FakableQueue
 - IStuff::FakableQueue, [19](#)
- ~Manager
 - IStuff::Manager, [23](#)
- ~Object
 - IStuff::Object, [26](#)
- ~Recognizer
 - IStuff::Recognizer, [29](#)
- ~Tracker
 - IStuff::Tracker, [32](#)
- actual_object
 - IStuff::Manager, [24](#)
- addLabel
 - IStuff::Object, [27](#)
- backgroundRecognizeFrame
 - IStuff::Recognizer, [29](#)
- backgroundTrackFrame
 - IStuff::Tracker, [32](#)
- boost, [11](#)
- boost::serialization, [11](#)
 - load, [11](#)
 - save, [11](#)
- build
 - IStuff::Database, [14](#)
- calcFeatures
 - IStuff::Tracker, [32](#), [33](#)
- color
 - IStuff::Label, [21](#)
- Database
 - IStuff::Database, [14](#)
- database.h
 - debug, [38](#)
- dbName
 - IStuff::Database, [15](#)
- dbPath
 - IStuff::Database, [15](#)
- debug
 - database.h, [38](#)
 - fakable_queue.h, [39](#)
 - main.h, [47](#)
 - manager.h, [41](#)
 - object.h, [42](#)
 - recognizer.h, [43](#)
 - tracker.h, [45](#)
- dequeue
 - IStuff::FakableQueue, [19](#)
- descriptorDB
 - IStuff::Database, [15](#)
- discard
 - IStuff::FakableQueue, [19](#)
- elaborateFrame
 - IStuff::Manager, [23](#)
- empty
 - IStuff::Object, [27](#)
- enqueue
 - IStuff::FakableQueue, [19](#)
- fakable_queue.h
 - debug, [39](#)
- FakableQueue
 - IStuff::FakableQueue, [19](#)
- Features
 - IStuff, [12](#)
- frames_tracked_count
 - IStuff::Manager, [24](#)
- getLabels
 - IStuff::Object, [27](#)
- getObject
 - IStuff::Manager, [23](#)
- getStarter
 - IStuff::FakableQueue, [19](#)
- hl_debug
 - main.h, [47](#)
 - manager.h, [41](#)
- IMG_RESIZE
 - IStuff::Tracker, [34](#)
- IStuff, [11](#)
 - Features, [12](#)
- IStuff::DBCException, [16](#)
 - what, [16](#)
- IStuff::DBLoadingException, [16](#)
 - what, [17](#)
- IStuff::DBSavingException, [17](#)
 - what, [17](#)
- IStuff::Database, [13](#)
 - ~Database, [14](#)
 - build, [14](#)
 - Database, [14](#)
 - dbName, [15](#)
 - dbPath, [15](#)
 - descriptorDB, [15](#)

- keypointDB, 15
- labelDB, 15
- load, 14
- MATCH_THRESHOLD, 15
- MIN_INLIER_RATIO, 15
- match, 14
- matcher, 15
- NNDR_RATIO, 16
- save, 15
- IStuff::FakableQueue, 18
 - ~FakableQueue, 19
 - dequeue, 19
 - discard, 19
 - enqueue, 19
 - FakableQueue, 19
 - getStarter, 19
 - queue_mutex, 20
 - real_queue, 20
 - saved_queue, 20
 - start, 20
 - TAG, 20
- IStuff::Label, 20
 - color, 21
 - Label, 21
 - name, 21
 - operator==, 21
 - position, 21
- IStuff::Manager, 21
 - ~Manager, 23
 - actual_object, 24
 - elaborateFrame, 23
 - frames_tracked_count, 24
 - getObject, 23
 - MSG_TRACKING_END, 25
 - Manager, 23
 - object_update, 25
 - paintObject, 23
 - RECOGNITION_PERIOD, 25
 - recognizer, 25
 - sendMessage, 24
 - setDatabase, 24
 - setObject, 24
 - TAG, 25
 - tracker, 25
- IStuff::Object, 25
 - ~Object, 26
 - addLabel, 27
 - empty, 27
 - getLabels, 27
 - labels, 27
 - Object, 26
 - paint, 27
 - TAG, 27
- IStuff::Recognizer, 28
 - ~Recognizer, 29
 - backgroundRecognizeFrame, 29
 - isRunning, 29
 - m_matcher, 30
 - m_running, 30
 - m_thread, 30
 - recognizeFrame, 29
 - Recognizer, 29
 - sendMessage, 30
 - setDatabase, 30
 - setRunning, 30
 - TAG, 30
- IStuff::Tracker, 31
 - ~Tracker, 32
 - backgroundTrackFrame, 32
 - calcFeatures, 32, 33
 - IMG_RESIZE, 34
 - isRunning, 33
 - LK_WINDOW, 34
 - m_detector, 35
 - m_features, 35
 - m_frame, 35
 - m_matcher, 35
 - m_object, 35
 - m_object_mutex, 35
 - m_running, 35
 - m_saved_features, 35
 - m_thread, 35
 - sendMessage, 33
 - setRunning, 34
 - TAG, 35
 - trackFrame, 34
 - Tracker, 32
 - updateObject, 34
- isRunning
 - IStuff::Recognizer, 29
 - IStuff::Tracker, 33
- keypointDB
 - IStuff::Database, 15
- LK_WINDOW
 - IStuff::Tracker, 34
- Label
 - IStuff::Label, 21
- labelDB
 - IStuff::Database, 15
- labels
 - IStuff::Object, 27
- load
 - boost::serialization, 11
 - IStuff::Database, 14
- m_detector
 - IStuff::Tracker, 35
- m_features
 - IStuff::Tracker, 35
- m_frame
 - IStuff::Tracker, 35
- m_matcher
 - IStuff::Recognizer, 30
 - IStuff::Tracker, 35
- m_object

- ISuff::Tracker, 35
- m_object_mutex
 - ISuff::Tracker, 35
- m_running
 - ISuff::Recognizer, 30
 - ISuff::Tracker, 35
- m_saved_features
 - ISuff::Tracker, 35
- m_thread
 - ISuff::Recognizer, 30
 - ISuff::Tracker, 35
- MATCH_THRESHOLD
 - ISuff::Database, 15
- MIN_INLIER_RATIO
 - ISuff::Database, 15
- MSG_TRACKING_END
 - ISuff::Manager, 25
- MSG_TRACKING_START
 - ISuff::Manager, 25
- main
 - main.cpp, 46
 - main.h, 47
- main.cpp
 - main, 46
 - printHelp, 46
- main.h
 - debug, 47
 - hl_debug, 47
 - main, 47
 - printHelp, 47
- Manager
 - ISuff::Manager, 23
- manager.h
 - debug, 41
 - hl_debug, 41
- match
 - ISuff::Database, 14
- matcher
 - ISuff::Database, 15
- NNDR_RATIO
 - ISuff::Database, 16
- name
 - ISuff::Label, 21
- Object
 - ISuff::Object, 26
- object.h
 - debug, 42
- object_update
 - ISuff::Manager, 25
- operator==
 - ISuff::Label, 21
- paint
 - ISuff::Object, 27
- paintObject
 - ISuff::Manager, 23
- position
 - ISuff::Label, 21
- printHelp
 - main.cpp, 46
 - main.h, 47
- queue_mutex
 - ISuff::FakableQueue, 20
- RECOGNITION_PERIOD
 - ISuff::Manager, 25
- real_queue
 - ISuff::FakableQueue, 20
- recognizeFrame
 - ISuff::Recognizer, 29
- Recognizer
 - ISuff::Recognizer, 29
- recognizer
 - ISuff::Manager, 25
- recognizer.h
 - debug, 43
- save
 - boost::serialization, 11
 - ISuff::Database, 15
- saved_queue
 - ISuff::FakableQueue, 20
- sendMessage
 - ISuff::Manager, 24
 - ISuff::Recognizer, 30
 - ISuff::Tracker, 33
- setDatabase
 - ISuff::Manager, 24
 - ISuff::Recognizer, 30
- setObject
 - ISuff::Manager, 24
- setRunning
 - ISuff::Recognizer, 30
 - ISuff::Tracker, 34
- src/ISuff/database.cpp, 37
- src/ISuff/database.h, 37
- src/ISuff/fakable_queue.cpp, 38
- src/ISuff/fakable_queue.h, 39
- src/ISuff/manager.cpp, 40
- src/ISuff/manager.h, 40
- src/ISuff/object.cpp, 41
- src/ISuff/object.h, 41
- src/ISuff/recognizer.cpp, 42
- src/ISuff/recognizer.h, 42
- src/ISuff/serialize_opencv.h, 43
- src/ISuff/tracker.cpp, 44
- src/ISuff/tracker.h, 44
- src/main.cpp, 45
- src/main.h, 46
- start
 - ISuff::FakableQueue, 20
- TAG
 - ISuff::FakableQueue, 20
 - ISuff::Manager, 25

- ISuff::Object, [27](#)
 - ISuff::Recognizer, [30](#)
 - ISuff::Tracker, [35](#)
- trackFrame
 - ISuff::Tracker, [34](#)
- Tracker
 - ISuff::Tracker, [32](#)
- tracker
 - ISuff::Manager, [25](#)
- tracker.h
 - debug, [45](#)
- updateObject
 - ISuff::Tracker, [34](#)
- what
 - ISuff::DBCException, [16](#)
 - ISuff::DBLoadingException, [17](#)
 - ISuff::DBSavingException, [17](#)