

## Algoritmo di tracking per una camera monoculare

Generated by Doxygen 1.8.6

Fri Apr 15 2016 17:21:53



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Camera Class Reference . . . . .	5
3.1.1	Detailed Description . . . . .	7
3.1.2	Constructor & Destructor Documentation . . . . .	7
3.1.2.1	Camera . . . . .	7
3.1.2.2	~Camera . . . . .	7
3.1.3	Member Function Documentation . . . . .	7
3.1.3.1	CallBackFunc . . . . .	7
3.1.3.2	ControllCamera . . . . .	7
3.1.3.3	ConvertPointFromWordToCam . . . . .	7
3.1.3.4	DetectWithSift . . . . .	7
3.1.3.5	FillCamMatrixPose . . . . .	7
3.1.3.6	FindAMinDistanceButton . . . . .	7
3.1.3.7	FindBottonPos3D . . . . .	7
3.1.3.8	FindContours . . . . .	8
3.1.3.9	ImageConverter . . . . .	8
3.1.3.10	InfoKf3d . . . . .	8
3.1.3.11	ProjectPointAndFindPosBot3d . . . . .	8
3.1.3.12	RobotMove . . . . .	8
3.1.3.13	ShapeDetect . . . . .	8
3.1.3.14	SOTreCamera . . . . .	8
3.1.3.15	Triangulation . . . . .	8
3.1.4	Member Data Documentation . . . . .	9
3.1.4.1	arrived_cam . . . . .	9
3.1.4.2	BottonCHosen . . . . .	9
3.1.4.3	cam_cx . . . . .	9

3.1.4.4	cam_cy	9
3.1.4.5	cam_d0	9
3.1.4.6	cam_d1	9
3.1.4.7	cam_d2	9
3.1.4.8	cam_d3	9
3.1.4.9	cam_d4	9
3.1.4.10	cam_fx	9
3.1.4.11	cam_fy	9
3.1.4.12	Cam_par_distortion	9
3.1.4.13	Camera2_S03	9
3.1.4.14	Camera_Matrix	9
3.1.4.15	camera_topic	9
3.1.4.16	count_n_passi	9
3.1.4.17	first_Step	9
3.1.4.18	FirstCalibration	9
3.1.4.19	frame1	9
3.1.4.20	frame_so3_ptam	9
3.1.4.21	frame_w_c	9
3.1.4.22	it	9
3.1.4.23	KeypointIm1Match	9
3.1.4.24	KeypointIm2	9
3.1.4.25	move_camera_end	9
3.1.4.26	Move_robot	9
3.1.4.27	movewebcamrobot	10
3.1.4.28	nh	10
3.1.4.29	pos_object	10
3.1.4.30	press_button	10
3.1.4.31	ptam_kf3d	10
3.1.4.32	ptam_sub	10
3.1.4.33	Ptamkf3d	10
3.1.4.34	SaveFirst	10
3.1.4.35	scala	10
3.1.4.36	scene	10
3.1.4.37	So3_prev_ptam	10
3.1.4.38	sub	10
3.1.4.39	sub_ptam_2	10
3.2	Camera::Obj Struct Reference	10
3.2.1	Member Data Documentation	10
3.2.1.1	Bot_C	10
3.2.1.2	Botton_2frame	10

3.2.1.3	Center_ . . . . .	11
3.2.1.4	descr_ . . . . .	11
3.2.1.5	figure_ . . . . .	11
3.2.1.6	keyp_ . . . . .	11
3.2.1.7	Pos3d_ . . . . .	11
3.3	Settings Class Reference . . . . .	11
3.3.1	Member Enumeration Documentation . . . . .	12
3.3.1.1	InputType . . . . .	12
3.3.1.2	Pattern . . . . .	12
3.3.2	Constructor & Destructor Documentation . . . . .	12
3.3.2.1	Settings . . . . .	12
3.3.3	Member Function Documentation . . . . .	12
3.3.3.1	interpret . . . . .	12
3.3.3.2	nextImage . . . . .	12
3.3.3.3	read . . . . .	12
3.3.3.4	readStringList . . . . .	12
3.3.3.5	write . . . . .	12
3.3.4	Member Data Documentation . . . . .	12
3.3.4.1	aspectRatio . . . . .	12
3.3.4.2	atImageList . . . . .	12
3.3.4.3	boardSize . . . . .	13
3.3.4.4	bwriteExtrinsics . . . . .	13
3.3.4.5	bwritePoints . . . . .	13
3.3.4.6	calibFixPrincipalPoint . . . . .	13
3.3.4.7	calibrationPattern . . . . .	13
3.3.4.8	calibZeroTangentDist . . . . .	13
3.3.4.9	cameraID . . . . .	13
3.3.4.10	delay . . . . .	13
3.3.4.11	flag . . . . .	13
3.3.4.12	flipVertical . . . . .	13
3.3.4.13	goodInput . . . . .	13
3.3.4.14	imageList . . . . .	13
3.3.4.15	input . . . . .	13
3.3.4.16	inputCapture . . . . .	13
3.3.4.17	inputType . . . . .	13
3.3.4.18	nrFrames . . . . .	13
3.3.4.19	outputFileName . . . . .	13
3.3.4.20	patternToUse . . . . .	13
3.3.4.21	showUndistorsed . . . . .	13
3.3.4.22	squareSize . . . . .	13

<b>4</b>	<b>File Documentation</b>	<b>15</b>
4.1	/home/daniela/code/src/eye_in_hand/include/calibration.h File Reference	15
4.1.1	Macro Definition Documentation	15
4.1.1.1	_CRT_SECURE_NO_WARNINGS	15
4.1.2	Enumeration Type Documentation	15
4.1.2.1	anonymous enum	16
4.1.3	Function Documentation	16
4.1.3.1	runCalibrationAndSave	16
4.2	/home/daniela/code/src/eye_in_hand/include/Camera_code.h File Reference	16
4.2.1	Function Documentation	17
4.2.1.1	EstimatePlane	17
4.2.1.2	FindACenter	17
4.2.1.3	FindMaxValue	17
4.2.1.4	FromCvPointToEigen	18
4.2.1.5	FromEigenVectorToCvPOint	19
4.2.1.6	FromMatToEigen	19
4.2.1.7	Media	19
4.2.1.8	ScalaReturn	19
4.2.1.9	setLabel	19
4.3	/home/daniela/code/src/eye_in_hand/include/converter_file.hpp File Reference	19
4.3.1	Function Documentation	20
4.3.1.1	FromCvPointToEigen	20
4.3.1.2	FromEigenVectorToCvPOint	21
4.3.1.3	FromMatToEigen	21
4.4	/home/daniela/code/src/eye_in_hand/include/geometry_function.hpp File Reference	21
4.4.1	Function Documentation	21
4.4.1.1	FindACenter	21
4.4.1.2	FindMaxValue	21
4.4.1.3	setLabel	22
4.5	/home/daniela/code/src/eye_in_hand/include/plane_estimate.hpp File Reference	22
4.5.1	Function Documentation	22
4.5.1.1	EstimatePlane	22
4.6	/home/daniela/code/src/eye_in_hand/src/calibration.cpp File Reference	22
4.6.1	Function Documentation	22
4.6.1.1	main	22
4.6.1.2	runCalibrationAndSave	23
4.7	/home/daniela/code/src/eye_in_hand/src/Camera_code.cpp File Reference	23
4.7.1	Function Documentation	23
4.7.1.1	main	23
4.7.1.2	rng	23

---

4.7.1.3	ScalaReturn . . . . .	23
4.8	/home/daniela/code/src/eye_in_hand/src/main_image_.cpp File Reference . . . . .	23
4.8.1	Function Documentation . . . . .	24
4.8.1.1	main . . . . .	24
<b>Index</b>		<b>25</b>





# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">Camera</a>	5
<a href="#">Camera::Obj</a>	10
<a href="#">Settings</a>	11



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

/home/daniela/code/src/eye_in_hand/include/calibration.h . . . . .	15
/home/daniela/code/src/eye_in_hand/include/Camera_code.h . . . . .	16
/home/daniela/code/src/eye_in_hand/include/converter_file.hpp . . . . .	19
/home/daniela/code/src/eye_in_hand/include/geometry_function.hpp . . . . .	21
/home/daniela/code/src/eye_in_hand/include/plane_estimate.hpp . . . . .	22
/home/daniela/code/src/eye_in_hand/src/calibration.cpp . . . . .	22
/home/daniela/code/src/eye_in_hand/src/Camera_code.cpp . . . . .	23
/home/daniela/code/src/eye_in_hand/src/main_image_.cpp . . . . .	23



## Chapter 3

# Class Documentation

### 3.1 Camera Class Reference

```
#include <Camera_code.h>
```

#### Classes

- struct [Obj](#)

#### Public Member Functions

- [Camera](#) ()  
: constructor
- [~Camera](#) ()  
: destructor
- void [ControllCamera](#) ()  
: Function that handle the first iteration.
- void [DetectWithSift](#) ()  
: Function that calculates the keypoint for the second image, and match it with the first image
- void [Triangulation](#) ()  
: Function that triangulate the point

#### Public Attributes

- ros::NodeHandle [nh](#)
- cv::Mat [Camera\\_Matrix](#)
- cv::Mat [Cam\\_par\\_distortion](#)
- cv::Mat [Camera2\\_S03](#)
- KDL::Frame [Move\\_robot](#)
- cv::Mat [scene](#)
- int [arrived\\_cam](#) = 0
- ros::Subscriber [movewebcamrobot](#)
- ros::Subscriber [ptam\\_kf3d](#)
- image\_transport::ImageTransport [it\\_](#)
- image\_transport::Subscriber [sub](#)
- ros::Subscriber [ptam\\_sub](#)
- struct [Camera::Obj](#) [BottonCHosen](#)

- `std::vector< cv::Point2f >` [KeypointIm2](#)
  - `std::vector< cv::Point2f >` [KeyPointIm1Match](#)
  - `KDL::Frame` [frame\\_so3\\_ptam](#)
  - `std::string` [camera\\_topic\\_](#)
  - `bool` [move\\_camera\\_end](#)
  - `int` [FirstCalibration](#)
  - `cv::Mat` [frame1\\_](#)
  - `double` [scala](#)
  - `KDL::Frame` [So3\\_prev\\_ptam](#)
  - `bool` [SaveFirst](#)
  - `bool` [sub\\_ptam\\_2](#)
  - `float` [cam\\_fx](#)
  - `float` [cam\\_d0](#)
  - `float` [cam\\_d1](#)
  - `float` [cam\\_d2](#)
  - `float` [cam\\_d3](#)
  - `float` [cam\\_d4](#)
  - `float` [cam\\_fy](#)
  - `float` [cam\\_cx](#)
  - `float` [cam\\_cy](#)
  - `pcl::PointCloud< pcl::PointXYZ >` [Ptamkf3d](#)
  - `KDL::Frame` [frame\\_w\\_c](#)
  - `int` [count\\_n\\_passi](#)
- camere in word*

### Static Public Attributes

- `static cv::Point` [pos\\_object](#)
- `static int` [press\\_button](#) = 0
- `static int` [first\\_Step](#) = 1

### Private Member Functions

- `void` [ShapeDetect](#) ()  
: *Function that find the botton desired*
- `std::pair< int, bool >` [FindAMinDistanceButton](#) (`std::vector< cv::Point >` &baricentro, `cv::Point` &point\_)
- `void` [ImageConverter](#) (`const sensor_msgs::Image::ConstPtr` &msg)  
: *Callback that convert sensor image with cv mat*
- `std::pair< std::vector< cv::Point >, std::vector< std::vector< cv::Point > > >` [FindContours](#) (`cv::Mat` bw, `cv::Mat` camera)  
: *Approximate contour with accuracy proportional to the contour perimeter*
- `void` [SOtreCamera](#) (`const geometry_msgs::PoseWithCovarianceStamped::ConstPtr` msg)  
: *Pose word into camera*
- `void` [RobotMove](#) (`const geometry_msgs::Pose` msg)  
: *Callback that save the real movement of robot*
- `void` [InfoKf3d](#) (`const sensor_msgs::PointCloud2::ConstPtr` &msg)  
: *Callback that convert PointCloud2 to poincloudXYX*
- `void` [ProjectPointAndFindPosBot3d](#) (`std::vector< cv::Point3d >` vect3d)  
: *Reproject the points 3d to 2d and find the 3d botton pose*
- `void` [FillCamMatrixPose](#) (`KDL::Frame` frame)
- `std::vector< cv::Point3d >` [ConvertPointFromWordToCam](#) ()  
: *Convert each point in camera frame*
- `void` [FindBottonPos3D](#) (`Eigen::Vector4f` plane\_param)  
: *Find the 3d botton pose*

## Static Private Member Functions

- static void [CallBackFunc](#) (int event, int x, int y, int flags, void \*userdata)  
: Function that save the 2d position of press botton

### 3.1.1 Detailed Description

test

### 3.1.2 Constructor & Destructor Documentation

#### 3.1.2.1 Camera::Camera ( )

: constructor

#### 3.1.2.2 Camera::~~Camera ( ) [inline]

: destructor

### 3.1.3 Member Function Documentation

#### 3.1.3.1 void Camera::CallBackFunc ( int event, int x, int y, int flags, void \* userdata ) [static], [private]

: Function that save the 2d position of press botton

#### 3.1.3.2 void Camera::ControllCamera ( )

: Function that handle the first iteration.

#### 3.1.3.3 std::vector< cv::Point3d > Camera::ConvertPointFromWordToCam ( ) [private]

: Convert each point in camera frame

Returns

: 3d point in camera frame

Scaling each point with the scale factor.

#### 3.1.3.4 void Camera::DetectWithSift ( )

: Function that calculates the keypoint for the second image, and match it with the first image

#### 3.1.3.5 void Camera::FillCamMatrixPose ( KDL::Frame frame ) [private]

#### 3.1.3.6 std::pair< int, bool > Camera::FindAMinDistanceButton ( std::vector< cv::Point > & baricentro, cv::Point & point\_ ) [private]

#### 3.1.3.7 void Camera::FindBottonPos3D ( Eigen::Vector4f plane\_param ) [private]

: Find the 3d botton pose

**Parameters**

in	plane param
----	-------------

Project the 2d botton position into 3d plane

**3.1.3.8** `std::pair< std::vector< cv::Point >, std::vector< std::vector< cv::Point > > > Camera::FindContours ( cv::Mat bw, cv::Mat camera )` [private]

: Approximate contour with accuracy proportional to the contour perimeter

**Returns**

: Center of each shape and his contours

**3.1.3.9** `void Camera::ImageConverter ( const sensor_msgs::Image::ConstPtr & msg )` [private]

: Callback that convert sensor image with cv mat

**3.1.3.10** `void Camera::InfoKf3d ( const sensor_msgs::PointCloud2::ConstPtr & msg )` [private]

: Callback that convert PointCloud2 to poincloudXYX

This pose is relative to word frame.

**3.1.3.11** `void Camera::ProjectPointAndFindPosBot3d ( std::vector< cv::Point3d > vect3d )` [private]

: Reproject the points 3d to 2d and find the 3d botton pose

**Parameters**

pose of each feature
----------------------

To find the 3d botton pose, we fit a plane with lms method and project the 2d botton position into this plane

**3.1.3.12** `void Camera::RobotMove ( const geometry_msgs::Pose msg )` [private]

: Callback that save the real movement of robot

This pose is relative to frame early

**3.1.3.13** `void Camera::ShapeDetect ( )` [private]

: Function that find the botton desired

**3.1.3.14** `void Camera::SOTreCamera ( const geometry_msgs::PoseWithCovarianceStamped::ConstPtr msg )` [private]

: Pose word into camera

**3.1.3.15** `void Camera::Triangulation ( )`

: Function that triangulate the point



### 3.1.4 Member Data Documentation

3.1.4.1 `int Camera::arrived_cam = 0`

3.1.4.2 `struct Camera::Obj Camera::BottonCHosen`

3.1.4.3 `float Camera::cam_cx`

3.1.4.4 `float Camera::cam_cy`

3.1.4.5 `float Camera::cam_d0`

3.1.4.6 `float Camera::cam_d1`

3.1.4.7 `float Camera::cam_d2`

3.1.4.8 `float Camera::cam_d3`

3.1.4.9 `float Camera::cam_d4`

3.1.4.10 `float Camera::cam_fx`

3.1.4.11 `float Camera::cam_fy`

3.1.4.12 `cv::Mat Camera::Cam_par_distortion`

3.1.4.13 `cv::Mat Camera::Camera2_S03`

3.1.4.14 `cv::Mat Camera::Camera_Matrix`

3.1.4.15 `std::string Camera::camera_topic_`

3.1.4.16 `int Camera::count_n_passi`

`camere in word`

3.1.4.17 `int Camera::first_Step = 1` `[static]`

3.1.4.18 `int Camera::FirstCalibration`

3.1.4.19 `cv::Mat Camera::frame1_`

3.1.4.20 `KDL::Frame Camera::frame_so3_ptam`

3.1.4.21 `KDL::Frame Camera::frame_w_c`

3.1.4.22 `image_transport::ImageTransport Camera::it_`

3.1.4.23 `std::vector<cv::Point2f> Camera::KeypointIm1Match`

3.1.4.24 `std::vector<cv::Point2f> Camera::KeypointIm2`

3.1.4.25 `bool Camera::move_camera_end`

3.1.4.26 `KDL::Frame Camera::Move_robot`

- 3.1.4.27 `ros::Subscriber Camera::movewebcamrobot`
- 3.1.4.28 `ros::NodeHandle Camera::nh`
- 3.1.4.29 `cv::Point Camera::pos_object` `[static]`
- 3.1.4.30 `int Camera::press_buttom = 0` `[static]`
- 3.1.4.31 `ros::Subscriber Camera::ptam_kf3d`
- 3.1.4.32 `ros::Subscriber Camera::ptam_sub`
- 3.1.4.33 `pcl::PointCloud<pcl::PointXYZ> Camera::Ptamkf3d`
- 3.1.4.34 `bool Camera::SaveFirst`
- 3.1.4.35 `double Camera::scala`
- 3.1.4.36 `cv::Mat Camera::scene`
- 3.1.4.37 `KDL::Frame Camera::So3_prev_ptam`
- 3.1.4.38 `image_transport::Subscriber Camera::sub`
- 3.1.4.39 `bool Camera::sub_ptam_2`

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

- [/home/daniela/code/src/eye\\_in\\_hand/include/Camera\\_code.h](#)
- [/home/daniela/code/src/eye\\_in\\_hand/include/geometry\\_function.hpp](#)
- [/home/daniela/code/src/eye\\_in\\_hand/include/plane\\_estimate.hpp](#)
- [/home/daniela/code/src/eye\\_in\\_hand/src/Camera\\_code.cpp](#)

## 3.2 Camera::Obj Struct Reference

```
#include <Camera_code.h>
```

### Public Attributes

- `std::vector< cv::Point >` [Bot\\_C](#)
- `cv::Point` [Center\\_](#)
- `std::vector< cv::KeyPoint >` [keyp\\_](#)
- `cv::Mat` [descr\\_](#)
- `cv::Mat` [figure\\_](#)
- `cv::Point3d` [Pos3d\\_](#)
- `cv::Point` [Botton\\_2frame](#)

### 3.2.1 Member Data Documentation

- 3.2.1.1 `std::vector<cv::Point> Camera::Obj::Bot_C`
- 3.2.1.2 `cv::Point Camera::Obj::Botton_2frame`

3.2.1.3 `cv::Point Camera::Obj::Center_`

3.2.1.4 `cv::Mat Camera::Obj::descr_`

3.2.1.5 `cv::Mat Camera::Obj::figure_`

3.2.1.6 `std::vector<cv::KeyPoint> Camera::Obj::keyp_`

3.2.1.7 `cv::Point3d Camera::Obj::Pos3d_`

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

- `/home/daniela/code/src/eye_in_hand/include/Camera_code.h`

## 3.3 Settings Class Reference

```
#include <calibration.h>
```

### Public Types

- enum `Pattern` { `NOT_EXISTING`, `CHESSBOARD`, `CIRCLES_GRID`, `ASYMMETRIC_CIRCLES_GRID` }
- enum `InputType` { `INVALID`, `CAMERA`, `VIDEO_FILE`, `IMAGE_LIST` }

### Public Member Functions

- `Settings` ()
- void `write` (FileStorage &fs) const
- void `read` (const FileNode &node)
- void `interpret` ()
- Mat `nextImage` ()

### Static Public Member Functions

- static bool `readStringList` (const string &filename, vector< string > &l)

### Public Attributes

- Size `boardSize`
- `Pattern` `calibrationPattern`
- float `squareSize`
- int `nrFrames`
- float `aspectRatio`
- int `delay`
- bool `bwritePoints`
- bool `bwriteExtrinsics`
- bool `calibZeroTangentDist`
- bool `calibFixPrincipalPoint`
- bool `flipVertical`
- string `outputFileName`
- bool `showUndistorted`
- string `input`
- int `cameraID`

- vector< string > [imageList](#)
- int [atImageList](#)
- VideoCapture [inputCapture](#)
- [InputType](#) [inputType](#)
- bool [goodInput](#)
- int [flag](#)

### Private Attributes

- string [patternToUse](#)

## 3.3.1 Member Enumeration Documentation

### 3.3.1.1 enum Settings::InputType

Enumerator

***INVALID***  
***CAMERA***  
***VIDEO\_FILE***  
***IMAGE\_LIST***

### 3.3.1.2 enum Settings::Pattern

Enumerator

***NOT\_EXISTING***  
***CHESSBOARD***  
***CIRCLES\_GRID***  
***ASYMMETRIC\_CIRCLES\_GRID***

## 3.3.2 Constructor & Destructor Documentation

### 3.3.2.1 Settings::Settings ( ) [inline]

## 3.3.3 Member Function Documentation

### 3.3.3.1 void Settings::Interprate ( ) [inline]

### 3.3.3.2 Mat Settings::nextImage ( ) [inline]

### 3.3.3.3 void Settings::read ( const FileNode & *node* ) [inline]

### 3.3.3.4 static bool Settings::readStringList ( const string & *filename*, vector< string > & *I* ) [inline],[static]

### 3.3.3.5 void Settings::write ( FileStorage & *fs* ) const [inline]

## 3.3.4 Member Data Documentation

### 3.3.4.1 float Settings::aspectRatio

### 3.3.4.2 int Settings::atImageList

- 3.3.4.3    `Size` `Settings::boardSize`
- 3.3.4.4    `bool` `Settings::bwriteExtrinsics`
- 3.3.4.5    `bool` `Settings::bwritePoints`
- 3.3.4.6    `bool` `Settings::calibFixPrincipalPoint`
- 3.3.4.7    `Pattern` `Settings::calibrationPattern`
- 3.3.4.8    `bool` `Settings::calibZeroTangentDist`
- 3.3.4.9    `int` `Settings::cameraID`
- 3.3.4.10   `int` `Settings::delay`
- 3.3.4.11   `int` `Settings::flag`
- 3.3.4.12   `bool` `Settings::flipVertical`
- 3.3.4.13   `bool` `Settings::goodInput`
- 3.3.4.14   `vector<string>` `Settings::imageList`
- 3.3.4.15   `string` `Settings::input`
- 3.3.4.16   `VideoCapture` `Settings::inputCapture`
- 3.3.4.17   `InputType` `Settings::inputType`
- 3.3.4.18   `int` `Settings::nrFrames`
- 3.3.4.19   `string` `Settings::outputFileName`
- 3.3.4.20   `string` `Settings::patternToUse`    `[private]`
- 3.3.4.21   `bool` `Settings::showUndistorted`
- 3.3.4.22   `float` `Settings::squareSize`

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

- `/home/daniela/code/src/eye_in_hand/include/calibration.h`



## Chapter 4

# File Documentation

### 4.1 /home/daniela/code/src/eye\_in\_hand/include/calibration.h File Reference

```
#include <iostream>
#include <sstream>
#include <time.h>
#include <stdio.h>
#include <opencv2/core/core.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/calib3d/calib3d.hpp>
#include <opencv2/highgui/highgui.hpp>
```

#### Classes

- class [Settings](#)

#### Macros

- `#define \_CRT\_SECURE\_NO\_WARNINGS`

#### Enumerations

- enum { [DETECTION](#) = 0, [CAPTURING](#) = 1, [CALIBRATED](#) = 2 }

#### Functions

- bool [runCalibrationAndSave](#) ([Settings](#) &s, Size imageSize, Mat &cameraMatrix, Mat &distCoeffs, vector< vector< Point2f > > imagePoints)

#### 4.1.1 Macro Definition Documentation

##### 4.1.1.1 `#define \_CRT\_SECURE\_NO\_WARNINGS`

#### 4.1.2 Enumeration Type Documentation

#### 4.1.2.1 anonymous enum

Enumerator

**DETECTION**

**CAPTURING**

**CALIBRATED**

### 4.1.3 Function Documentation

4.1.3.1 **bool runCalibrationAndSave ( Settings & s, Size *imageSize*, Mat & *cameraMatrix*, Mat & *distCoeffs*, vector< vector< Point2f > > *imagePoints* )**

## 4.2 /home/daniela/code/src/eye\_in\_hand/include/Camera\_code.h File Reference

```
#include <utility>
#include <list>
#include <string>
#include <eigen3/Eigen/Dense>
#include <kdl/frames.hpp>
#include <kdl_parser/kdl_parser.hpp>
#include <kdl/kdl.hpp>
#include <kdl/frames_io.hpp>
#include <ptam_com/KeyFrame_msg.h>
#include <ros/ros.h>
#include <image_transport/image_transport.h>
#include <cv_bridge/cv_bridge.h>
#include <sensor_msgs/image_encodings.h>
#include <opencv2/imgproc/imgproc.hpp>
#include <geometry_msgs/Pose.h>
#include <opencv2/highgui/highgui.hpp>
#include <std_msgs/Bool.h>
#include <sensor_msgs/PointCloud2.h>
#include <stdio.h>
#include <iostream>
#include <fstream>
#include "opencv2/core/core.hpp"
#include "opencv2/features2d/features2d.hpp"
#include "opencv2/nonfree/features2d.hpp"
#include "opencv2/nonfree/nonfree.hpp"
#include <tf/transform_broadcaster.h>
#include <geometry_msgs/PoseWithCovarianceStamped.h>
#include "opencv2/calib3d/calib3d.hpp"
#include <tf_conversions/tf_kdl.h>
#include <pcl/io/pcd_io.h>
#include <pcl_conversions/pcl_conversions.h>
#include <std_msgs/Float32MultiArray.h>
#include <visualization_msgs/MarkerArray.h>
```

### Classes

- class [Camera](#)
- struct [Camera::Obj](#)



## Functions

- Eigen::Vector4f [EstimatePlane](#) (std::vector< cv::Point3d > Point\_Near)  
: Plane estimation
- void [setLabel](#) (cv::Mat &im, const std::string label, std::vector< cv::Point > &contour)
- cv::Point [FindACenter](#) (std::vector< cv::Point > &geometry)  
: Find a centroid of shape
- std::pair< int, int > [FindMaxValue](#) (cv::Mat &matrix, cv::Point &point)  
: function that calculates the max value of width and height for the roi
- double [Media](#) (cv::Mat triangulatedPoints3D, double MaxLenght, int col)
- void [FromCvPointToEigen](#) (cv::Point3d point\_, Eigen::VectorXd &vect)  
: Convert cv point into eigen vector
- void [FromMatToEigen](#) (cv::Mat Mat\_, Eigen::MatrixXd &Eigen)  
: Convert cv mat into eigen matrix
- void [FromEigenVectorToCvPOint](#) (Eigen::VectorXd Eigen, cv::Point3d &mat)  
: Convert eigen vector to cv point
- double [ScalaReturn](#) (double ptam, double ptam\_prev, double robot)  
: Calculates the scale factor

### 4.2.1 Function Documentation

#### 4.2.1.1 Eigen::Vector4f EstimatePlane ( std::vector< cv::Point3d > Point\_Near )

: Plane estimation

Parameters

in		3d position of each point near the botton
----	--	---

Estimate the plane with lms method

#### 4.2.1.2 cv::Point FindACenter ( std::vector< cv::Point > & geometry )

: Find a centroid of shape

Parameters

in		shape contours
----	--	----------------

Returns

: center position

#### 4.2.1.3 std::pair<int,int> FindMaxValue ( cv::Mat & matrix, cv::Point & point )

: function that calculates the max value of width and height for the roi

Parameters

in		mat of scene
in		point of interest

Returns

: range to cut the image to make a roi

4.2.1.4 void FromCvPointToEigen ( cv::Point3d *point\_*, Eigen::VectorXd & *vect* )

: Convert cv point into eigen vector

## Parameters

in		cv::Point3d
out		Eigen::VectorXd homogeneous

4.2.1.5 void FromEigenVectorToCvP0int ( Eigen::VectorXd *Eigen*, cv::Point3d & *mat* )

: Convert eigen vector to cv point

## Parameters

in		Eigen::VectorXd
out		cv::Point3d

4.2.1.6 void FromMatToEigen ( cv::Mat *Mat\_*, Eigen::MatrixXd & *Eigen* )

: Convert cv mat into eigen matrix

## Parameters

in		cv::Mat
out		Eigen::MatrixXd homogeneous

4.2.1.7 double Media ( cv::Mat *triangulatedPoints3D*, double *MaxLenght*, int *col* )

4.2.1.8 double ScalaReturn ( double *ptam*, double *ptam\_prev*, double *robot* )

: Calculates the scale factor

## Parameters

in		actual position
in		previously position
in		robot movements

## Returns

scale factor

4.2.1.9 void setLabel ( cv::Mat & *im*, const std::string *label*, std::vector< cv::Point > & *contour* )

## 4.3 /home/daniela/code/src/eye\_in\_hand/include/converter\_file.hpp File Reference

## Functions

- void [FromMatToEigen](#) (cv::Mat *Mat\_*, Eigen::MatrixXd & *Eigen1*)  
: Convert cv mat into eigen matrix
- void [FromCvPointToEigen](#) (cv::Point3d *point\_*, Eigen::VectorXd & *vect*)  
: Convert cv point into eigen vector
- void [FromEigenVectorToCvP0int](#) (Eigen::VectorXd *Eigen*, cv::Point3d & *mat*)  
: Convert eigen vector to cv point

### 4.3.1 Function Documentation

4.3.1.1 void FromCvPointToEigen ( cv::Point3d *point\_*, Eigen::VectorXd & *vect* )

: Convert cv point into eigen vector

## Parameters

in		cv::Point3d
out		Eigen::VectorXd homogeneous

4.3.1.2 void FromEigenVectorToCvP0int ( Eigen::VectorXd *Eigen*, cv::Point3d & *mat* )

: Convert eigen vector to cv point

## Parameters

in		Eigen::VectorXd
out		cv::Point3d

4.3.1.3 void FromMatToEigen ( cv::Mat *Mat\_*, Eigen::MatrixXd & *Eigen* )

: Convert cv mat into eigen matrix

## Parameters

in		cv::Mat
out		Eigen::MatrixXd homogeneous

## 4.4 /home/daniela/code/src/eye\_in\_hand/include/geometry\_function.hpp File Reference

## Functions

- std::pair< int, int > [FindMaxValue](#) (cv::Mat &matrix, cv::Point &point)  
: function that calculates the max value of width and height for the roi
- cv::Point [FindACenter](#) (std::vector< cv::Point > &geometry)  
: Find a centroid of shape
- void [setLabel](#) (cv::Mat &im, const std::string label, std::vector< cv::Point > &contour)

## 4.4.1 Function Documentation

4.4.1.1 cv::Point FindACenter ( std::vector< cv::Point > & *geometry* )

: Find a centroid of shape

## Parameters

in		shape contours
----	--	----------------

## Returns

: center position

4.4.1.2 std::pair<int,int> FindMaxValue ( cv::Mat & *matrix*, cv::Point & *point* )

: function that calculates the max value of width and height for the roi

**Parameters**

in		mat of scene
in		point of interest

**Returns**

: range to cut the image to make a roi

4.4.1.3 void setLabel ( cv::Mat & *im*, const std::string *label*, std::vector< cv::Point > & *contour* )

**4.5 /home/daniela/code/src/eye\_in\_hand/include/plane\_estimate.hpp File Reference**

```
#include <pcl/search/kdtree.h>
#include <pcl/point_cloud.h>
#include <pcl/io/pcd_io.h>
#include <pcl/point_types.h>
#include <pcl/features/normal_3d.h>
```

**Functions**

- Eigen::Vector4f [EstimatePlane](#) (std::vector< cv::Point3d > *Point\_Near*)  
: *Plane estimation*

**4.5.1 Function Documentation**

4.5.1.1 Eigen::Vector4f EstimatePlane ( std::vector< cv::Point3d > *Point\_Near* )

: Plane estimation

**Parameters**

in		3d position of each point near the botton
----	--	---

Estimate the plane with lms method

**4.6 /home/daniela/code/src/eye\_in\_hand/src/calibration.cpp File Reference**

```
#include <calibration.h>
```

**Functions**

- int [main](#) (int argc, char \*argv[])
- bool [runCalibrationAndSave](#) ([Settings](#) &s, Size imageSize, Mat &cameraMatrix, Mat &distCoeffs, vector< vector< Point2f > > imagePoints)

**4.6.1 Function Documentation**

4.6.1.1 int main ( int *argc*, char \* *argv*[] )

4.6.1.2 `bool runCalibrationAndSave ( Settings & s, Size imageSize, Mat & cameraMatrix, Mat & distCoeffs, vector< vector< Point2f > > imagePoints )`

## 4.7 /home/daniela/code/src/eye\_in\_hand/src/Camera\_code.cpp File Reference

```
#include <Camera_code.h>
#include <geometry_function.hpp>
#include <converter_file.hpp>
#include <plane_estimate.hpp>
```

### Functions

- `cv::RNG rng (12345)`
- `int main (int argc, char **argv)`
- `double ScalaReturn (double ptam, double ptam_prev, double robot)`  
: Calculates the scale factor

#### 4.7.1 Function Documentation

4.7.1.1 `int main ( int argc, char ** argv )`

4.7.1.2 `cv::RNG rng ( 12345 )`

4.7.1.3 `double ScalaReturn ( double ptam, double ptam_prev, double robot )`

: Calculates the scale factor

#### Parameters

in		actual position
in		previously position
in		robot movements

#### Returns

scale factor

## 4.8 /home/daniela/code/src/eye\_in\_hand/src/main\_image\_.cpp File Reference

```
#include <ros/ros.h>
#include <image_transport/image_transport.h>
#include <cv_bridge/cv_bridge.h>
#include <stdio.h>
#include <iostream>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/highgui/highgui.hpp>
```

### Functions

- `int main (int argc, char **argv)`

## 4.8.1 Function Documentation

4.8.1.1 `int main ( int argc, char ** argv )`



# Index

- ~Camera
  - Camera, [7](#)
- /home/daniela/code/src/eye\_in\_hand/include/Camera\_ -  
code.h, [16](#)
- /home/daniela/code/src/eye\_in\_hand/include/calibration\_ -  
h, [15](#)
- /home/daniela/code/src/eye\_in\_hand/include/converter\_ -  
\_file.hpp, [19](#)
- /home/daniela/code/src/eye\_in\_hand/include/geometry\_ -  
\_function.hpp, [21](#)
- /home/daniela/code/src/eye\_in\_hand/include/plane\_ -  
estimate.hpp, [22](#)
- /home/daniela/code/src/eye\_in\_hand/src/Camera\_ -  
code.cpp, [23](#)
- /home/daniela/code/src/eye\_in\_hand/src/calibration\_ -  
cpp, [22](#)
- /home/daniela/code/src/eye\_in\_hand/src/main\_image\_ -  
.cpp, [23](#)
- ASYMMETRIC\_CIRCLES\_GRID
  - Settings, [12](#)
- arrived\_cam
  - Camera, [9](#)
- aspectRatio
  - Settings, [12](#)
- atImageList
  - Settings, [12](#)
- boardSize
  - Settings, [12](#)
- Bot\_C
  - Camera::Obj, [10](#)
- Botton\_2frame
  - Camera::Obj, [10](#)
- BottonCHosen
  - Camera, [9](#)
- bwriteExtrinsics
  - Settings, [13](#)
- bwritePoints
  - Settings, [13](#)
- CALIBRATED
  - calibration.h, [16](#)
- CAMERA
  - Settings, [12](#)
- CAPTURING
  - calibration.h, [16](#)
- CHESSBOARD
  - Settings, [12](#)
- CIRCLES\_GRID
  - Settings, [12](#)
- calibFixPrincipalPoint
  - Settings, [13](#)
- calibZeroTangentDist
  - Settings, [13](#)
- calibration.h
  - CALIBRATED, [16](#)
  - CAPTURING, [16](#)
  - DETECTION, [16](#)
- calibration.cpp
  - main, [22](#)
  - runCalibrationAndSave, [22](#)
- calibration.h
  - runCalibrationAndSave, [16](#)
- calibrationPattern
  - Settings, [13](#)
- CallBackFunc
  - Camera, [7](#)
- cam\_cx
  - Camera, [9](#)
- cam\_cy
  - Camera, [9](#)
- cam\_d0
  - Camera, [9](#)
- cam\_d1
  - Camera, [9](#)
- cam\_d2
  - Camera, [9](#)
- cam\_d3
  - Camera, [9](#)
- cam\_d4
  - Camera, [9](#)
- cam\_fx
  - Camera, [9](#)
- cam\_fy
  - Camera, [9](#)
- Cam\_par\_distortion
  - Camera, [9](#)
- Camera, [5](#)
  - ~Camera, [7](#)
  - arrived\_cam, [9](#)
  - BottonCHosen, [9](#)
  - CallBackFunc, [7](#)
  - cam\_cx, [9](#)
  - cam\_cy, [9](#)
  - cam\_d0, [9](#)
  - cam\_d1, [9](#)
  - cam\_d2, [9](#)
  - cam\_d3, [9](#)

- cam\_d4, 9
- cam\_fx, 9
- cam\_fy, 9
- Cam\_par\_distortion, 9
- Camera, 7
- Camera2\_S03, 9
- Camera\_Matrix, 9
- camera\_topic\_, 9
- ControllCamera, 7
- ConvertPointFromWordToCam, 7
- count\_n\_passi, 9
- DetectWithSift, 7
- FillCamMatrixPose, 7
- FindAMinDistanceButton, 7
- FindBottonPos3D, 7
- FindContours, 8
- first\_Step, 9
- FirstCalibration, 9
- frame1\_, 9
- frame\_so3\_ptam, 9
- frame\_w\_c, 9
- ImageConverter, 8
- InfoKf3d, 8
- it\_, 9
- KeyPointIm1Match, 9
- KeypointIm2, 9
- move\_camera\_end, 9
- Move\_robot, 9
- movewebcamrobot, 9
- nh, 10
- pos\_object, 10
- press\_button, 10
- ProjectPointAndFindPosBot3d, 8
- ptam\_kf3d, 10
- ptam\_sub, 10
- Ptamkf3d, 10
- RobotMove, 8
- SOtreCamera, 8
- SaveFirst, 10
- scala, 10
- scene, 10
- ShapeDetect, 8
- So3\_prev\_ptam, 10
- sub, 10
- sub\_ptam\_2, 10
- Triangulation, 8
- Camera2\_S03
  - Camera, 9
- Camera::Obj, 10
  - Bot\_C, 10
  - Botton\_2frame, 10
  - Center\_, 10
  - descr\_, 11
  - figure\_, 11
  - keyp\_, 11
  - Pos3d\_, 11
- Camera\_Matrix
  - Camera, 9
- Camera\_code.cpp
  - main, 23
  - rng, 23
  - ScalaReturn, 23
- Camera\_code.h
  - EstimatePlane, 17
  - FindACenter, 17
  - FindMaxValue, 17
  - FromCvPointToEigen, 17
  - FromEigenVectorToCvPPoint, 19
  - FromMatToEigen, 19
  - Media, 19
  - ScalaReturn, 19
  - setLabel, 19
- camera\_topic\_
  - Camera, 9
- cameraID
  - Settings, 13
- Center\_
  - Camera::Obj, 10
- ControllCamera
  - Camera, 7
- ConvertPointFromWordToCam
  - Camera, 7
- converter\_file.hpp
  - FromCvPointToEigen, 20
  - FromEigenVectorToCvPPoint, 21
  - FromMatToEigen, 21
- count\_n\_passi
  - Camera, 9
- DETECTION
  - calibration.h, 16
- delay
  - Settings, 13
- descr\_
  - Camera::Obj, 11
- DetectWithSift
  - Camera, 7
- EstimatePlane
  - Camera\_code.h, 17
  - plane\_estimate.hpp, 22
- figure\_
  - Camera::Obj, 11
- FillCamMatrixPose
  - Camera, 7
- FindACenter
  - Camera\_code.h, 17
  - geometry\_function.hpp, 21
- FindAMinDistanceButton
  - Camera, 7
- FindBottonPos3D
  - Camera, 7
- FindContours
  - Camera, 8
- FindMaxValue
  - Camera\_code.h, 17

- geometry\_function.hpp, 21
- first\_Step
  - Camera, 9
- FirstCalibration
  - Camera, 9
- flag
  - Settings, 13
- flipVertical
  - Settings, 13
- frame1\_
  - Camera, 9
- frame\_so3\_ptam
  - Camera, 9
- frame\_w\_c
  - Camera, 9
- FromCvPointToEigen
  - Camera\_code.h, 17
  - converter\_file.hpp, 20
- FromEigenVectorToCvPoint
  - Camera\_code.h, 19
  - converter\_file.hpp, 21
- FromMatToEigen
  - Camera\_code.h, 19
  - converter\_file.hpp, 21
- geometry\_function.hpp
  - FindACenter, 21
  - FindMaxValue, 21
  - setLabel, 22
- goodInput
  - Settings, 13
- IMAGE\_LIST
  - Settings, 12
- INVALID
  - Settings, 12
- ImageConverter
  - Camera, 8
- imageList
  - Settings, 13
- InfoKf3d
  - Camera, 8
- input
  - Settings, 13
- inputCapture
  - Settings, 13
- InputType
  - Settings, 12
- inputType
  - Settings, 13
- interpret
  - Settings, 12
- it\_
  - Camera, 9
- KeyPointIm1Match
  - Camera, 9
- keyp\_
  - Camera::Obj, 11
- KeypointIm2
  - Camera, 9
- main
  - calibration.cpp, 22
  - Camera\_code.cpp, 23
  - main\_image\_.cpp, 24
- main\_image\_.cpp
  - main, 24
- Media
  - Camera\_code.h, 19
- move\_camera\_end
  - Camera, 9
- Move\_robot
  - Camera, 9
- movewebcamrobot
  - Camera, 9
- NOT\_EXISTING
  - Settings, 12
- nextImage
  - Settings, 12
- nh
  - Camera, 10
- nrFrames
  - Settings, 13
- outputFileName
  - Settings, 13
- Pattern
  - Settings, 12
- patternToUse
  - Settings, 13
- plane\_estimate.hpp
  - EstimatePlane, 22
- Pos3d\_
  - Camera::Obj, 11
- pos\_object
  - Camera, 10
- press\_button
  - Camera, 10
- ProjectPointAndFindPosBot3d
  - Camera, 8
- ptam\_kf3d
  - Camera, 10
- ptam\_sub
  - Camera, 10
- Ptamkf3d
  - Camera, 10
- read
  - Settings, 12
- readStringList
  - Settings, 12
- rng
  - Camera\_code.cpp, 23
- RobotMove
  - Camera, 8

- runCalibrationAndSave
  - calibration.cpp, 22
  - calibration.h, 16
- SOTreCamera
  - Camera, 8
- SaveFirst
  - Camera, 10
- scala
  - Camera, 10
- ScalaReturn
  - Camera\_code.cpp, 23
  - Camera\_code.h, 19
- scene
  - Camera, 10
- setLabel
  - Camera\_code.h, 19
  - geometry\_function.hpp, 22
- Settings, 11
  - ASYMMETRIC\_CIRCLES\_GRID, 12
  - aspectRatio, 12
  - atImageList, 12
  - boardSize, 12
  - bwriteExtrinsics, 13
  - bwritePoints, 13
  - CAMERA, 12
  - CHESSBOARD, 12
  - CIRCLES\_GRID, 12
  - calibFixPrincipalPoint, 13
  - calibZeroTangentDist, 13
  - calibrationPattern, 13
  - cameraID, 13
  - delay, 13
  - flag, 13
  - flipVertical, 13
  - goodInput, 13
  - IMAGE\_LIST, 12
  - INVALID, 12
  - imageList, 13
  - input, 13
  - inputCapture, 13
  - InputType, 12
  - inputType, 13
  - interprate, 12
  - NOT\_EXISTING, 12
  - nextImage, 12
  - nrFrames, 13
  - outputFileName, 13
  - Pattern, 12
  - patternToUse, 13
  - read, 12
  - readStringList, 12
  - Settings, 12
  - showUndistorsed, 13
  - squareSize, 13
  - VIDEO\_FILE, 12
  - write, 12
- ShapeDetect
  - Camera, 8
- showUndistorsed
  - Settings, 13
- So3\_prev\_ptam
  - Camera, 10
- squareSize
  - Settings, 13
- sub
  - Camera, 10
- sub\_ptam\_2
  - Camera, 10
- Triangulation
  - Camera, 8
- VIDEO\_FILE
  - Settings, 12
- write
  - Settings, 12