VINS MONO COMMENTED

1.0.0

Generated by Doxygen 1.8.13

Contents

Chapter 1

readme

part of `camodocal`

`Google Ceres` is needed.

Calibration:

Use `intrinsic_calib.cc` to calibrate your camera.

Undistortion:

See `Camera.h` for general interface:

- liftProjective: Lift points from the image plane to the projective space.
- spaceToPlane: Projects 3D points to the image plane (Pi function)

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

- camodocal::Camera ??
  - camodocal::CataCamera ??
  - camodocal::EquidistantCamera ??
  - camodocal::OCAMCamera ??
  - camodocal::PinholeCamera ??
- camodocal::CameraCalibration ??
- camodocal::CameraFactory ??
- CameraPoseVisualization ??
- camodocal::Chessboard ??
- camodocal::ChessboardCorner ??
- camodocal::ChessboardQuad ??
- camodocal::ComprehensionError< CameraT > ??
- CostFunction
  - MarginalizationFactor ??
- camodocal::CostFunctionFactory ??
- Estimator ??
- FeatureManager ??
- FeaturePerFrame ??
- FeaturePerId ??
- FileSystemHelper ??
- GlobalSFM ??
- ImageFrame ??
- InitialEXRotation ??
- IntegrationBase ??
- LocalParameterization
  - camodocal::EigenQuaternionParameterization ??
  - PoseLocalParameterization ??
- MarginalizationInfo ??
- MotionEstimator ??
- camodocal::Camera::Parameters ??
  - camodocal::CataCamera::Parameters ??
  - camodocal::EquidistantCamera::Parameters ??
  - camodocal::OCAMCamera::Parameters ??
  - camodocal::PinholeCamera::Parameters ??
- camodocal::ReprojectionError1< CameraT > ??

camodocal::ReprojectionError2< CameraT > . . . . **??**
camodocal::ReprojectionError3< CameraT > . . . . **??**
ReprojectionError3D . . . . **??**
ResidualBlockInfo . . . . **??**
SFMFeature . . . . **??**
SizedCostFunction
- IMUFactor . . . . **??**
- ProjectionFactor . . . . **??**
- ProjectionTdFactor . . . . **??**

camodocal::StereoReprojectionError< CameraT > . . . . **??**
ThreadsStruct . . . . **??**
TicToc . . . . **??**
camodocal::Transform . . . . **??**
Utility::uint_< N > . . . . **??**
Utility . . . . **??**
vector
- Spline . . . . **??**

Chapter 3

Class Index

3.1 Class List

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

camodocal::Camera
Base class for all camera model . . . . ??
camodocal::CameraCalibration . . . . ??
camodocal::CameraFactory . . . . ??
CameraPoseVisualization . . . . ??
camodocal::CataCamera . . . . ??
camodocal::Chessboard . . . . ??
camodocal::ChessboardCorner . . . . ??
camodocal::ChessboardQuad . . . . ??
camodocal::ComprehensionError< CameraT > . . . . ??
camodocal::CostFunctionFactory . . . . ??
camodocal::EigenQuaternionParameterization . . . . ??
camodocal::EquidistantCamera . . . . ??
Estimator . . . . ??
FeatureManager . . . . ??
FeaturePerFrame . . . . ??
FeaturePerId . . . . ??
FileSystemHelper . . . . ??
GlobalSFM . . . . ??
ImageFrame . . . . ??
IMUFactor . . . . ??
InitialEXRotation . . . . ??
IntegrationBase . . . . ??
MarginalizationFactor . . . . ??
MarginalizationInfo . . . . ??
MotionEstimator . . . . ??
camodocal::OCAMCamera . . . . ??
camodocal::CataCamera::Parameters . . . . ??
camodocal::EquidistantCamera::Parameters . . . . ??
camodocal::OCAMCamera::Parameters . . . . ??
camodocal::PinholeCamera::Parameters . . . . ??
camodocal::Camera::Parameters
Nested class for camera parameters . . . . ??
camodocal::PinholeCamera . . . . ??
PoseLocalParameterization . . . . ??

ProjectionFactor . . . . ??
ProjectionTdFactor . . . . ??
camodocal::ReprojectionError1< CameraT > . . . . ??
camodocal::ReprojectionError2< CameraT > . . . . ??
camodocal::ReprojectionError3< CameraT > . . . . ??
ReprojectionError3D . . . . ??
ResidualBlockInfo . . . . ??
SFMFeature . . . . ??
Spline . . . . ??
camodocal::StereoReprojectionError< CameraT > . . . . ??
ThreadsStruct . . . . ??
TicToc . . . . ??
camodocal::Transform . . . . ??
Utility::uint_< N > . . . . ??
Utility . . . . ??

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

camera_model/include/camodocal/calib/**CameraCalibration.h** . . . . ??
camera_model/include/camodocal/camera_models/Camera.h . . . . ??
camera_model/include/camodocal/camera_models/**CameraFactory.h** . . . . ??
camera_model/include/camodocal/camera_models/**CataCamera.h** . . . . ??
camera_model/include/camodocal/camera_models/**CostFunctionFactory.h** . . . . ??
camera_model/include/camodocal/camera_models/**EquidistantCamera.h** . . . . ??
camera_model/include/camodocal/camera_models/**PinholeCamera.h** . . . . ??
camera_model/include/camodocal/camera_models/**ScaramuzzaCamera.h** . . . . ??
camera_model/include/camodocal/chessboard/**Chessboard.h** . . . . ??
camera_model/include/camodocal/chessboard/**ChessboardCorner.h** . . . . ??
camera_model/include/camodocal/chessboard/**ChessboardQuad.h** . . . . ??
camera_model/include/camodocal/chessboard/**Spline.h** . . . . ??
camera_model/include/camodocal/gpl/**EigenQuaternionParameterization.h** . . . . ??
camera_model/include/camodocal/gpl/**EigenUtils.h** . . . . ??
camera_model/include/camodocal/gpl/**gpl.h** . . . . ??
camera_model/include/camodocal/sparse_graph/**Transform.h** . . . . ??
camera_model/src/camera_models/Camera.cc
Base class of all other camera class . . . . ??
vins_estimator/src/**estimator.h** . . . . ??
vins_estimator/src/**feature_manager.h** . . . . ??
vins_estimator/src/**parameters.h** . . . . ??
vins_estimator/src/factor/**imu_factor.h** . . . . ??
vins_estimator/src/factor/**integration_base.h** . . . . ??
vins_estimator/src/factor/**marginalization_factor.h** . . . . ??
vins_estimator/src/factor/**pose_local_parameterization.h** . . . . ??
vins_estimator/src/factor/**projection_factor.h** . . . . ??
vins_estimator/src/factor/**projection_td_factor.h** . . . . ??
vins_estimator/src/initial/**initial_alignment.h** . . . . ??
vins_estimator/src/initial/**initial_ex_rotation.h** . . . . ??
vins_estimator/src/initial/**initial_sfm.h** . . . . ??
vins_estimator/src/initial/**solve_5pts.h** . . . . ??
vins_estimator/src/utility/**CameraPoseVisualization.h** . . . . ??
vins_estimator/src/utility/**tic_toc.h** . . . . ??
vins_estimator/src/utility/**utility.h** . . . . ??
vins_estimator/src/utility/**visualization.h** . . . . ??

Chapter 5

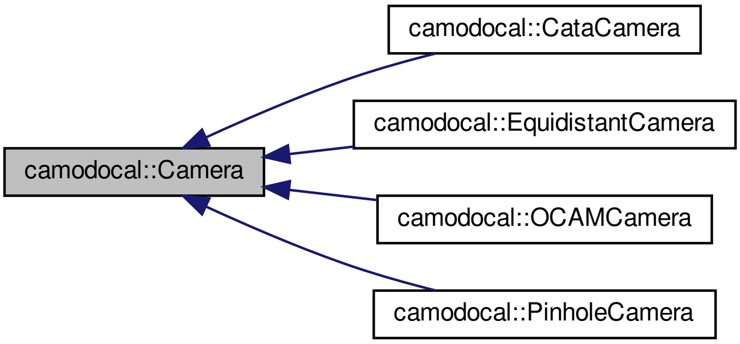
Class Documentation

5.1 camodocal::Camera Class Reference

base class for all camera model

`#include <Camera.h>`

Inheritance diagram for camodocal::Camera:



Classes

- class Parameters

  nested class for camera parameters

Public Types

- enum ModelType { **KANNALA_BRANDT**, **MEI**, **PINHOLE**, **SCARAMUZZA** }

  enumerate variable of camera model

Public Member Functions

- virtual ModelType modelType (void) const =0

  virtual type of function modelType
- virtual const std::string & cameraName (void) const =0

  virtual type of funtion cameraName
- virtual int imageWidth (void) const =0

  virtual type of function imageWidth
- virtual int imageHeight (void) const =0

  virtual type of function imageHeight
- virtual cv::Mat & mask (void)

  virtual function of image Mask
- virtual const cv::Mat & mask (void) const

  virtual function of image Mask
- virtual void estimateIntrinsics (const cv::Size &boardSize, const std::vector< std::vector< cv::Point3f > > &objectPoints, const std::vector< std::vector< cv::Point2f > > &imagePoints)=0

  virtual function of camera intrinsics
- virtual void estimateExtrinsics (const std::vector< cv::Point3f > &objectPoints, const std::vector< cv::Point2f > &imagePoints, cv::Mat &rvec, cv::Mat &tvec) const

  calculate extrinsics with unit intrinsics
- virtual void liftSphere (const Eigen::Vector2d &p, Eigen::Vector3d &P) const =0

  Lift points from the image plane to the sphere.
- virtual void liftProjective (const Eigen::Vector2d &p, Eigen::Vector3d &P) const =0

  Lift points from the image plane to the projective space.
- virtual void spaceToPlane (const Eigen::Vector3d &P, Eigen::Vector2d &p) const =0

  Projects 3D points to the image plane (Pi function)
- virtual void **undistToPlane** (const Eigen::Vector2d &p_u, Eigen::Vector2d &p) const =0
- virtual cv::Mat **initUndistortRectifyMap** (cv::Mat &map1, cv::Mat &map2, float fx=-1.0f, float fy=-1.0f, cv::↩ Size imageSize=cv::Size(0, 0), float cx=-1.0f, float cy=-1.0f, cv::Mat rmat=cv::Mat::eye(3, 3, CV_32F)) const =0
- virtual int parameterCount (void) const =0

  pure virtual function of parameter count
- virtual void readParameters (const std::vector< double > &parameters)=0

  pure virtual function of reading parameters
- virtual void writeParameters (std::vector< double > &parameters) const =0

  pure virtual function of writing parameters
- virtual void writeParametersToYamlFile (const std::string &filename) const =0

  pure virtual function of writing parameters to YAML file
- virtual std::string parametersToString (void) const =0

  pure virtual of converting parameters to string
- double reprojectionDist (const Eigen::Vector3d &P1, const Eigen::Vector3d &P2) const

  Calculates the reprojection distance between points.
- double reprojectionError (const std::vector< std::vector< cv::Point3f > > &objectPoints, const std::vector< std::vector< cv::Point2f > > &imagePoints, const std::vector< cv::Mat > &rvecs, const std::vector< cv::Mat > &tvecs, cv::OutputArray perViewErrors=cv::noArray()) const

  calculate average reprojection error of all points in all frames (total error divede by total points) with 3D points and 2D points
- double reprojectionError (const Eigen::Vector3d &P, const Eigen::Quaterniond &camera_q, const Eigen::↩ Vector3d &camera_t, const Eigen::Vector2d &observed_p) const

  calculate reprojection error of one 3D point with camera pose P & Q
- void projectPoints (const std::vector< cv::Point3f > &objectPoints, const cv::Mat &rvec, const cv::Mat &tvec, std::vector< cv::Point2f > &imagePoints) const

  project 3D points to 2d plane

Protected Attributes

- cv::Mat m_mask

  image mask

5.1.1 Detailed Description

base class for all camera model

5.1.2 Member Function Documentation

5.1.2.1 estimateExtrinsics()

```
void camodocal::Camera::estimateExtrinsics (
             const std::vector< cv::Point3f > & objectPoints,
             const std::vector< cv::Point2f > & imagePoints,
             cv::Mat & rvec,
             cv::Mat & tvec ) const  [virtual]
```

calculate extrinsics with unit intrinsics

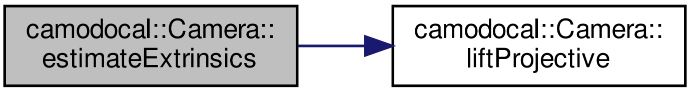
Parameters

objectPoints	3D points on object
imagePoints	2D point on image plane
rvec	rotation vector
tvec	transformation vector

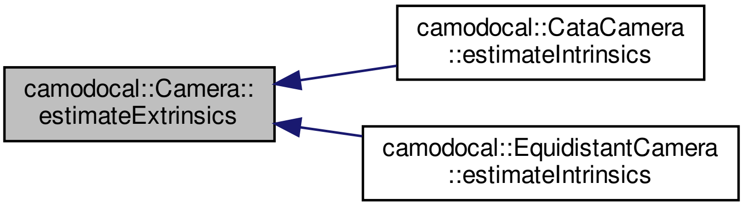
References liftProjective().

Referenced by camodocal::CataCamera::estimateIntrinsics(), and camodocal::EquidistantCamera::estimateIntrinsics().

Here is the call graph for this function:



Here is the caller graph for this function:



5.1.2.2 projectPoints()

```
void camodocal::Camera::projectPoints (
            const std::vector< cv::Point3f > & objectPoints,
            const cv::Mat & rvec,
            const cv::Mat & tvec,
            std::vector< cv::Point2f > & imagePoints ) const
```

project 3D points to 2d plane

Parameters

objectPoints	
rvec	
tvec	
imagePoints	

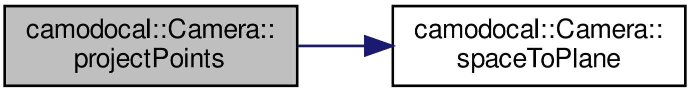
reserve space for imagePoints according to objectPoints

convert from rotation vector to rotation matrix

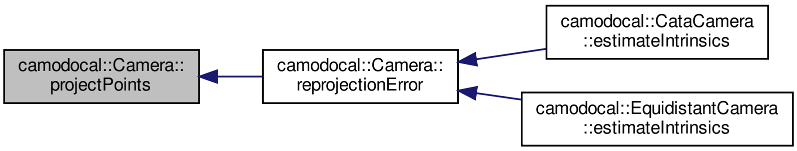
References spaceToPlane().

Referenced by reprojectionError().

Here is the call graph for this function:



Here is the caller graph for this function:



5.1.2.3 reprojectionDist()

```
double camodocal::Camera::reprojectionDist (
            const Eigen::Vector3d & P1,
            const Eigen::Vector3d & P2 ) const
```

Calculates the reprojection distance between points.

Parameters

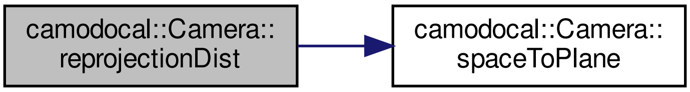
P1	first 3D point coordinates
P2	second 3D point coordinates

Returns

euclidean distance in the plane

References spaceToPlane().

Here is the call graph for this function:



5.1.2.4 reprojectionError() [1/2]

```
double camodocal::Camera::reprojectionError (
            const std::vector< std::vector< cv::Point3f > > & objectPoints,
            const std::vector< std::vector< cv::Point2f > > & imagePoints,
            const std::vector< cv::Mat > & rvecs,
            const std::vector< cv::Mat > & tvecs,
            cv::OutputArray perViewErrors = cv::noArray() ) const
```

calculate average reprojection error of all points in all frames (total error divede by total points) with 3D points and 2D points

Parameters

objectPoints	3D coordinates of object points
imagePoints	2D coordinates of image points
rvecs	rotation vectors
tvecs	transformation vectors
perViewErrors	

Returns

:

count of image frames

current points in all image frames

check whether preVireError needed to be computed

imageCount ∗ 1

using getMat() to convert from cv::OutputArray to cv::Mat

project points of frame i to image plane

calculate repeojection error of frame i()

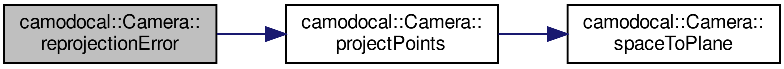
calculate average reprojection error of frame i

return average reprojection error

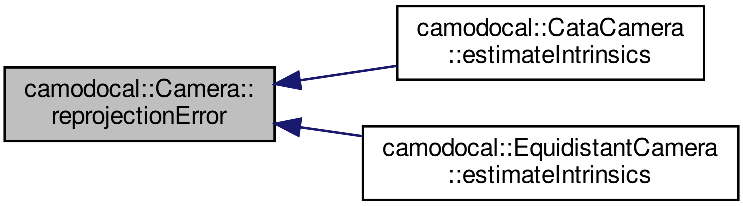
References projectPoints().

Referenced by camodocal::CataCamera::estimateIntrinsics(), and camodocal::EquidistantCamera::estimateIntrinsics().

Here is the call graph for this function:



Here is the caller graph for this function:



5.1.2.5 reprojectionError() [2/2]

```
double camodocal::Camera::reprojectionError (
            const Eigen::Vector3d & P,
            const Eigen::Quaterniond & camera_q,
            const Eigen::Vector3d & camera_t,
            const Eigen::Vector2d & observed_p ) const
```

calculate reprojection error of one 3D point with camera pose P & Q

Parameters

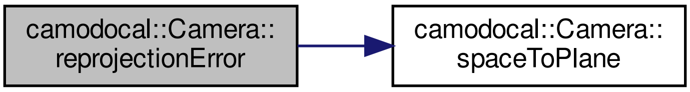
P	3D coordinates of point
camera_q	quaternion of camera pose
camera_t	translation of camera pose
observed↩_p	corresponding image point

Returns

:

References spaceToPlane().

Here is the call graph for this function:



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

- camera_model/include/camodocal/camera_models/Camera.h
- camera_model/src/camera_models/Camera.cc

5.2 camodocal::CameraCalibration Class Reference

Public Member Functions

- **CameraCalibration** (Camera::ModelType modelType, const std::string &cameraName, const cv::Size &imageSize, const cv::Size &boardSize, float squareSize)
- void **clear** (void)
- void **addChessboardData** (const std::vector< cv::Point2f > &corners)
- bool **calibrate** (void)
- int **sampleCount** (void) const
- std::vector< std::vector< cv::Point2f > > & **imagePoints** (void)
- const std::vector< std::vector< cv::Point2f > > & **imagePoints** (void) const
- std::vector< std::vector< cv::Point3f > > & **scenePoints** (void)
- const std::vector< std::vector< cv::Point3f > > & **scenePoints** (void) const
- CameraPtr & **camera** (void)
- const CameraConstPtr **camera** (void) const
- Eigen::Matrix2d & **measurementCovariance** (void)
- const Eigen::Matrix2d & **measurementCovariance** (void) const
- cv::Mat & **cameraPoses** (void)
- const cv::Mat & **cameraPoses** (void) const
- void **drawResults** (std::vector< cv::Mat > &images) const
- void **writeParams** (const std::string &filename) const
- bool **writeChessboardData** (const std::string &filename) const
- bool **readChessboardData** (const std::string &filename)
- void **setVerbose** (bool verbose)

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

- camera_model/include/camodocal/calib/CameraCalibration.h
- camera_model/src/calib/CameraCalibration.cc

5.3 camodocal::CameraFactory Class Reference

Public Member Functions

- CameraPtr generateCamera (Camera::ModelType modelType, const std::string &cameraName, cv::Size imageSize) const

  generate Camera object according to modelType
- CameraPtr generateCameraFromYamlFile (const std::string &filename)

  generate camera object according to YAML file

Static Public Member Functions

- static boost::shared_ptr< CameraFactory > instance (void)

  Get point of cameraFactory.

5.3.1 Member Function Documentation

5.3.1.1 generateCamera()

```
CameraPtr camodocal::CameraFactory::generateCamera (
            Camera::ModelType modelType,
            const std::string & cameraName,
            cv::Size imageSize ) const
```

generate Camera object according to modelType

Parameters

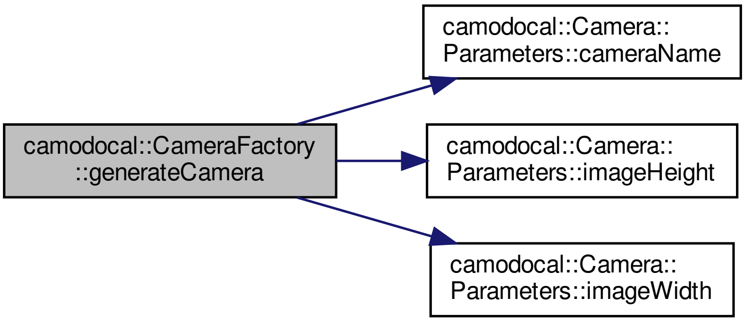
modelType	camera model
cameraName	camera name
imageSize	image Size

Returns

: shared_ptr of camera

References camodocal::Camera::Parameters::cameraName(), camodocal::Camera::Parameters::imageHeight(), and camodocal::Camera::Parameters::imageWidth().

Here is the call graph for this function:


camodocal::CameraFactory ::generateCamera
camodocal::Camera:: Parameters::cameraName
camodocal::Camera:: Parameters::imageHeight
camodocal::Camera:: Parameters::imageWidth

5.3.1.2 generateCameraFromYamlFile()

```
CameraPtr camodocal::CameraFactory::generateCameraFromYamlFile (
            const std::string & filename )
```

generate camera object according to YAML file

Parameters

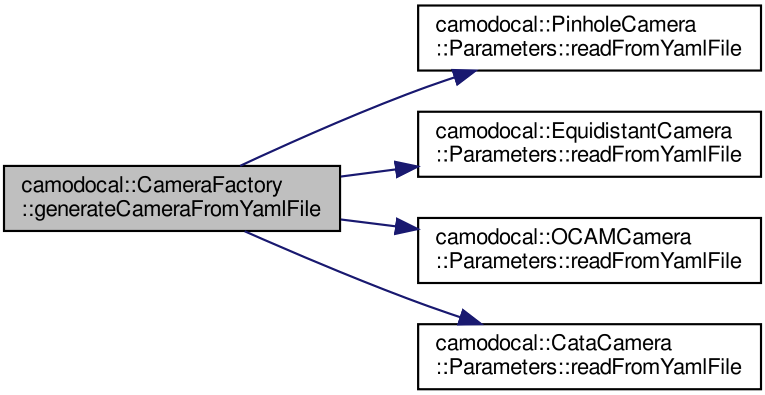
filename	name of YAML file

Returns

: shared_ptr of camera

References camodocal::PinholeCamera::Parameters::readFromYamlFile(), camodocal::EquidistantCamera::↩ Parameters::readFromYamlFile(), camodocal::OCAMCamera::Parameters::readFromYamlFile(), and camodocal↩ ::CataCamera::Parameters::readFromYamlFile().

Here is the call graph for this function:


camodocal::CameraFactory ::generateCameraFromYamlFile
camodocal::PinholeCamera ::Parameters::readFromYamlFile
camodocal::EquidistantCamera ::Parameters::readFromYamlFile
camodocal::OCAMCamera ::Parameters::readFromYamlFile
camodocal::CataCamera ::Parameters::readFromYamlFile

5.3.1.3 instance()

```
boost::shared_ptr< CameraFactory > camodocal::CameraFactory::instance (
            void ) [static]
```

Get point of cameraFactory.

Returns

: shared_ptr of CameraFactory

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

- camera_model/include/camodocal/camera_models/CameraFactory.h
- camera_model/src/camera_models/CameraFactory.cc

5.4 CameraPoseVisualization Class Reference

Public Member Functions

- **CameraPoseVisualization** (float r, float g, float b, float a)
- void **setImageBoundaryColor** (float r, float g, float b, float a=1.0)
- void **setOpticalCenterConnectorColor** (float r, float g, float b, float a=1.0)
- void **setScale** (double s)
- void **setLineWidth** (double width)
- void **add_pose** (const Eigen::Vector3d &p, const Eigen::Quaterniond &q)
- void **reset** ()
- void **publish_by** (ros::Publisher &pub, const std_msgs::Header &header)
- void **add_edge** (const Eigen::Vector3d &p0, const Eigen::Vector3d &p1)
- void **add_loopedge** (const Eigen::Vector3d &p0, const Eigen::Vector3d &p1)

Public Attributes

- std::string **m_marker_ns**

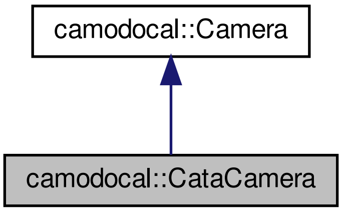
The documentation for this class was generated from the following files:

- vins_estimator/src/utility/CameraPoseVisualization.h
- vins_estimator/src/utility/CameraPoseVisualization.cpp

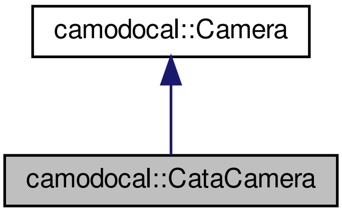
5.5 camodocal::CataCamera Class Reference

```
#include <CataCamera.h>
```

Inheritance diagram for camodocal::CataCamera:



Collaboration diagram for camodocal::CataCamera:



Classes

- class Parameters

Public Member Functions

- CataCamera (const std::string &cameraName, int imageWidth, int imageHeight, double xi, double k1, double k2, double p1, double p2, double gamma1, double gamma2, double u0, double v0)

  Constructor from the projection model parameters.
- CataCamera (const Parameters &params)

  Constructor from the projection model parameters.
- Camera::ModelType modelType (void) const

  virtual type of function modelType
- const std::string & cameraName (void) const

  virtual type of funtion cameraName
- int imageWidth (void) const

  virtual type of function imageWidth
- int imageHeight (void) const

  virtual type of function imageHeight
- void estimateIntrinsics (const cv::Size &boardSize, const std::vector< std::vector< cv::Point3f > > &objectPoints, const std::vector< std::vector< cv::Point2f > > &imagePoints)

  virtual function of camera intrinsics
- void liftSphere (const Eigen::Vector2d &p, Eigen::Vector3d &P) const

  Lifts a point from the image plane to the unit sphere.
- void liftProjective (const Eigen::Vector2d &p, Eigen::Vector3d &P) const

  Lifts a point from the image plane to its projective ray.
- void spaceToPlane (const Eigen::Vector3d &P, Eigen::Vector2d &p) const

  Project a 3D point (x,y,z) to the image plane in (u,v)
- void **spaceToPlane** (const Eigen::Vector3d &P, Eigen::Vector2d &p, Eigen::Matrix< double, 2, 3 > &J) const
- void undistToPlane (const Eigen::Vector2d &p_u, Eigen::Vector2d &p) const

  Projects an undistorted 2D point p_u to the image plane.
- void distortion (const Eigen::Vector2d &p_u, Eigen::Vector2d &d_u) const

  Apply distortion to input point (from the normalised plane)
- void distortion (const Eigen::Vector2d &p_u, Eigen::Vector2d &d_u, Eigen::Matrix2d &J) const

  Apply distortion to input point (from the normalised plane) and calculate Jacobian.

- void **initUndistortMap** (cv::Mat &map1, cv::Mat &map2, double fScale=1.0) const
- cv::Mat **initUndistortRectifyMap** (cv::Mat &map1, cv::Mat &map2, float fx=-1.0f, float fy=-1.0f, cv::Size imageSize=cv::Size(0, 0), float cx=-1.0f, float cy=-1.0f, cv::Mat rmat=cv::Mat::eye(3, 3, CV_32F)) const
- int parameterCount (void) const

  pure virtual function of parameter count
- const Parameters & **getParameters** (void) const
- void **setParameters** (const Parameters &parameters)
- void readParameters (const std::vector< double > &parameterVec)

  pure virtual function of reading parameters
- void writeParameters (std::vector< double > &parameterVec) const

  pure virtual function of writing parameters
- void writeParametersToYamlFile (const std::string &filename) const

  pure virtual function of writing parameters to YAML file
- std::string parametersToString (void) const

  pure virtual of converting parameters to string

Static Public Member Functions

- template<typename T >
  static void **spaceToPlane** (const T ∗const params, const T ∗const q, const T ∗const t, const Eigen::Matrix< T, 3, 1 > &P, Eigen::Matrix< T, 2, 1 > &p)

Additional Inherited Members

5.5.1 Detailed Description

C. Mei, and P. Rives, Single View Point Omnidirectional Camera Calibration from Planar Grids, ICRA 2007

5.5.2 Member Function Documentation

5.5.2.1 distortion() [1/2]

```
void camodocal::CataCamera::distortion (
            const Eigen::Vector2d & p_u,
            Eigen::Vector2d & d_u ) const
```

Apply distortion to input point (from the normalised plane)

Parameters

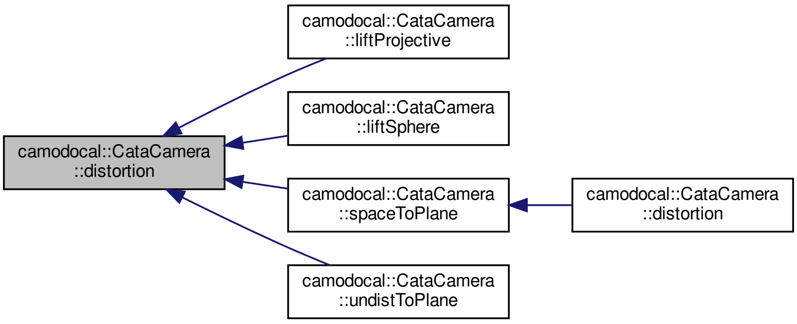
p_u	undistorted coordinates of point on the normalised plane

Returns

to obtain the distorted point: p_d = p_u + d_u

Referenced by liftProjective(), liftSphere(), spaceToPlane(), and undistToPlane().

Here is the caller graph for this function:


camodocal::CataCamera ::liftProjective
camodocal::CataCamera ::liftSphere
camodocal::CataCamera ::distortion
camodocal::CataCamera ::spaceToPlane
camodocal::CataCamera ::distortion
camodocal::CataCamera ::undistToPlane

5.5.2.2 distortion() [2/2]

```
void camodocal::CataCamera::distortion (
             const Eigen::Vector2d & p_u,
             Eigen::Vector2d & d_u,
             Eigen::Matrix2d & J ) const
```

Apply distortion to input point (from the normalised plane) and calculate Jacobian.

Parameters

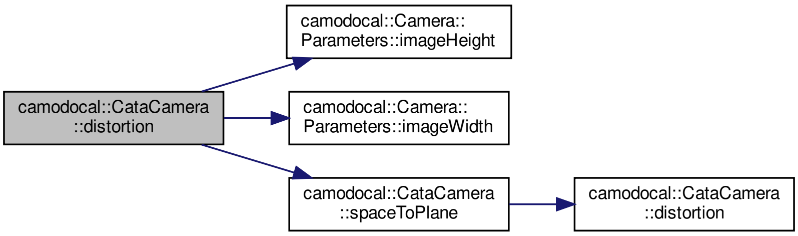
p↩ _u	undistorted coordinates of point on the normalised plane

Returns

to obtain the distorted point: p_d = p_u + d_u

References camodocal::Camera::Parameters::imageHeight(), camodocal::Camera::Parameters::imageWidth(), and spaceToPlane().

Here is the call graph for this function:


camodocal::Camera::Parameters::imageHeight
camodocal::CataCamera::distortion
camodocal::Camera::Parameters::imageWidth
camodocal::CataCamera::spaceToPlane
camodocal::CataCamera::distortion

5.5.2.3 liftProjective()

```
void camodocal::CataCamera::liftProjective (
             const Eigen::Vector2d & p,
             Eigen::Vector3d & P ) const  [virtual]
```

Lifts a point from the image plane to its projective ray.

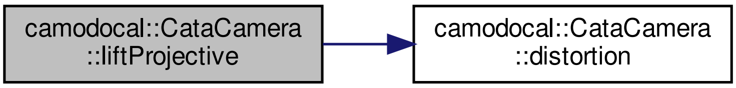
Parameters

p	image coordinates
P	coordinates of the projective ray

Implements camodocal::Camera.

References distortion().

Here is the call graph for this function:


camodocal::CataCamera::liftProjective
camodocal::CataCamera::distortion

5.5.2.4 liftSphere()

```
void camodocal::CataCamera::liftSphere (
             const Eigen::Vector2d & p,
             Eigen::Vector3d & P ) const  [virtual]
```

Lifts a point from the image plane to the unit sphere.

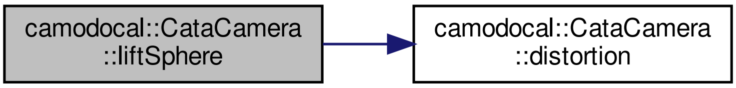
Parameters

p	image coordinates
P	coordinates of the point on the sphere

Implements camodocal::Camera.

References distortion().

Here is the call graph for this function:


camodocal::CataCamera ::liftSphere
camodocal::CataCamera ::distortion

5.5.2.5 spaceToPlane()

```
void camodocal::CataCamera::spaceToPlane (
             const Eigen::Vector3d & P,
             Eigen::Vector2d & p ) const  [virtual]
```

Project a 3D point (*x*,*y*,*z*) to the image plane in (*u*,*v*)

Parameters

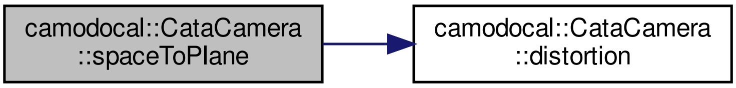
P	3D point coordinates
p	return value, contains the image point coordinates

Implements camodocal::Camera.

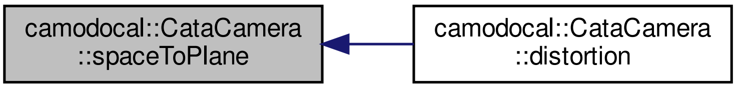
References distortion().

Referenced by distortion().

Here is the call graph for this function:


camodocal::CataCamera ::spaceToPlane
camodocal::CataCamera ::distortion

Here is the caller graph for this function:


camodocal::CataCamera ::spaceToPlane
camodocal::CataCamera ::distortion

5.5.2.6 undistToPlane()

```
void camodocal::CataCamera::undistToPlane (
             const Eigen::Vector2d & p_u,
             Eigen::Vector2d & p ) const  [virtual]
```

Projects an undistorted 2D point p_u to the image plane.

Parameters

p← _u	2D point coordinates

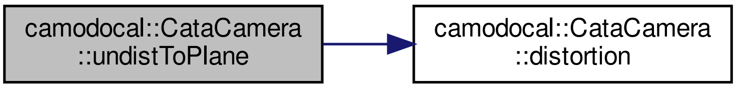
Returns

image point coordinates

Implements camodocal::Camera.

References distortion().

Here is the call graph for this function:


camodocal::CataCamera ::undistToPlane
camodocal::CataCamera ::distortion

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

- camera_model/include/camodocal/camera_models/CataCamera.h
- camera_model/src/camera_models/CataCamera.cc

5.6 camodocal::Chessboard Class Reference

Public Member Functions

- **Chessboard** (cv::Size boardSize, cv::Mat &image)
- void **findCorners** (bool useOpenCV=false)
- const std::vector< cv::Point2f > & **getCorners** (void) const
- bool **cornersFound** (void) const
- const cv::Mat & **getImage** (void) const
- const cv::Mat & **getSketch** (void) const

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

- camera_model/include/camodocal/chessboard/Chessboard.h
- camera_model/src/chessboard/Chessboard.cc

5.7 camodocal::ChessboardCorner Class Reference

Public Member Functions

- float **meanDist** (int &n) const

Public Attributes

- cv::Point2f **pt**
- int **row**
- int **column**
- bool **needsNeighbor**
- int **count**
- ChessboardCornerPtr **neighbors** [4]

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

- camera_model/include/camodocal/chessboard/ChessboardCorner.h

5.8 camodocal::ChessboardQuad Class Reference

Public Attributes

- int **count**
- int **group_idx**
- float **edge_len**
- ChessboardCornerPtr **corners** [4]
- ChessboardQuadPtr **neighbors** [4]
- bool **labeled**

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

- camera_model/include/camodocal/chessboard/ChessboardQuad.h

5.9 camodocal::ComprehensionError< CameraT > Class Template Reference

Public Member Functions

- EIGEN_MAKE_ALIGNED_OPERATOR_NEW **ComprehensionError** (const Eigen::Vector3d &observed_P, const Eigen::Vector2d &observed_p)
- template<typename T >
  bool **operator()** (const T ∗const intrinsic_params, const T ∗const q, const T ∗const t, T ∗residuals) const

Public Attributes

- Eigen::Vector3d **m_observed_P**
- Eigen::Vector2d **m_observed_p**
- Eigen::Matrix2d **m_sqrtPrecisionMat**

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

- camera_model/src/camera_models/CostFunctionFactory.cc

5.10 camodocal::CostFunctionFactory Class Reference

Public Member Functions

- ceres::CostFunction ∗ **generateCostFunction** (const CameraConstPtr &camera, const Eigen::Vector3d &observed_P, const Eigen::Vector2d &observed_p, int flags) const
- ceres::CostFunction ∗ **generateCostFunction** (const CameraConstPtr &camera, const Eigen::Vector3d &observed_P, const Eigen::Vector2d &observed_p, const Eigen::Matrix2d &sqrtPrecisionMat, int flags) const
- ceres::CostFunction ∗ **generateCostFunction** (const CameraConstPtr &camera, const Eigen::Vector2d &observed_p, int flags, bool optimize_cam_odo_z=true) const
- ceres::CostFunction ∗ **generateCostFunction** (const CameraConstPtr &camera, const Eigen::Vector2d &observed_p, const Eigen::Matrix2d &sqrtPrecisionMat, int flags, bool optimize_cam_odo_z=true) const
- ceres::CostFunction ∗ **generateCostFunction** (const CameraConstPtr &camera, const Eigen::Vector3d &odo_pos, const Eigen::Vector3d &odo_att, const Eigen::Vector2d &observed_p, int flags, bool optimize←↩_cam_odo_z=true) const
- ceres::CostFunction ∗ **generateCostFunction** (const CameraConstPtr &camera, const Eigen::Quaterniond &cam_odo_q, const Eigen::Vector3d &cam_odo_t, const Eigen::Vector3d &odo_pos, const Eigen::Vector3d &odo_att, const Eigen::Vector2d &observed_p, int flags) const
- ceres::CostFunction ∗ **generateCostFunction** (const CameraConstPtr &cameraLeft, const CameraConstPtr &cameraRight, const Eigen::Vector3d &observed_P, const Eigen::Vector2d &observed_p_left, const Eigen←↩::Vector2d &observed_p_right) const

Static Public Member Functions

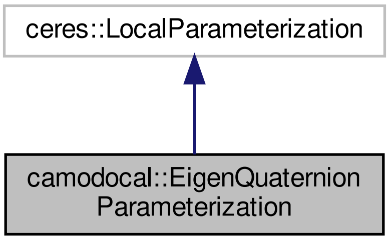
- static boost::shared_ptr< CostFunctionFactory > **instance** (void)

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

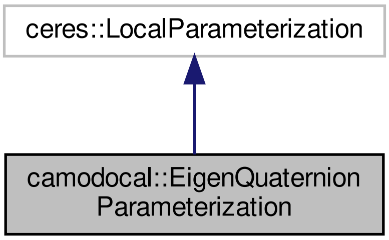
- camera_model/include/camodocal/camera_models/CostFunctionFactory.h
- camera_model/src/camera_models/CostFunctionFactory.cc

5.11 camodocal::EigenQuaternionParameterization Class Reference

Inheritance diagram for camodocal::EigenQuaternionParameterization:


ceres::LocalParameterization
camodocal::EigenQuaternion Parameterization

Collaboration diagram for camodocal::EigenQuaternionParameterization:


ceres::LocalParameterization
camodocal::EigenQuaternion Parameterization

Public Member Functions

- virtual bool **Plus** (const double ∗x, const double ∗delta, double ∗x_plus_delta) const
- virtual bool **ComputeJacobian** (const double ∗x, double ∗jacobian) const
- virtual int **GlobalSize** () const
- virtual int **LocalSize** () const

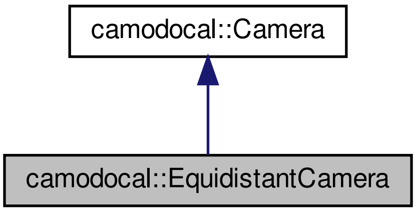
The documentation for this class was generated from the following files:

- camera_model/include/camodocal/gpl/EigenQuaternionParameterization.h
- camera_model/src/gpl/EigenQuaternionParameterization.cc

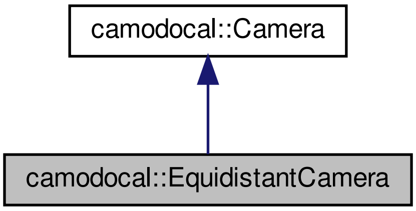
5.12 camodocal::EquidistantCamera Class Reference

```
#include <EquidistantCamera.h>
```

Inheritance diagram for camodocal::EquidistantCamera:



Collaboration diagram for camodocal::EquidistantCamera:



Classes

- class Parameters

Public Member Functions

- EquidistantCamera (const std::string &cameraName, int imageWidth, int imageHeight, double k2, double k3, double k4, double k5, double mu, double mv, double u0, double v0)

  Constructor from the projection model parameters.
- EquidistantCamera (const Parameters &params)

  Constructor from the projection model parameters.
- Camera::ModelType modelType (void) const

  virtual type of function modelType
- const std::string & cameraName (void) const

  virtual type of funtion cameraName

- int imageWidth (void) const

  virtual type of function imageWidth
- int imageHeight (void) const

  virtual type of function imageHeight
- void estimateIntrinsics (const cv::Size &boardSize, const std::vector< std::vector< cv::Point3f > > &object↩ Points, const std::vector< std::vector< cv::Point2f > > &imagePoints)

  virtual function of camera intrinsics
- virtual void liftSphere (const Eigen::Vector2d &p, Eigen::Vector3d &P) const

  Lifts a point from the image plane to the unit sphere.
- void liftProjective (const Eigen::Vector2d &p, Eigen::Vector3d &P) const

  Lifts a point from the image plane to its projective ray.
- void spaceToPlane (const Eigen::Vector3d &P, Eigen::Vector2d &p) const

  Project a 3D point (x,y,z) to the image plane in (u,v)
- void spaceToPlane (const Eigen::Vector3d &P, Eigen::Vector2d &p, Eigen::Matrix< double, 2, 3 > &J) const

  Project a 3D point to the image plane and calculate Jacobian.
- void undistToPlane (const Eigen::Vector2d &p_u, Eigen::Vector2d &p) const

  Projects an undistorted 2D point p_u to the image plane.
- void **initUndistortMap** (cv::Mat &map1, cv::Mat &map2, double fScale=1.0) const
- cv::Mat **initUndistortRectifyMap** (cv::Mat &map1, cv::Mat &map2, float fx=-1.0f, float fy=-1.0f, cv::Size imageSize=cv::Size(0, 0), float cx=-1.0f, float cy=-1.0f, cv::Mat rmat=cv::Mat::eye(3, 3, CV_32F)) const
- int parameterCount (void) const

  pure virtual function of parameter count
- const Parameters & **getParameters** (void) const
- void **setParameters** (const Parameters &parameters)
- void readParameters (const std::vector< double > &parameterVec)

  pure virtual function of reading parameters
- void writeParameters (std::vector< double > &parameterVec) const

  pure virtual function of writing parameters
- void writeParametersToYamlFile (const std::string &filename) const

  pure virtual function of writing parameters to YAML file
- std::string parametersToString (void) const

  pure virtual of converting parameters to string

Static Public Member Functions

- template<typename T >
  static void **spaceToPlane** (const T ∗const params, const T ∗const q, const T ∗const t, const Eigen::Matrix< T, 3, 1 > &P, Eigen::Matrix< T, 2, 1 > &p)

Additional Inherited Members

5.12.1 Detailed Description

J. Kannala, and S. Brandt, A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses, PAMI 2006

5.12.2 Member Function Documentation

5.12.2.1 liftProjective()

```
void camodocal::EquidistantCamera::liftProjective (
            const Eigen::Vector2d & p,
            Eigen::Vector3d & P ) const  [virtual]
```

Lifts a point from the image plane to its projective ray.

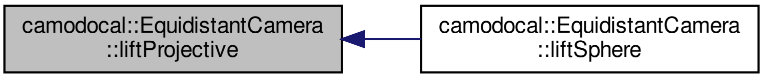
Parameters

p	image coordinates
P	coordinates of the projective ray

Implements camodocal::Camera.

Referenced by liftSphere().

Here is the caller graph for this function:


camodocal::EquidistantCamera ::liftProjective
camodocal::EquidistantCamera ::liftSphere

5.12.2.2 liftSphere()

```
void camodocal::EquidistantCamera::liftSphere (
            const Eigen::Vector2d & p,
            Eigen::Vector3d & P ) const  [virtual]
```

Lifts a point from the image plane to the unit sphere.

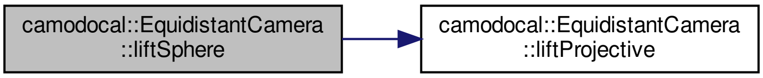
Parameters

p	image coordinates
P	coordinates of the point on the sphere

Implements camodocal::Camera.

References liftProjective().

Here is the call graph for this function:



5.12.2.3 spaceToPlane() [1/2]

```
void camodocal::EquidistantCamera::spaceToPlane (
             const Eigen::Vector3d & P,
             Eigen::Vector2d & p ) const  [virtual]
```

Project a 3D point (*x*,*y*,*z*) to the image plane in (*u*,*v*)

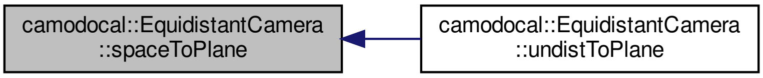
Parameters

P	3D point coordinates
p	return value, contains the image point coordinates

Implements camodocal::Camera.

Referenced by undistToPlane().

Here is the caller graph for this function:



5.12.2.4 spaceToPlane() [2/2]

```
void camodocal::EquidistantCamera::spaceToPlane (
             const Eigen::Vector3d & P,
             Eigen::Vector2d & p,
             Eigen::Matrix< double, 2, 3 > & J ) const
```

Project a 3D point to the image plane and calculate Jacobian.

Parameters

P	3D point coordinates
p	return value, contains the image point coordinates

5.12.2.5 undistToPlane()

```
void camodocal::EquidistantCamera::undistToPlane (
             const Eigen::Vector2d & p_u,
             Eigen::Vector2d & p ) const  [virtual]
```

Projects an undistorted 2D point p_u to the image plane.

Parameters

p↩ *_u*	2D point coordinates

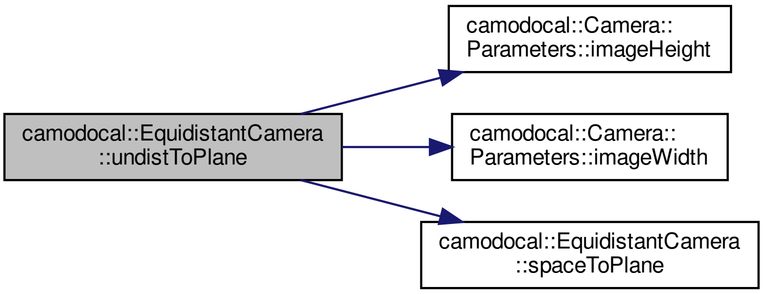
Returns

image point coordinates

Implements camodocal::Camera.

References camodocal::Camera::Parameters::imageHeight(), camodocal::Camera::Parameters::imageWidth(), and spaceToPlane().

Here is the call graph for this function:

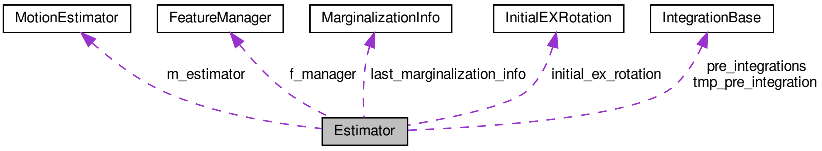

camodocal::EquidistantCamera ::undistToPlane
camodocal::Camera:: Parameters::imageHeight
camodocal::Camera:: Parameters::imageWidth
camodocal::EquidistantCamera ::spaceToPlane

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

- camera_model/include/camodocal/camera_models/EquidistantCamera.h
- camera_model/src/camera_models/EquidistantCamera.cc

5.13 Estimator Class Reference

Collaboration diagram for Estimator:



Public Types

- enum **SolverFlag** { **INITIAL**, **NON_LINEAR** }
- enum **MarginalizationFlag** { **MARGIN_OLD** = 0, **MARGIN_SECOND_NEW** = 1 }

Public Member Functions

- void **setParameter** ()
- void **processIMU** (double t, const Vector3d &linear_acceleration, const Vector3d &angular_velocity)
- void **processImage** (const map< int, vector< pair< int, Eigen::Matrix< double, 7, 1 >>>> &image, const std_msgs::Header &header)
- void **setReloFrame** (double _frame_stamp, int _frame_index, vector< Vector3d > &_match_points, Vector3d _relo_t, Matrix3d _relo_r)
- void **clearState** ()
- bool **initialStructure** ()
- bool **visualInitialAlign** ()
- bool **relativePose** (Matrix3d &relative_R, Vector3d &relative_T, int &l)
- void **slideWindow** ()
- void **solveOdometry** ()
- void **slideWindowNew** ()
- void **slideWindowOld** ()
- void **optimization** ()
- void **vector2double** ()
- void **double2vector** ()
- bool **failureDetection** ()

Public Attributes

- SolverFlag **solver_flag**
- MarginalizationFlag **marginalization_flag**
- Vector3d **g**
- MatrixXd **Ap** [2]
- MatrixXd **backup_A**
- VectorXd **bp** [2]
- VectorXd **backup_b**
- Matrix3d **ric** [NUM_OF_CAM]
- Vector3d **tic** [NUM_OF_CAM]
- Vector3d **Ps** [(WINDOW_SIZE+1)]

- Vector3d **Vs** [(WINDOW_SIZE+1)]
- Matrix3d **Rs** [(WINDOW_SIZE+1)]
- Vector3d **Bas** [(WINDOW_SIZE+1)]
- Vector3d **Bgs** [(WINDOW_SIZE+1)]
- double **td**
- Matrix3d **back_R0**
- Matrix3d **last_R**
- Matrix3d **last_R0**
- Vector3d **back_P0**
- Vector3d **last_P**
- Vector3d **last_P0**
- std_msgs::Header **Headers** [(WINDOW_SIZE+1)]
- IntegrationBase ∗ **pre_integrations** [(WINDOW_SIZE+1)]
- Vector3d **acc_0**
- Vector3d **gyr_0**
- vector< double > **dt_buf** [(WINDOW_SIZE+1)]
- vector< Vector3d > **linear_acceleration_buf** [(WINDOW_SIZE+1)]
- vector< Vector3d > **angular_velocity_buf** [(WINDOW_SIZE+1)]
- int **frame_count**
- int **sum_of_outlier**
- int **sum_of_back**
- int **sum_of_front**
- int **sum_of_invalid**
- FeatureManager **f_manager**
- MotionEstimator **m_estimator**
- InitialEXRotation **initial_ex_rotation**
- bool **first_imu**
- bool **is_valid**
- bool **is_key**
- bool **failure_occur**
- vector< Vector3d > **point_cloud**
- vector< Vector3d > **margin_cloud**
- vector< Vector3d > **key_poses**
- double **initial_timestamp**
- double **para_Pose** [WINDOW_SIZE+1][SIZE_POSE]
- double **para_SpeedBias** [WINDOW_SIZE+1][SIZE_SPEEDBIAS]
- double **para_Feature** [NUM_OF_F][SIZE_FEATURE]
- double **para_Ex_Pose** [NUM_OF_CAM][SIZE_POSE]
- double **para_Retrive_Pose** [SIZE_POSE]
- double **para_Td** [1][1]
- double **para_Tr** [1][1]
- int **loop_window_index**
- MarginalizationInfo ∗ **last_marginalization_info**
- vector< double ∗ > **last_marginalization_parameter_blocks**
- map< double, ImageFrame > **all_image_frame**
- IntegrationBase ∗ **tmp_pre_integration**
- bool **relocalization_info**
- double **relo_frame_stamp**
- double **relo_frame_index**
- int **relo_frame_local_index**
- vector< Vector3d > **match_points**
- double **relo_Pose** [SIZE_POSE]
- Matrix3d **drift_correct_r**
- Vector3d **drift_correct_t**
- Vector3d **prev_relo_t**

- Matrix3d **prev_relo_r**
- Vector3d **relo_relative_t**
- Quaterniond **relo_relative_q**
- double **relo_relative_yaw**

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

- vins_estimator/src/estimator.h
- vins_estimator/src/estimator.cpp

5.14 FeatureManager Class Reference

Public Member Functions

- **FeatureManager** (Matrix3d _Rs[ ])
- void **setRic** (Matrix3d _ric[ ])
- void **clearState** ()
- int **getFeatureCount** ()
- bool **addFeatureCheckParallax** (int frame_count, const map< int, vector< pair< int, Eigen::Matrix< double, 7, 1 >>>> &image, double td)
- void **debugShow** ()
- vector< pair< Vector3d, Vector3d > > **getCorresponding** (int frame_count_l, int frame_count_r)
- void **setDepth** (const VectorXd &x)
- void **removeFailures** ()
- void **clearDepth** (const VectorXd &x)
- VectorXd **getDepthVector** ()
- void **triangulate** (Vector3d Ps[ ], Vector3d tic[ ], Matrix3d ric[ ])
- void **removeBackShiftDepth** (Eigen::Matrix3d marg_R, Eigen::Vector3d marg_P, Eigen::Matrix3d new_R, Eigen::Vector3d new_P)
- void **removeBack** ()
- void **removeFront** (int frame_count)
- void **removeOutlier** ()

Public Attributes

- list< FeaturePerId > **feature**
- int **last_track_num**

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

- vins_estimator/src/feature_manager.h
- vins_estimator/src/feature_manager.cpp

5.15 FeaturePerFrame Class Reference

Public Member Functions

- **FeaturePerFrame** (const Eigen::Matrix< double, 7, 1 > &_point, double td)

Public Attributes

- double **cur_td**
- Vector3d **point**
- Vector2d **uv**
- Vector2d **velocity**
- double **z**
- bool **is_used**
- double **parallax**
- MatrixXd **A**
- VectorXd **b**
- double **dep_gradient**

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

- vins_estimator/src/feature_manager.h

5.16 FeaturePerId Class Reference

Public Member Functions

- **FeaturePerId** (int _feature_id, int _start_frame)
- int **endFrame** ()

Public Attributes

- const int **feature_id**
- int **start_frame**
- vector< FeaturePerFrame > **feature_per_frame**
- int **used_num**
- bool **is_outlier**
- bool **is_margin**
- double **estimated_depth**
- int **solve_flag**
- Vector3d **gt_p**

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

- vins_estimator/src/feature_manager.h
- vins_estimator/src/feature_manager.cpp

5.17 FileSystemHelper Class Reference

Static Public Member Functions

- static int **createDirectoryIfNotExists** (const char ∗path)

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

- vins_estimator/src/utility/utility.h

5.18 GlobalSFM Class Reference

Public Member Functions

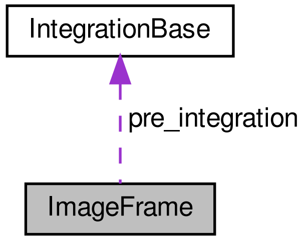
- bool **construct** (int frame_num, Quaterniond ∗q, Vector3d ∗T, int l, const Matrix3d relative_R, const Vector3d relative_T, vector< SFMFeature > &sfm_f, map< int, Vector3d > &sfm_tracked_points)

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

- vins_estimator/src/initial/initial_sfm.h
- vins_estimator/src/initial/initial_sfm.cpp

5.19 ImageFrame Class Reference

Collaboration diagram for ImageFrame:



Public Member Functions

- **ImageFrame** (const map< int, vector< pair< int, Eigen::Matrix< double, 7, 1 >>>> &_points, double _t)

Public Attributes

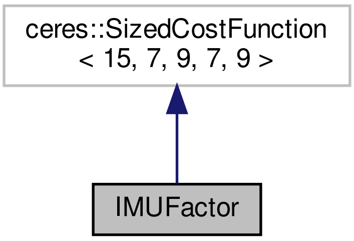
- map< int, vector< pair< int, Eigen::Matrix< double, 7, 1 > > > > **points**
- double **t**
- Matrix3d **R**
- Vector3d **T**
- IntegrationBase ∗ **pre_integration**
- bool **is_key_frame**

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

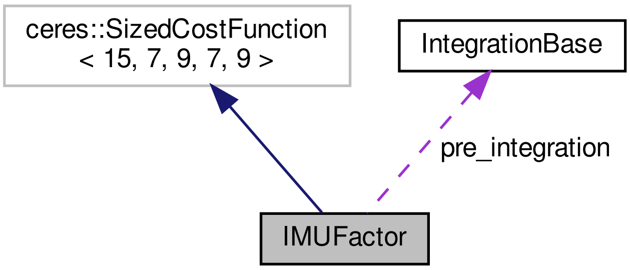
- vins_estimator/src/initial/initial_alignment.h

5.20 IMUFactor Class Reference

Inheritance diagram for IMUFactor:



Collaboration diagram for IMUFactor:



Public Member Functions

- **IMUFactor** (IntegrationBase ∗_pre_integration)
- virtual bool Evaluate (double const ∗const ∗parameters, double ∗residuals, double ∗∗jacobians) const

Public Attributes

- IntegrationBase ∗ **pre_integration**

5.20.1 Member Function Documentation

5.20.1.1 Evaluate()

```
virtual bool IMUFactor::Evaluate (
            double const *const * parameters,
            double * residuals,
            double ** jacobians ) const  [inline], [virtual]
```

ROS_BREAK();

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

- vins_estimator/src/factor/imu_factor.h

5.21 InitialEXRotation Class Reference

Public Member Functions

- bool **CalibrationExRotation** (vector< pair< Vector3d, Vector3d >> corres, Quaterniond delta_q_imu, Matrix3d &calib_ric_result)

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

- vins_estimator/src/initial/initial_ex_rotation.h
- vins_estimator/src/initial/initial_ex_rotation.cpp

5.22 IntegrationBase Class Reference

Public Member Functions

- **IntegrationBase** (const Eigen::Vector3d &_acc_0, const Eigen::Vector3d &_gyr_0, const Eigen::Vector3d &_linearized_ba, const Eigen::Vector3d &_linearized_bg)
- void **push_back** (double dt, const Eigen::Vector3d &acc, const Eigen::Vector3d &gyr)
- void **repropagate** (const Eigen::Vector3d &_linearized_ba, const Eigen::Vector3d &_linearized_bg)
- void **midPointIntegration** (double _dt, const Eigen::Vector3d &_acc_0, const Eigen::Vector3d &_gyr↩ _0, const Eigen::Vector3d &_acc_1, const Eigen::Vector3d &_gyr_1, const Eigen::Vector3d &delta_↩ p, const Eigen::Quaterniond &delta_q, const Eigen::Vector3d &delta_v, const Eigen::Vector3d &linearized↩ _ba, const Eigen::Vector3d &linearized_bg, Eigen::Vector3d &result_delta_p, Eigen::Quaterniond &result_↩ delta_q, Eigen::Vector3d &result_delta_v, Eigen::Vector3d &result_linearized_ba, Eigen::Vector3d &result↩ _linearized_bg, bool update_jacobian)
- void **propagate** (double _dt, const Eigen::Vector3d &_acc_1, const Eigen::Vector3d &_gyr_1)
- Eigen::Matrix< double, 15, 1 > **evaluate** (const Eigen::Vector3d &Pi, const Eigen::Quaterniond &Qi, const Eigen::Vector3d &Vi, const Eigen::Vector3d &Bai, const Eigen::Vector3d &Bgi, const Eigen::Vector3d &Pj, const Eigen::Quaterniond &Qj, const Eigen::Vector3d &Vj, const Eigen::Vector3d &Baj, const Eigen::Vector3d &Bgj)

Public Attributes

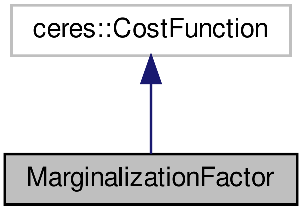
- double **dt**
- Eigen::Vector3d **acc_0**
- Eigen::Vector3d **gyr_0**
- Eigen::Vector3d **acc_1**
- Eigen::Vector3d **gyr_1**
- const Eigen::Vector3d **linearized_acc**
- const Eigen::Vector3d **linearized_gyr**
- Eigen::Vector3d **linearized_ba**
- Eigen::Vector3d **linearized_bg**
- Eigen::Matrix< double, 15, 15 > **jacobian**
- Eigen::Matrix< double, 15, 15 > **covariance**
- Eigen::Matrix< double, 15, 15 > **step_jacobian**
- Eigen::Matrix< double, 15, 18 > **step_V**
- Eigen::Matrix< double, 18, 18 > **noise**
- double **sum_dt**
- Eigen::Vector3d **delta_p**
- Eigen::Quaterniond **delta_q**
- Eigen::Vector3d **delta_v**
- std::vector< double > **dt_buf**
- std::vector< Eigen::Vector3d > **acc_buf**
- std::vector< Eigen::Vector3d > **gyr_buf**

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

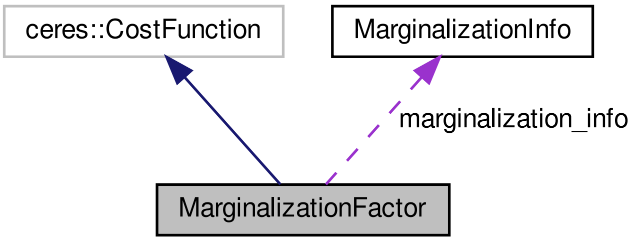
- vins_estimator/src/factor/integration_base.h

5.23 MarginalizationFactor Class Reference

Inheritance diagram for MarginalizationFactor:



Collaboration diagram for MarginalizationFactor:



Public Member Functions

- **MarginalizationFactor** (MarginalizationInfo ∗_marginalization_info)
- virtual bool **Evaluate** (double const ∗const ∗parameters, double ∗residuals, double ∗∗jacobians) const

Public Attributes

- MarginalizationInfo ∗ **marginalization_info**

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

- vins_estimator/src/factor/marginalization_factor.h
- vins_estimator/src/factor/marginalization_factor.cpp

5.24 MarginalizationInfo Class Reference

Public Member Functions

- int **localSize** (int size) const
- int **globalSize** (int size) const
- void **addResidualBlockInfo** (ResidualBlockInfo ∗residual_block_info)
- void **preMarginalize** ()
- void **marginalize** ()
- std::vector< double ∗ > **getParameterBlocks** (std::unordered_map< long, double ∗> &addr_shift)

Public Attributes

- std::vector< ResidualBlockInfo * > **factors**
- int **m**
- int **n**
- std::unordered_map< long, int > **parameter_block_size**
- int **sum_block_size**
- std::unordered_map< long, int > **parameter_block_idx**
- std::unordered_map< long, double * > **parameter_block_data**
- std::vector< int > **keep_block_size**
- std::vector< int > **keep_block_idx**
- std::vector< double * > **keep_block_data**
- Eigen::MatrixXd **linearized_jacobians**
- Eigen::VectorXd **linearized_residuals**
- const double **eps** = 1e-8

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

- vins_estimator/src/factor/marginalization_factor.h
- vins_estimator/src/factor/marginalization_factor.cpp

5.25 MotionEstimator Class Reference

Public Member Functions

- bool **solveRelativeRT** (const vector< pair< Vector3d, Vector3d >> &corres, Matrix3d &R, Vector3d &T)

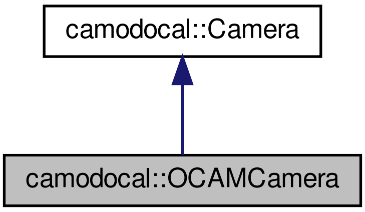
The documentation for this class was generated from the following files:

- vins_estimator/src/initial/solve_5pts.h
- vins_estimator/src/initial/solve_5pts.cpp

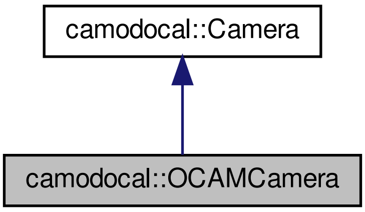
5.26 camodocal::OCAMCamera Class Reference

```
#include <ScaramuzzaCamera.h>
```

Inheritance diagram for camodocal::OCAMCamera:



Collaboration diagram for camodocal::OCAMCamera:



Classes

- class Parameters

Public Member Functions

- OCAMCamera (const Parameters &params)

  Constructor from the projection model parameters.
- Camera::ModelType modelType (void) const

  virtual type of function modelType
- const std::string & cameraName (void) const

  virtual type of funtion cameraName
- int imageWidth (void) const

  virtual type of function imageWidth
- int imageHeight (void) const

  virtual type of function imageHeight
- void estimateIntrinsics (const cv::Size &boardSize, const std::vector< std::vector< cv::Point3f > > &objectPoints, const std::vector< std::vector< cv::Point2f > > &imagePoints)

  virtual function of camera intrinsics
- void liftSphere (const Eigen::Vector2d &p, Eigen::Vector3d &P) const

  Lifts a point from the image plane to the unit sphere.
- void liftProjective (const Eigen::Vector2d &p, Eigen::Vector3d &P) const

  Lifts a point from the image plane to its projective ray.
- void spaceToPlane (const Eigen::Vector3d &P, Eigen::Vector2d &p) const

  Project a 3D point (x,y,z) to the image plane in (u,v)
- void undistToPlane (const Eigen::Vector2d &p_u, Eigen::Vector2d &p) const

  Projects an undistorted 2D point p_u to the image plane.
- void **initUndistortMap** (cv::Mat &map1, cv::Mat &map2, double fScale=1.0) const
- cv::Mat **initUndistortRectifyMap** (cv::Mat &map1, cv::Mat &map2, float fx=-1.0f, float fy=-1.0f, cv::Size imageSize=cv::Size(0, 0), float cx=-1.0f, float cy=-1.0f, cv::Mat rmat=cv::Mat::eye(3, 3, CV_32F)) const
- int parameterCount (void) const

  pure virtual function of parameter count
- const Parameters & **getParameters** (void) const
- void **setParameters** (const Parameters &parameters)
- void readParameters (const std::vector< double > &parameterVec)

pure virtual function of reading parameters

- void writeParameters (std::vector< double > &parameterVec) const

  pure virtual function of writing parameters

- void writeParametersToYamlFile (const std::string &filename) const

  pure virtual function of writing parameters to YAML file

- std::string parametersToString (void) const

  pure virtual of converting parameters to string

Static Public Member Functions

- template<typename T >
  static void **spaceToPlane** (const T ∗const params, const T ∗const q, const T ∗const t, const Eigen::Matrix< T, 3, 1 > &P, Eigen::Matrix< T, 2, 1 > &p)
- template<typename T >
  static void **spaceToSphere** (const T ∗const params, const T ∗const q, const T ∗const t, const Eigen::Matrix< T, 3, 1 > &P, Eigen::Matrix< T, 3, 1 > &P_s)
- template<typename T >
  static void **LiftToSphere** (const T ∗const params, const Eigen::Matrix< T, 2, 1 > &p, Eigen::Matrix< T, 3, 1 > &P)
- template<typename T >
  static void **SphereToPlane** (const T ∗const params, const Eigen::Matrix< T, 3, 1 > &P, Eigen::Matrix< T, 2, 1 > &p)

Additional Inherited Members

5.26.1 Detailed Description

Scaramuzza Camera (Omnidirectional) https://sites.google.com/site/scarabotix/ocamcalib-toolbox

5.26.2 Member Function Documentation

5.26.2.1 liftProjective()

```
void camodocal::OCAMCamera::liftProjective (
             const Eigen::Vector2d & p,
             Eigen::Vector3d & P ) const  [virtual]
```

Lifts a point from the image plane to its projective ray.

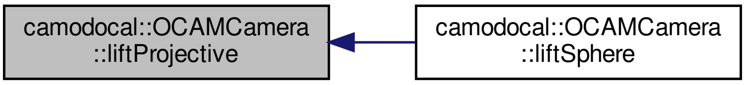
Parameters

p	image coordinates
P	coordinates of the projective ray

Implements camodocal::Camera.

Referenced by liftSphere().

Here is the caller graph for this function:



5.26.2.2 liftSphere()

```
void camodocal::OCAMCamera::liftSphere (
             const Eigen::Vector2d & p,
             Eigen::Vector3d & P ) const  [virtual]
```

Lifts a point from the image plane to the unit sphere.

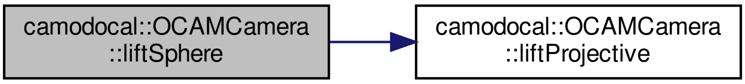
Parameters

p	image coordinates
P	coordinates of the point on the sphere

Implements camodocal::Camera.

References liftProjective().

Here is the call graph for this function:



5.26.2.3 spaceToPlane()

```
void camodocal::OCAMCamera::spaceToPlane (
             const Eigen::Vector3d & P,
             Eigen::Vector2d & p ) const  [virtual]
```

Project a 3D point (*x*,*y*,*z*) to the image plane in (*u*,*v*)

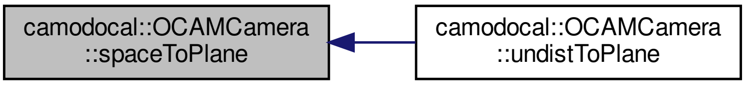
Parameters

P	3D point coordinates
p	return value, contains the image point coordinates

Implements camodocal::Camera.

Referenced by undistToPlane().

Here is the caller graph for this function:


camodocal::OCAMCamera ::spaceToPlane
camodocal::OCAMCamera ::undistToPlane

5.26.2.4 undistToPlane()

```
void camodocal::OCAMCamera::undistToPlane (
            const Eigen::Vector2d & p_u,
            Eigen::Vector2d & p ) const  [virtual]
```

Projects an undistorted 2D point p_u to the image plane.

Parameters

p↩ _u	2D point coordinates

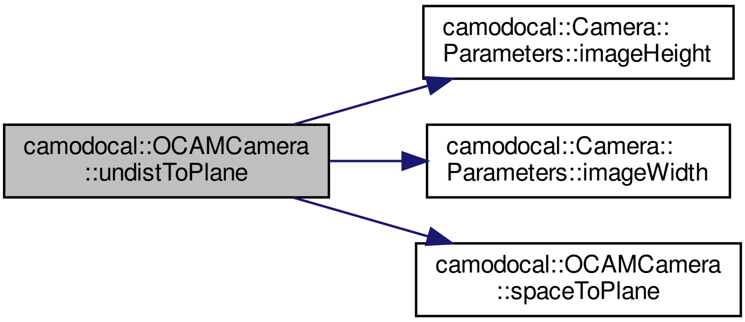
Returns

image point coordinates

Implements camodocal::Camera.

References camodocal::Camera::Parameters::imageHeight(), camodocal::Camera::Parameters::imageWidth(), and spaceToPlane().

Here is the call graph for this function:

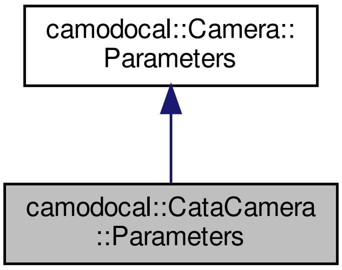


The documentation for this class was generated from the following files:

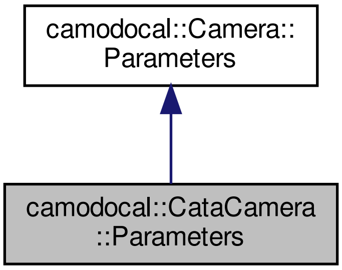
- camera_model/include/camodocal/camera_models/ScaramuzzaCamera.h
- camera_model/src/camera_models/ScaramuzzaCamera.cc

5.27 camodocal::CataCamera::Parameters Class Reference

Inheritance diagram for camodocal::CataCamera::Parameters:



Collaboration diagram for camodocal::CataCamera::Parameters:



Public Member Functions

- **Parameters** (const std::string &cameraName, int w, int h, double xi, double k1, double k2, double p1, double p2, double gamma1, double gamma2, double u0, double v0)
- double & **xi** (void)
- double & **k1** (void)
- double & **k2** (void)
- double & **p1** (void)
- double & **p2** (void)
- double & **gamma1** (void)
- double & **gamma2** (void)
- double & **u0** (void)
- double & **v0** (void)
- double **xi** (void) const
- double **k1** (void) const
- double **k2** (void) const
- double **p1** (void) const
- double **p2** (void) const
- double **gamma1** (void) const
- double **gamma2** (void) const
- double **u0** (void) const
- double **v0** (void) const
- bool readFromYamlFile (const std::string &filename)

  read parameters from YAML or XML file, (pure virtual function which must be override in son classes, following virtual function with '=0' are all pure virtual function)
- void writeToYamlFile (const std::string &filename) const

  write parameters to YAML or XML file;
- Parameters & **operator=** (const Parameters &other)

Friends

- std::ostream & **operator**<< (std::ostream &out, const Parameters &params)

Additional Inherited Members

5.27.1 Member Function Documentation

5.27.1.1 readFromYamlFile()

```
bool camodocal::CataCamera::Parameters::readFromYamlFile (
            const std::string & filename )  [virtual]
```

read parameters from YAML or XML file, (pure virtual function which must be override in son classes, following virtual function with '=0' are all pure virtual function)

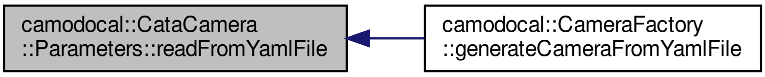
Parameters

filename	filename of YAML or XML file

Implements camodocal::Camera::Parameters.

Referenced by camodocal::CameraFactory::generateCameraFromYamlFile().

Here is the caller graph for this function:



5.27.1.2 writeToYamlFile()

```
void camodocal::CataCamera::Parameters::writeToYamlFile (
            const std::string & filename ) const  [virtual]
```

write parameters to YAML or XML file;

Parameters

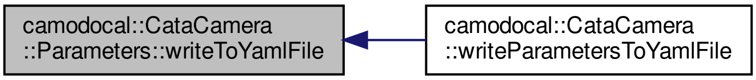
filename	filename of YAML of XML file

Implements camodocal::Camera::Parameters.

References camodocal::Camera::Parameters::m_cameraName, camodocal::Camera::Parameters::m_imageHeight, and camodocal::Camera::Parameters::m_imageWidth.

Referenced by camodocal::CataCamera::writeParametersToYamlFile().

Here is the caller graph for this function:

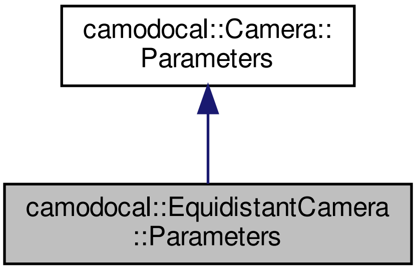


The documentation for this class was generated from the following files:

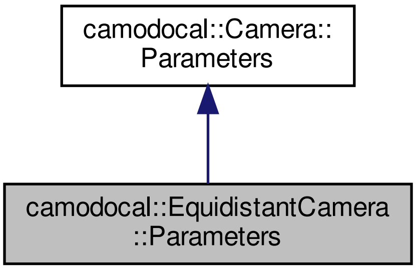
- camera_model/include/camodocal/camera_models/CataCamera.h
- camera_model/src/camera_models/CataCamera.cc

5.28 camodocal::EquidistantCamera::Parameters Class Reference

Inheritance diagram for camodocal::EquidistantCamera::Parameters:



Collaboration diagram for camodocal::EquidistantCamera::Parameters:



Public Member Functions

- **Parameters** (const std::string &cameraName, int w, int h, double k2, double k3, double k4, double k5, double mu, double mv, double u0, double v0)
- double & **k2** (void)
- double & **k3** (void)
- double & **k4** (void)
- double & **k5** (void)
- double & **mu** (void)
- double & **mv** (void)
- double & **u0** (void)
- double & **v0** (void)
- double **k2** (void) const
- double **k3** (void) const
- double **k4** (void) const
- double **k5** (void) const
- double **mu** (void) const
- double **mv** (void) const
- double **u0** (void) const
- double **v0** (void) const
- bool readFromYamlFile (const std::string &filename)

  read parameters from YAML or XML file, (pure virtual function which must be override in son classes, following virtual function with '=0' are all pure virtual function)
- void writeToYamlFile (const std::string &filename) const

  write parameters to YAML or XML file;
- Parameters & **operator=** (const Parameters &other)

Friends

- std::ostream & **operator**<< (std::ostream &out, const Parameters &params)

Additional Inherited Members

5.28.1 Member Function Documentation

5.28.1.1 readFromYamlFile()

```
bool camodocal::EquidistantCamera::Parameters::readFromYamlFile (
             const std::string & filename )  [virtual]
```

read parameters from YAML or XML file, (pure virtual function which must be override in son classes, following virtual function with '=0' are all pure virtual function)

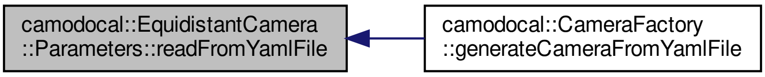
Parameters

filename	filename of YAML or XML file

Implements camodocal::Camera::Parameters.

Referenced by camodocal::CameraFactory::generateCameraFromYamlFile().

Here is the caller graph for this function:


camodocal::EquidistantCamera
::Parameters::readFromYamlFile
camodocal::CameraFactory
::generateCameraFromYamlFile

5.28.1.2 writeToYamlFile()

```
void camodocal::EquidistantCamera::Parameters::writeToYamlFile (
             const std::string & filename ) const  [virtual]
```

write parameters to YAML or XML file;

Parameters

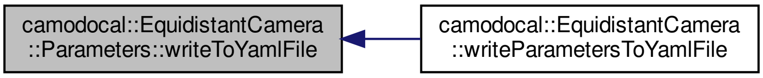
filename	filename of YAML of XML file

Implements camodocal::Camera::Parameters.

References camodocal::Camera::Parameters::m_cameraName, camodocal::Camera::Parameters::m_image↩ Height, and camodocal::Camera::Parameters::m_imageWidth.

Referenced by camodocal::EquidistantCamera::writeParametersToYamlFile().

Here is the caller graph for this function:

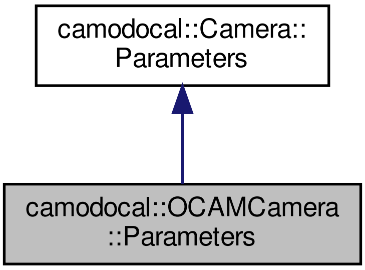

camodocal::EquidistantCamera
::Parameters::writeToYamlFile
camodocal::EquidistantCamera
::writeParametersToYamlFile

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

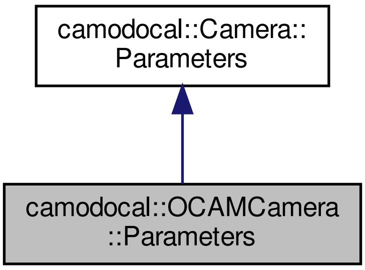
- camera_model/include/camodocal/camera_models/EquidistantCamera.h
- camera_model/src/camera_models/EquidistantCamera.cc

5.29 camodocal::OCAMCamera::Parameters Class Reference

Inheritance diagram for camodocal::OCAMCamera::Parameters:



Collaboration diagram for camodocal::OCAMCamera::Parameters:



Public Member Functions

- double & **C** (void)
- double & **D** (void)
- double & **E** (void)
- double & **center_x** (void)
- double & **center_y** (void)
- double & **poly** (int idx)
- double & **inv_poly** (int idx)
- double **C** (void) const
- double **D** (void) const
- double **E** (void) const
- double **center_x** (void) const
- double **center_y** (void) const
- double **poly** (int idx) const

- double **inv_poly** (int idx) const
- bool readFromYamlFile (const std::string &filename)

  read parameters from YAML or XML file, (pure virtual function which must be override in son classes, following virtual function with '=0' are all pure virtual function)
- void writeToYamlFile (const std::string &filename) const

  write parameters to YAML or XML file;
- Parameters & **operator=** (const Parameters &other)

Friends

- std::ostream & **operator**<< (std::ostream &out, const Parameters &params)

Additional Inherited Members

5.29.1 Member Function Documentation

5.29.1.1 readFromYamlFile()

```
bool camodocal::OCAMCamera::Parameters::readFromYamlFile (
            const std::string & filename )  [virtual]
```

read parameters from YAML or XML file, (pure virtual function which must be override in son classes, following virtual function with '=0' are all pure virtual function)

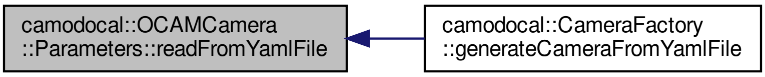
Parameters

filename	filename of YAML or XML file

Implements camodocal::Camera::Parameters.

Referenced by camodocal::CameraFactory::generateCameraFromYamlFile().

Here is the caller graph for this function:


camodocal::OCAMCamera ::Parameters::readFromYamlFile
camodocal::CameraFactory ::generateCameraFromYamlFile

5.29.1.2 writeToYamlFile()

```
void camodocal::OCAMCamera::Parameters::writeToYamlFile (
            const std::string & filename ) const  [virtual]
```

write parameters to YAML or XML file;

Parameters

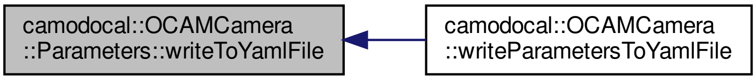
filename	filename of YAML of XML file

Implements camodocal::Camera::Parameters.

References camodocal::Camera::Parameters::m_cameraName, camodocal::Camera::Parameters::m_image↵Height, and camodocal::Camera::Parameters::m_imageWidth.

Referenced by camodocal::OCAMCamera::writeParametersToYamlFile().

Here is the caller graph for this function:

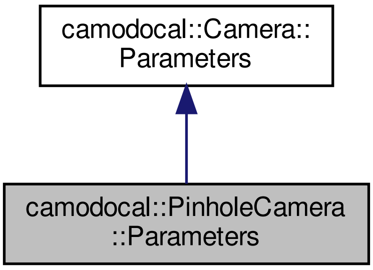

camodocal::OCAMCamera ::Parameters::writeToYamlFile
camodocal::OCAMCamera ::writeParametersToYamlFile

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

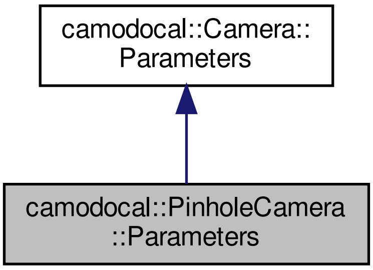
- camera_model/include/camodocal/camera_models/ScaramuzzaCamera.h
- camera_model/src/camera_models/ScaramuzzaCamera.cc

5.30 camodocal::PinholeCamera::Parameters Class Reference

Inheritance diagram for camodocal::PinholeCamera::Parameters:


camodocal::Camera:: Parameters
camodocal::PinholeCamera ::Parameters

Collaboration diagram for camodocal::PinholeCamera::Parameters:



Public Member Functions

- Parameters ()

  Construct a new Pinhole Camera:: Parameters:: Parameters object.
- **Parameters** (const std::string &cameraName, int w, int h, double k1, double k2, double p1, double p2, double fx, double fy, double cx, double cy)
- double & **k1** (void)
- double & **k2** (void)
- double & **p1** (void)
- double & **p2** (void)
- double & **fx** (void)
- double & **fy** (void)
- double & **cx** (void)
- double & **cy** (void)
- double **xi** (void) const
- double **k1** (void) const
- double **k2** (void) const
- double **p1** (void) const
- double **p2** (void) const
- double **fx** (void) const
- double **fy** (void) const
- double **cx** (void) const
- double **cy** (void) const
- bool readFromYamlFile (const std::string &filename)

  read parameters from YAML or XML file, (pure virtual function which must be override in son classes, following virtual function with '=0' are all pure virtual function)
- void writeToYamlFile (const std::string &filename) const

  write parameters to YAML or XML file;
- Parameters & operator= (const Parameters &other)

  override operator "=" to make it work as normal "="

Friends

- std::ostream & operator<< (std::ostream &out, const Parameters &params)

  override operator "<<" to output Parameters with ostream

Additional Inherited Members

5.30.1 Member Function Documentation

5.30.1.1 operator=()

```
PinholeCamera::Parameters & camodocal::PinholeCamera::Parameters::operator= (
            const Parameters & other )
```

override operator "=" to make it work as normal "="

Parameters

other	other parameters object

Returns

: other parameters object

References camodocal::Camera::Parameters::m_cameraName, camodocal::Camera::Parameters::m_image↩ Height, and camodocal::Camera::Parameters::m_imageWidth.

5.30.1.2 readFromYamlFile()

```
bool camodocal::PinholeCamera::Parameters::readFromYamlFile (
            const std::string & filename )  [virtual]
```

read parameters from YAML or XML file, (pure virtual function which must be override in son classes, following virtual function with '=0' are all pure virtual function)

Parameters

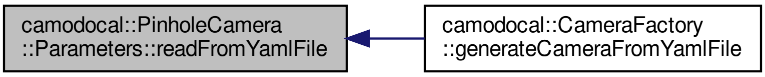
filename	filename of YAML or XML file

Implements camodocal::Camera::Parameters.

References camodocal::Camera::Parameters::m_cameraName, camodocal::Camera::Parameters::m_image↩ Height, and camodocal::Camera::Parameters::m_imageWidth.

Referenced by camodocal::CameraFactory::generateCameraFromYamlFile().

Here is the caller graph for this function:



5.30.1.3 writeToYamlFile()

```
void camodocal::PinholeCamera::Parameters::writeToYamlFile (
             const std::string & filename ) const  [virtual]
```

write parameters to YAML or XML file;

Parameters

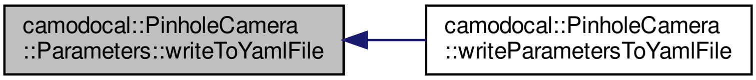
filename	filename of YAML of XML file

Implements camodocal::Camera::Parameters.

References camodocal::Camera::Parameters::m_cameraName, camodocal::Camera::Parameters::m_imageHeight, and camodocal::Camera::Parameters::m_imageWidth.

Referenced by camodocal::PinholeCamera::writeParametersToYamlFile().

Here is the caller graph for this function:



5.30.2 Friends And Related Function Documentation

5.30.2.1 operator<<

```
std::ostream& operator<< (
             std::ostream & out,
             const Parameters & params )  [friend]
```

override operator "<<" to output Parameters with ostream

Parameters

out	
params	

Returns

:

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

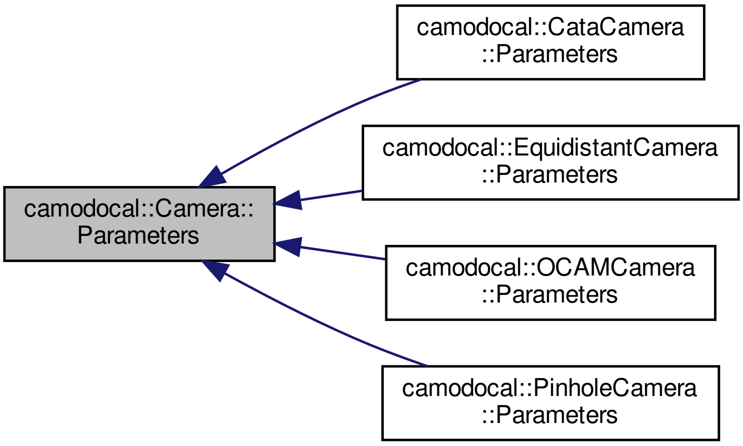
- camera_model/include/camodocal/camera_models/PinholeCamera.h
- camera_model/src/camera_models/PinholeCamera.cc

5.31 camodocal::Camera::Parameters Class Reference

nested class for camera parameters

```
#include <Camera.h>
```

Inheritance diagram for camodocal::Camera::Parameters:


camodocal::CataCamera ::Parameters
camodocal::EquidistantCamera ::Parameters
camodocal::Camera:: Parameters
camodocal::OCAMCamera ::Parameters
camodocal::PinholeCamera ::Parameters

Public Member Functions

- EIGEN_MAKE_ALIGNED_OPERATOR_NEW Parameters (ModelType modelType)

  Construct a new Parameters object.
- Parameters (ModelType modelType, const std::string &cameraName, int w, int h)

  Construct a new Parameters object.
- ModelType & modelType (void)

return modelType of current camera object

- std::string & cameraName (void)

  return name of current camera object
- int & imageWidth (void)

  return image width of current camera object
- int & imageHeight (void)

  return image height of current camera object
- ModelType modelType (void) const

  return modelType of current camera object (const data member)
- const std::string & cameraName (void) const

  return name of current camera object(const data member)
- int imageWidth (void) const

  return image width of current camera object(const data member)
- int imageHeight (void) const

  return image height of current camera object (const data member)
- int nIntrinsics (void) const

  return number of camera intrinsics(const data member)
- virtual bool readFromYamlFile (const std::string &filename)=0

  read parameters from YAML or XML file, (pure virtual function which must be override in son classes, following virtual function with '=0' are all pure virtual function)
- virtual void writeToYamlFile (const std::string &filename) const =0

  write parameters to YAML or XML file;

Protected Attributes

- ModelType **m_modelType**
- int m_nIntrinsics
- std::string m_cameraName

  camera name
- int m_imageWidth

  image width in pixels
- int m_imageHeight

  image height in pixels

5.31.1 Detailed Description

nested class for camera parameters

5.31.2 Constructor & Destructor Documentation

5.31.2.1 Parameters() [1/2]

```
camodocal::Camera::Parameters::Parameters (
            ModelType modelType )
```

Construct a new Parameters object.

Parameters

modelType	model of current camera object

References m_nIntrinsics.

5.31.2.2 Parameters() [2/2]

```
camodocal::Camera::Parameters::Parameters (
            ModelType modelType,
            const std::string & cameraName,
            int w,
            int h )
```

Construct a new Parameters object.

Parameters

modelType	model of current camera object
cameraName	name of current camera object
w	image width of current camera object
h	image height of current camera object

References m_nIntrinsics.

5.31.3 Member Function Documentation

5.31.3.1 cameraName() [1/2]

```
std::string & camodocal::Camera::Parameters::cameraName (
            void )
```

return name of current camera object

Returns

:

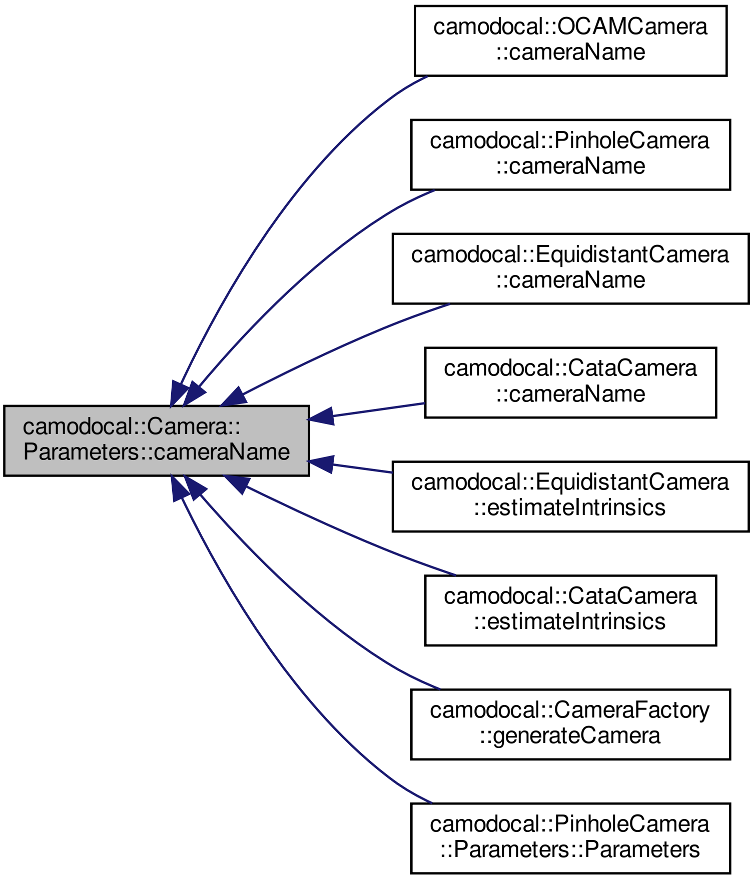
Return values

cameraName	std::string type

References m_cameraName.

Referenced by camodocal::OCAMCamera::cameraName(), camodocal::PinholeCamera::cameraName(), camodocal↩ ::EquidistantCamera::cameraName(), camodocal::CataCamera::cameraName(), camodocal::EquidistantCamera↩ ::estimateIntrinsics(), camodocal::CataCamera::estimateIntrinsics(), camodocal::CameraFactory::generate↩ Camera(), and camodocal::PinholeCamera::Parameters::Parameters().

Here is the caller graph for this function:


camodocal::OCAMCamera ::cameraName
camodocal::PinholeCamera ::cameraName
camodocal::EquidistantCamera ::cameraName
camodocal::CataCamera ::cameraName
camodocal::Camera:: Parameters::cameraName
camodocal::EquidistantCamera ::estimateIntrinsics
camodocal::CataCamera ::estimateIntrinsics
camodocal::CameraFactory ::generateCamera
camodocal::PinholeCamera ::Parameters::Parameters

5.31.3.2 cameraName() [2/2]

```
const std::string & camodocal::Camera::Parameters::cameraName (
            void ) const
```

return name of current camera object(const data member)

Returns

:

Return values

cameraName	std::string type

References m_cameraName.

5.31.3.3 imageHeight() [1/2]

```
int & camodocal::Camera::Parameters::imageHeight (
            void  )
```

return image height of current camera object

Returns

:

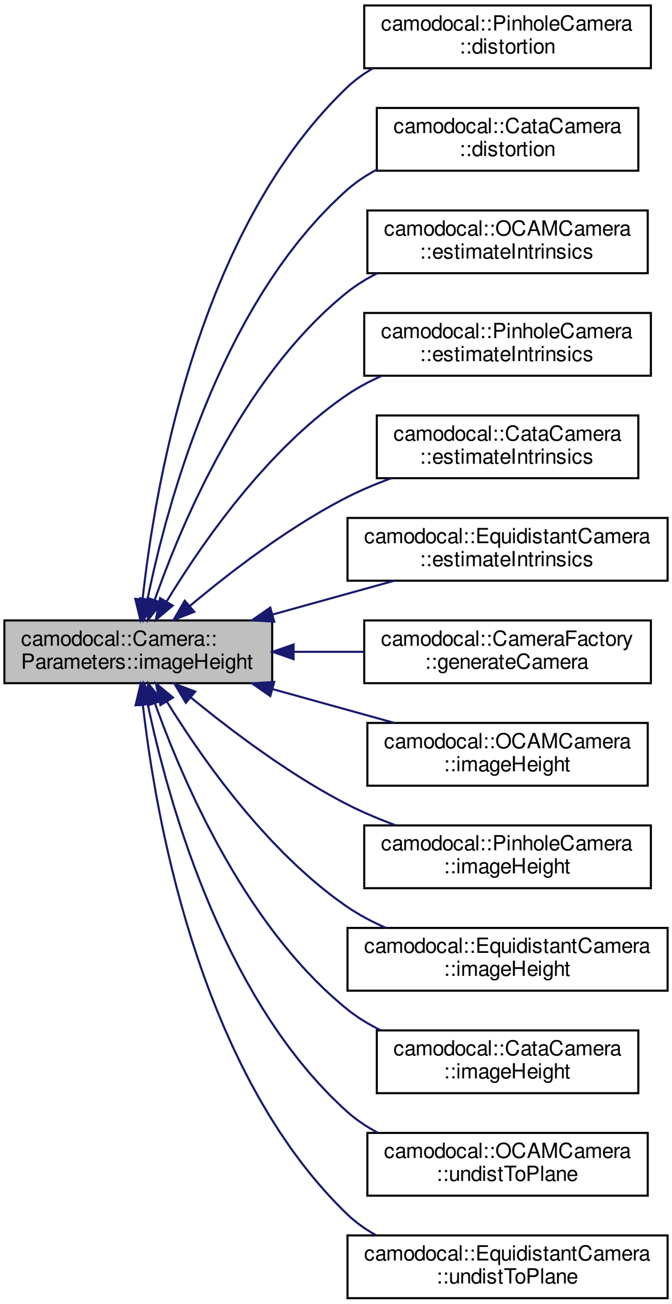
Return values

imageHeight	image height in pixels, int type

References m_imageHeight.

Referenced by camodocal::PinholeCamera::distortion(), camodocal::CataCamera::distortion(), camodocal::OCAMCamera::estimateIntrinsics(), camodocal::PinholeCamera::estimateIntrinsics(), camodocal::CataCamera::estimateIntrinsics(), camodocal::EquidistantCamera::estimateIntrinsics(), camodocal::CameraFactory::generateCamera(), camodocal::OCAMCamera::imageHeight(), camodocal::PinholeCamera::imageHeight(), camodocal::EquidistantCamera::imageHeight(), camodocal::CataCamera::imageHeight(), camodocal::OCAMCamera::undistToPlane(), and camodocal::EquidistantCamera::undistToPlane().

Here is the caller graph for this function:



5.31.3.4 imageHeight() [2/2]

```
int camodocal::Camera::Parameters::imageHeight (
             void ) const
```

return image height of current camera object (const data member)

Returns

:

Return values

imageHeight	image height in pixels, int type

References m_imageHeight.

5.31.3.5 imageWidth() [1/2]

```
int & camodocal::Camera::Parameters::imageWidth (
            void  )
```

return image width of current camera object

Returns

:

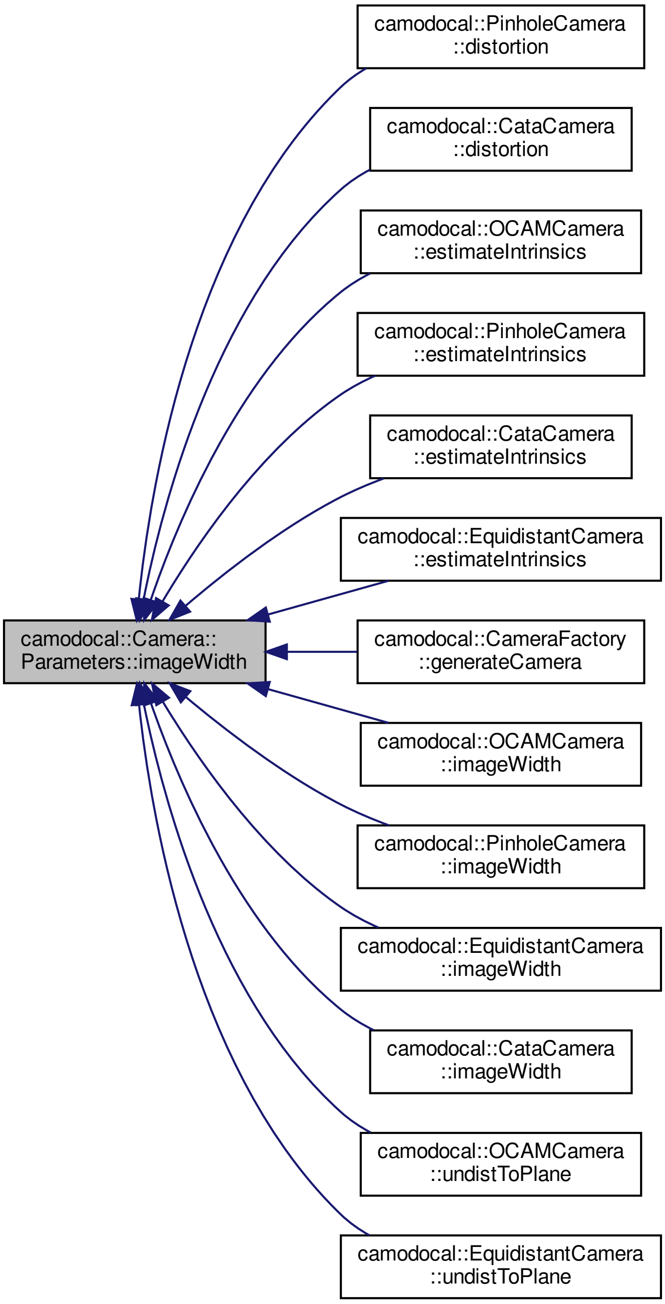
Return values

imageWidth	image width in pixels, int type

References m_imageWidth.

Referenced by camodocal::PinholeCamera::distortion(), camodocal::CataCamera::distortion(), camodocal::OCAMCamera::estimateIntrinsics(), camodocal::PinholeCamera::estimateIntrinsics(), camodocal::CataCamera::estimateIntrinsics(), camodocal::EquidistantCamera::estimateIntrinsics(), camodocal::CameraFactory::generateCamera(), camodocal::OCAMCamera::imageWidth(), camodocal::PinholeCamera::imageWidth(), camodocal::EquidistantCamera::imageWidth(), camodocal::CataCamera::imageWidth(), camodocal::OCAMCamera::undistToPlane(), and camodocal::EquidistantCamera::undistToPlane().

Here is the caller graph for this function:



5.31.3.6 imageWidth() [2/2]

```
int camodocal::Camera::Parameters::imageWidth (
             void  ) const
```

return image width of current camera object(const data member)

Returns

:

Return values

imageWidth	image width in pixels, int type

References m_imageWidth.

5.31.3.7 modelType() [1/2]

```
Camera::ModelType & camodocal::Camera::Parameters::modelType (
            void )
```

return modelType of current camera object

Returns

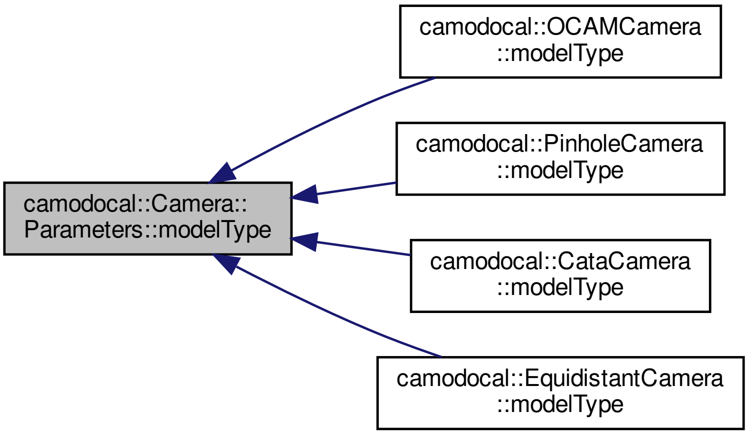
:

Return values

modelType	modelType of current camera object

Referenced by camodocal::OCAMCamera::modelType(), camodocal::PinholeCamera::modelType(), camodocal::↩ CataCamera::modelType(), and camodocal::EquidistantCamera::modelType().

Here is the caller graph for this function:



5.31.3.8 modelType() [2/2]

```
Camera::ModelType camodocal::Camera::Parameters::modelType (
            void ) const
```

return modelType of current camera object (const data member)

Returns

:

Return values

modelType	modelType of current camera object

5.31.3.9 nIntrinsics()

```
int camodocal::Camera::Parameters::nIntrinsics (
            void ) const
```

return number of camera intrinsics(const data member)

Returns

:

Return values

nIntrinsics	number of intrinsics, int type

References m_nIntrinsics.

5.31.3.10 readFromYamlFile()

```
virtual bool camodocal::Camera::Parameters::readFromYamlFile (
            const std::string & filename ) [pure virtual]
```

read parameters from YAML or XML file, (pure virtual function which must be override in son classes, following virtual function with '=0' are all pure virtual function)

Parameters

filename	filename of YAML or XML file

Implemented in camodocal::CataCamera::Parameters, camodocal::OCAMCamera::Parameters, camodocal::↩ EquidistantCamera::Parameters, and camodocal::PinholeCamera::Parameters.

5.31.3.11 writeToYamlFile()

```
virtual void camodocal::Camera::Parameters::writeToYamlFile (
             const std::string & filename ) const  [pure virtual]
```

write parameters to YAML or XML file;

Parameters

filename	filename of YAML of XML file

Implemented in camodocal::CataCamera::Parameters, camodocal::OCAMCamera::Parameters, camodocal::↩ EquidistantCamera::Parameters, and camodocal::PinholeCamera::Parameters.

5.31.4 Member Data Documentation

5.31.4.1 m_nIntrinsics

```
int camodocal::Camera::Parameters::m_nIntrinsics  [protected]
```

number of camera Intrinsics, for pinhole camera model are four distortion parameters: k1, k2, p1, p2 four projection parameters: fx, fy, cx, cy

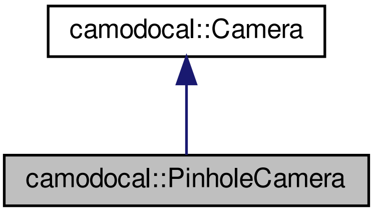
Referenced by nIntrinsics(), and Parameters().

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

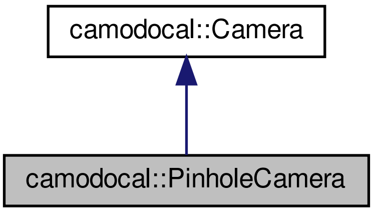
- camera_model/include/camodocal/camera_models/Camera.h
- camera_model/src/camera_models/Camera.cc

5.32 camodocal::PinholeCamera Class Reference

Inheritance diagram for camodocal::PinholeCamera:



Collaboration diagram for camodocal::PinholeCamera:



Classes

- class Parameters

Public Member Functions

- PinholeCamera ()

  Construct a new Pinhole Camera:: Pinhole Camera object.
- PinholeCamera (const std::string &cameraName, int imageWidth, int imageHeight, double k1, double k2, double p1, double p2, double fx, double fy, double cx, double cy)

  Constructor from the projection model parameters.
- PinholeCamera (const Parameters &params)

  Constructor from the projection model parameters.
- Camera::ModelType modelType (void) const

  virtual type of function modelType
- const std::string & cameraName (void) const

  virtual type of funtion cameraName
- int imageWidth (void) const

  virtual type of function imageWidth
- int imageHeight (void) const

  virtual type of function imageHeight
- void estimateIntrinsics (const cv::Size &boardSize, const std::vector< std::vector< cv::Point3f > > &objectPoints, const std::vector< std::vector< cv::Point2f > > &imagePoints)

  virtual function of camera intrinsics
- virtual void liftSphere (const Eigen::Vector2d &p, Eigen::Vector3d &P) const

  Lifts a point from the image plane to the unit sphere.
- void liftProjective (const Eigen::Vector2d &p, Eigen::Vector3d &P) const

  Lifts a point from the image plane to its projective ray.
- void spaceToPlane (const Eigen::Vector3d &P, Eigen::Vector2d &p) const

  Project a 3D point (x,y,z) to the image plane in (u,v)
- void **spaceToPlane** (const Eigen::Vector3d &P, Eigen::Vector2d &p, Eigen::Matrix< double, 2, 3 > &J) const
- void undistToPlane (const Eigen::Vector2d &p_u, Eigen::Vector2d &p) const

  Projects an undistorted 2D point p_u to the image plane.
- void distortion (const Eigen::Vector2d &p_u, Eigen::Vector2d &d_u) const

  Apply distortion to input point (from the normalised plane)

- void distortion (const Eigen::Vector2d &p_u, Eigen::Vector2d &d_u, Eigen::Matrix2d &J) const

    Apply distortion to input point (from the normalised plane) and calculate Jacobian.
- void **initUndistortMap** (cv::Mat &map1, cv::Mat &map2, double fScale=1.0) const
- cv::Mat **initUndistortRectifyMap** (cv::Mat &map1, cv::Mat &map2, float fx=-1.0f, float fy=-1.0f, cv::Size imageSize=cv::Size(0, 0), float cx=-1.0f, float cy=-1.0f, cv::Mat rmat=cv::Mat::eye(3, 3, CV_32F)) const
- int parameterCount (void) const

    pure virtual function of parameter count
- const Parameters & **getParameters** (void) const
- void **setParameters** (const Parameters &parameters)
- void readParameters (const std::vector< double > &parameterVec)

    pure virtual function of reading parameters
- void writeParameters (std::vector< double > &parameterVec) const

    pure virtual function of writing parameters
- void writeParametersToYamlFile (const std::string &filename) const

    pure virtual function of writing parameters to YAML file
- std::string parametersToString (void) const

    pure virtual of converting parameters to string

Static Public Member Functions

- template< typename T >
  static void **spaceToPlane** (const T ∗const params, const T ∗const q, const T ∗const t, const Eigen::Matrix< T, 3, 1 > &P, Eigen::Matrix< T, 2, 1 > &p)

Additional Inherited Members

5.32.1 Constructor & Destructor Documentation

5.32.1.1 PinholeCamera() [1/2]

```
camodocal::PinholeCamera::PinholeCamera (
              const std::string & cameraName,
              int imageWidth,
              int imageHeight,
              double k1,
              double k2,
              double p1,
              double p2,
              double fx,
              double fy,
              double cx,
              double cy )
```

Constructor from the projection model parameters.

Construct a new Pinhole Camera:: Pinhole Camera object.

Parameters

cameraName	name of camera
imageWidth	width of current image
imageHeight	height of current image
k1	distortion parameter
k2	distortion parameter
p1	distortion parameter
p2	distortion parameter
fx	projection parameter
fy	projection parameter
cx	projection parameter
cy	projection parameter

5.32.1.2 PinholeCamera() [2/2]

```
camodocal::PinholeCamera::PinholeCamera (
             const Parameters & params )
```

Constructor from the projection model parameters.

Construct a new Pinhole Camera:: Pinhole Camera object.

Parameters

params	PinholeCamera::Parameters object

5.32.2 Member Function Documentation

5.32.2.1 distortion() [1/2]

```
void camodocal::PinholeCamera::distortion (
             const Eigen::Vector2d & p_u,
             Eigen::Vector2d & d_u ) const
```

Apply distortion to input point (from the normalised plane)

Parameters

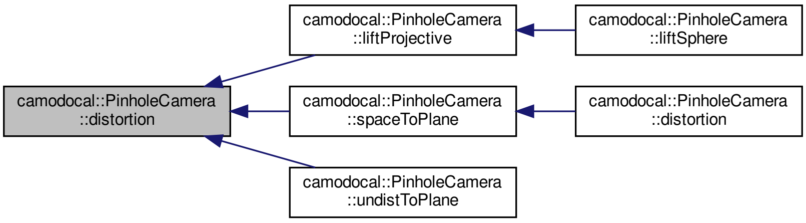
p← _u	undistorted coordinates of point on the normalised plane

Returns

to obtain the distorted point: p_d = p_u + d_u

Referenced by liftProjective(), spaceToPlane(), and undistToPlane().

Here is the caller graph for this function:


camodocal::PinholeCamera ::distortion
camodocal::PinholeCamera ::liftProjective
camodocal::PinholeCamera ::liftSphere
camodocal::PinholeCamera ::spaceToPlane
camodocal::PinholeCamera ::distortion
camodocal::PinholeCamera ::undistToPlane

5.32.2.2 distortion() [2/2]

```
void camodocal::PinholeCamera::distortion (
             const Eigen::Vector2d & p_u,
             Eigen::Vector2d & d_u,
             Eigen::Matrix2d & J ) const
```

Apply distortion to input point (from the normalised plane) and calculate Jacobian.

Parameters

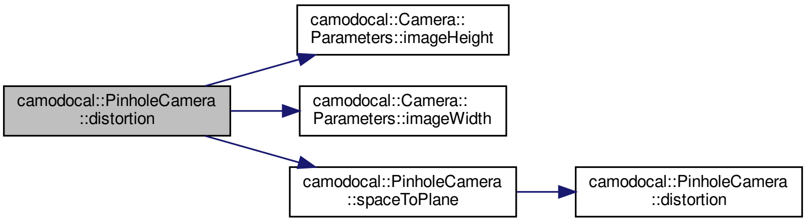
p← _u	undistorted coordinates of point on the normalised plane

Returns

to obtain the distorted point: p_d = p_u + d_u

References camodocal::Camera::Parameters::imageHeight(), camodocal::Camera::Parameters::imageWidth(), and spaceToPlane().

Here is the call graph for this function:


camodocal::PinholeCamera ::distortion
camodocal::Camera:: Parameters::imageHeight
camodocal::Camera:: Parameters::imageWidth
camodocal::PinholeCamera ::spaceToPlane
camodocal::PinholeCamera ::distortion

5.32.2.3 liftProjective()

```
void camodocal::PinholeCamera::liftProjective (
            const Eigen::Vector2d & p,
            Eigen::Vector3d & P ) const  [virtual]
```

Lifts a point from the image plane to its projective ray.

Parameters

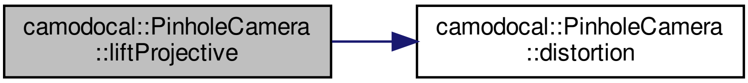
p	image coordinates
P	coordinates of the projective ray

Implements camodocal::Camera.

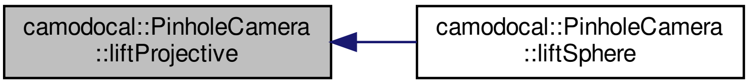
References distortion().

Referenced by liftSphere().

Here is the call graph for this function:


camodocal::PinholeCamera ::liftProjective
camodocal::PinholeCamera ::distortion

Here is the caller graph for this function:



5.32.2.4 liftSphere()

```
void camodocal::PinholeCamera::liftSphere (
             const Eigen::Vector2d & p,
             Eigen::Vector3d & P ) const  [virtual]
```

Lifts a point from the image plane to the unit sphere.

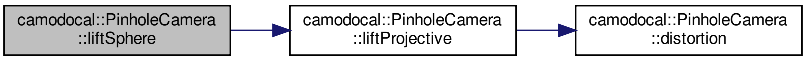
Parameters

p	image coordinates
P	coordinates of the point on the sphere

Implements camodocal::Camera.

References liftProjective().

Here is the call graph for this function:



5.32.2.5 spaceToPlane()

```
void camodocal::PinholeCamera::spaceToPlane (
             const Eigen::Vector3d & P,
             Eigen::Vector2d & p ) const  [virtual]
```

Project a 3D point (*x*,*y*,*z*) to the image plane in (*u*,*v*)

Parameters

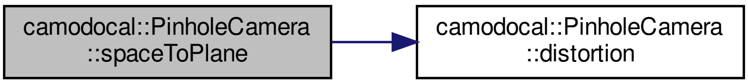
P	3D point coordinates
p	return value, contains the image point coordinates

Implements camodocal::Camera.

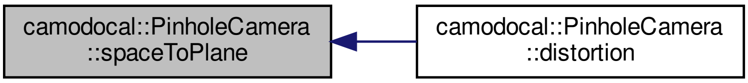
References distortion().

Referenced by distortion().

Here is the call graph for this function:


camodocal::PinholeCamera ::spaceToPlane
camodocal::PinholeCamera ::distortion

Here is the caller graph for this function:


camodocal::PinholeCamera ::spaceToPlane
camodocal::PinholeCamera ::distortion

5.32.2.6 undistToPlane()

```
void camodocal::PinholeCamera::undistToPlane (
            const Eigen::Vector2d & p_u,
            Eigen::Vector2d & p ) const  [virtual]
```

Projects an undistorted 2D point p_u to the image plane.

Parameters

p← *_u*	2D point coordinates

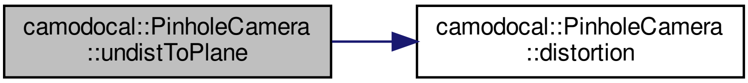
Returns

image point coordinates

Implements camodocal::Camera.

References distortion().

Here is the call graph for this function:

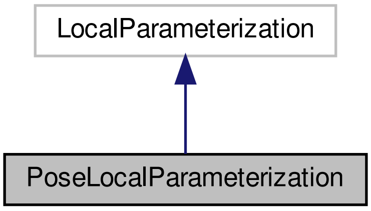

camodocal::PinholeCamera ::undistToPlane
camodocal::PinholeCamera ::distortion

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

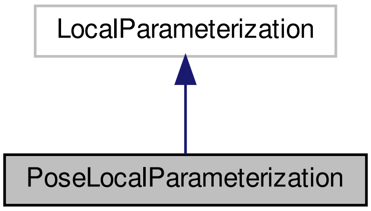
- camera_model/include/camodocal/camera_models/PinholeCamera.h
- camera_model/src/camera_models/PinholeCamera.cc

5.33 PoseLocalParameterization Class Reference

Inheritance diagram for PoseLocalParameterization:



Collaboration diagram for PoseLocalParameterization:

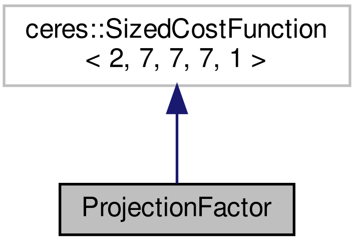


The documentation for this class was generated from the following files:

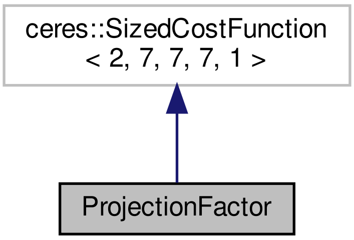
- vins_estimator/src/factor/pose_local_parameterization.h
- vins_estimator/src/factor/pose_local_parameterization.cpp

5.34 ProjectionFactor Class Reference

Inheritance diagram for ProjectionFactor:



Collaboration diagram for ProjectionFactor:



Public Member Functions

- **ProjectionFactor** (const Eigen::Vector3d &_pts_i, const Eigen::Vector3d &_pts_j)
- virtual bool **Evaluate** (double const ∗const ∗parameters, double ∗residuals, double ∗∗jacobians) const
- void **check** (double ∗∗parameters)

Public Attributes

- Eigen::Vector3d **pts_i**
- Eigen::Vector3d **pts_j**
- Eigen::Matrix< double, 2, 3 > **tangent_base**

Static Public Attributes

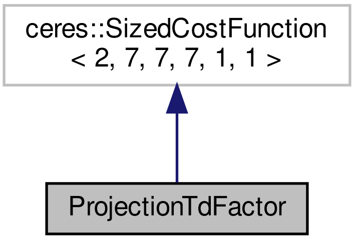
- static Eigen::Matrix2d **sqrt_info**
- static double **sum_t**

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

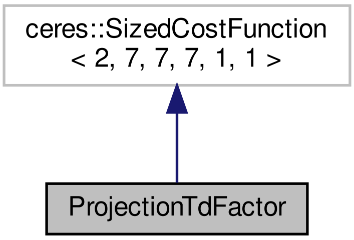
- vins_estimator/src/factor/projection_factor.h
- vins_estimator/src/factor/projection_factor.cpp

5.35 ProjectionTdFactor Class Reference

Inheritance diagram for ProjectionTdFactor:



Collaboration diagram for ProjectionTdFactor:



Public Member Functions

- **ProjectionTdFactor** (const Eigen::Vector3d &_pts_i, const Eigen::Vector3d &_pts_j, const Eigen::Vector2d &_velocity_i, const Eigen::Vector2d &_velocity_j, const double _td_i, const double _td_j, const double _row↩ _i, const double _row_j)
- virtual bool **Evaluate** (double const ∗const ∗parameters, double ∗residuals, double ∗∗jacobians) const
- void **check** (double ∗∗parameters)

Public Attributes

- Eigen::Vector3d **pts_i**
- Eigen::Vector3d **pts_j**
- Eigen::Vector3d **velocity_i**
- Eigen::Vector3d **velocity_j**
- double **td_i**
- double **td_j**
- Eigen::Matrix< double, 2, 3 > **tangent_base**
- double **row_i**
- double **row_j**

Static Public Attributes

- static Eigen::Matrix2d **sqrt_info**
- static double **sum_t**

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

- vins_estimator/src/factor/projection_td_factor.h
- vins_estimator/src/factor/projection_td_factor.cpp

5.36 camodocal::ReprojectionError1< CameraT > Class Template Reference

Public Member Functions

- EIGEN_MAKE_ALIGNED_OPERATOR_NEW **ReprojectionError1** (const Eigen::Vector3d &observed_↩ P, const Eigen::Vector2d &observed_p)
- **ReprojectionError1** (const Eigen::Vector3d &observed_P, const Eigen::Vector2d &observed_p, const Eigen::Matrix2d &sqrtPrecisionMat)
- **ReprojectionError1** (const std::vector< double > &intrinsic_params, const Eigen::Vector3d &observed_P, const Eigen::Vector2d &observed_p)
- template<typename T >
  bool **operator()** (const T ∗const intrinsic_params, const T ∗const q, const T ∗const t, T ∗residuals) const
- template<typename T >
  bool **operator()** (const T ∗const q_cam_odo, const T ∗const t_cam_odo, const T ∗const p_odo, const T ∗const att_odo, T ∗residuals) const

Public Attributes

- std::vector< double > **m_intrinsic_params**
- Eigen::Vector3d **m_observed_P**
- Eigen::Vector2d **m_observed_p**
- Eigen::Matrix2d **m_sqrtPrecisionMat**

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

- camera_model/src/camera_models/CostFunctionFactory.cc

5.37 camodocal::ReprojectionError2< CameraT > Class Template Reference

Public Member Functions

- EIGEN_MAKE_ALIGNED_OPERATOR_NEW **ReprojectionError2** (const std::vector< double > &intrinsic_params, const Eigen::Vector2d &observed_p)
- template<typename T >
  bool **operator()** (const T ∗const q, const T ∗const t, const T ∗const point, T ∗residuals) const

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

- camera_model/src/camera_models/CostFunctionFactory.cc

5.38 camodocal::ReprojectionError3< CameraT > Class Template Reference

Public Member Functions

- EIGEN_MAKE_ALIGNED_OPERATOR_NEW **ReprojectionError3** (const Eigen::Vector2d &observed_p)
- **ReprojectionError3** (const Eigen::Vector2d &observed_p, const Eigen::Matrix2d &sqrtPrecisionMat)
- **ReprojectionError3** (const std::vector< double > &intrinsic_params, const Eigen::Vector2d &observed_p)
- **ReprojectionError3** (const std::vector< double > &intrinsic_params, const Eigen::Vector2d &observed_p, const Eigen::Matrix2d &sqrtPrecisionMat)
- **ReprojectionError3** (const std::vector< double > &intrinsic_params, const Eigen::Vector3d &odo_pos, const Eigen::Vector3d &odo_att, const Eigen::Vector2d &observed_p, bool optimize_cam_odo_z)
- **ReprojectionError3** (const std::vector< double > &intrinsic_params, const Eigen::Quaterniond &cam_↩ odo_q, const Eigen::Vector3d &cam_odo_t, const Eigen::Vector3d &odo_pos, const Eigen::Vector3d &odo↩ _att, const Eigen::Vector2d &observed_p)
- template<typename T >
  bool **operator()** (const T ∗const intrinsic_params, const T ∗const q_cam_odo, const T ∗const t_cam_odo, const T ∗const p_odo, const T ∗const att_odo, const T ∗const point, T ∗residuals) const
- template<typename T >
  bool **operator()** (const T ∗const q_cam_odo, const T ∗const t_cam_odo, const T ∗const point, T ∗residuals) const
- template<typename T >
  bool **operator()** (const T ∗const q_cam_odo, const T ∗const t_cam_odo, const T ∗const p_odo, const T ∗const att_odo, const T ∗const point, T ∗residuals) const
- template<typename T >
  bool **operator()** (const T ∗const point, T ∗residuals) const

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

- camera_model/src/camera_models/CostFunctionFactory.cc

5.39 ReprojectionError3D Struct Reference

Public Member Functions

- **ReprojectionError3D** (double observed_u, double observed_v)
- template<typename T >
  bool **operator()** (const T ∗const camera_R, const T ∗const camera_T, const T ∗point, T ∗residuals) const

Static Public Member Functions

- static ceres::CostFunction ∗ **Create** (const double observed_x, const double observed_y)

Public Attributes

- double **observed_u**
- double **observed_v**

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

- vins_estimator/src/initial/initial_sfm.h

5.40 ResidualBlockInfo Struct Reference

Public Member Functions

- **ResidualBlockInfo** (ceres::CostFunction ∗_cost_function, ceres::LossFunction ∗_loss_function, std↩ ::vector< double ∗> _parameter_blocks, std::vector< int > _drop_set)
- void **Evaluate** ()
- int **localSize** (int size)

Public Attributes

- ceres::CostFunction ∗ **cost_function**
- ceres::LossFunction ∗ **loss_function**
- std::vector< double ∗ > **parameter_blocks**
- std::vector< int > **drop_set**
- double ∗∗ **raw_jacobians**
- std::vector< Eigen::Matrix< double, Eigen::Dynamic, Eigen::Dynamic, Eigen::RowMajor > > **jacobians**
- Eigen::VectorXd **residuals**

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

- vins_estimator/src/factor/marginalization_factor.h
- vins_estimator/src/factor/marginalization_factor.cpp

5.41 SFMFeature Struct Reference

Public Attributes

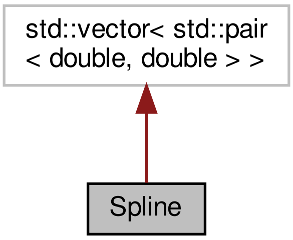
- bool **state**
- int **id**
- vector< pair< int, Vector2d > > **observation**
- double **position** [3]
- double **depth**

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

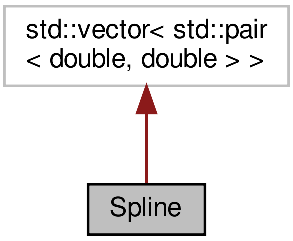
- vins_estimator/src/initial/initial_sfm.h

5.42 Spline Class Reference

Inheritance diagram for Spline:



Collaboration diagram for Spline:



Public Types

- enum **BC_type** { **FIXED_1ST_DERIV_BC**, **FIXED_2ND_DERIV_BC**, **PARABOLIC_RUNOUT_BC** }
- enum **Spline_type** { **LINEAR**, **CUBIC** }
- typedef std::vector< std::pair< double, double > > **base**
- typedef base::const_iterator **const_iterator**

Public Member Functions

- const_iterator **begin** () const
- const_iterator **end** () const
- void **clear** ()
- size_t **size** () const
- size_t **max_size** () const
- size_t **capacity** () const

- bool **empty** () const
- void **addPoint** (double x, double y)
- void **setLowBC** (BC_type BC, double val=0)
- void **setHighBC** (BC_type BC, double val=0)
- void **setType** (Spline_type type)
- double **operator()** (double xval)

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

- camera_model/include/camodocal/chessboard/Spline.h

5.43 camodocal::StereoReprojectionError< CameraT > Class Template Reference

Public Member Functions

- EIGEN_MAKE_ALIGNED_OPERATOR_NEW **StereoReprojectionError** (const Eigen::Vector3d &observed←_P, const Eigen::Vector2d &observed_p_l, const Eigen::Vector2d &observed_p_r)
- template< typename T >
  bool **operator()** (const T ∗const intrinsic_params_l, const T ∗const intrinsic_params_r, const T ∗const q_l, const T ∗const t_l, const T ∗const q_l_r, const T ∗const t_l_r, T ∗residuals) const

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

- camera_model/src/camera_models/CostFunctionFactory.cc

5.44 ThreadsStruct Struct Reference

Public Attributes

- std::vector< ResidualBlockInfo ∗ > **sub_factors**
- Eigen::MatrixXd **A**
- Eigen::VectorXd **b**
- std::unordered_map< long, int > **parameter_block_size**
- std::unordered_map< long, int > **parameter_block_idx**

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

- vins_estimator/src/factor/marginalization_factor.h

5.45 TicToc Class Reference

Public Member Functions

- void **tic** ()
- double **toc** ()

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

- vins_estimator/src/utility/tic_toc.h

5.46 camodocal::Transform Class Reference

Public Member Functions

- **Transform** (const Eigen::Matrix4d &H)
- Eigen::Quaterniond & **rotation** (void)
- const Eigen::Quaterniond & **rotation** (void) const
- double ∗ **rotationData** (void)
- const double ∗const **rotationData** (void) const
- Eigen::Vector3d & **translation** (void)
- const Eigen::Vector3d & **translation** (void) const
- double ∗ **translationData** (void)
- const double ∗const **translationData** (void) const
- Eigen::Matrix4d **toMatrix** (void) const

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

- camera_model/include/camodocal/sparse_graph/Transform.h
- camera_model/src/sparse_graph/Transform.cc

5.47 Utility::uint_< N > Struct Template Reference

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

- vins_estimator/src/utility/utility.h

5.48 Utility Class Reference

Classes

- struct uint_

Public Member Functions

- template<size_t N, typename Lambda , typename IterT >
void **unroller** (const Lambda &f, const IterT &iter, uint_< N >)
- template<typename Lambda , typename IterT >
void **unroller** (const Lambda &f, const IterT &iter, uint_< 0 >)

Static Public Member Functions

- template<typename Derived >
  static Eigen::Quaternion< typename Derived::Scalar > **deltaQ** (const Eigen::MatrixBase< Derived > &theta)
- template<typename Derived >
  static Eigen::Matrix< typename Derived::Scalar, 3, 3 > **skewSymmetric** (const Eigen::MatrixBase< Derived > &q)
- template<typename Derived >
  static Eigen::Quaternion< typename Derived::Scalar > **positify** (const Eigen::QuaternionBase< Derived > &q)
- template<typename Derived >
  static Eigen::Matrix< typename Derived::Scalar, 4, 4 > **Qleft** (const Eigen::QuaternionBase< Derived > &q)
- template<typename Derived >
  static Eigen::Matrix< typename Derived::Scalar, 4, 4 > **Qright** (const Eigen::QuaternionBase< Derived > &p)
- static Eigen::Vector3d **R2ypr** (const Eigen::Matrix3d &R)
- template<typename Derived >
  static Eigen::Matrix< typename Derived::Scalar, 3, 3 > **ypr2R** (const Eigen::MatrixBase< Derived > &ypr)
- static Eigen::Matrix3d **g2R** (const Eigen::Vector3d &g)
- template<typename T >
  static T **normalizeAngle** (const T &angle_degrees)

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

- vins_estimator/src/utility/utility.h
- vins_estimator/src/utility/utility.cpp

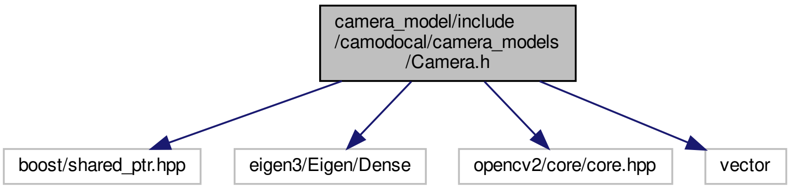
Chapter 6

File Documentation

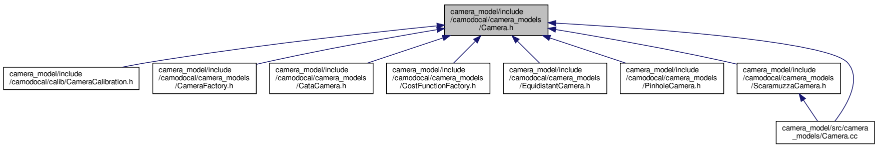
6.1 camera_model/include/camodocal/camera_models/Camera.h File Reference

```
#include <boost/shared_ptr.hpp>
#include <eigen3/Eigen/Dense>
#include <opencv2/core/core.hpp>
#include <vector>
```

Include dependency graph for Camera.h:



This graph shows which files directly or indirectly include this file:



Classes

- class camodocal::Camera

    base class for all camera model

- class camodocal::Camera::Parameters

    nested class for camera parameters

Typedefs

- typedef boost::shared_ptr< Camera > camodocal::CameraPtr

  cameraPtr
- typedef boost::shared_ptr< const Camera > camodocal::CameraConstPtr

  CameraConstPtr.

6.1.1 Detailed Description

Author

yuntianli (yuntianlee91@hotmail.com)

Version

0.1

Date

2019-08-18

Copyright

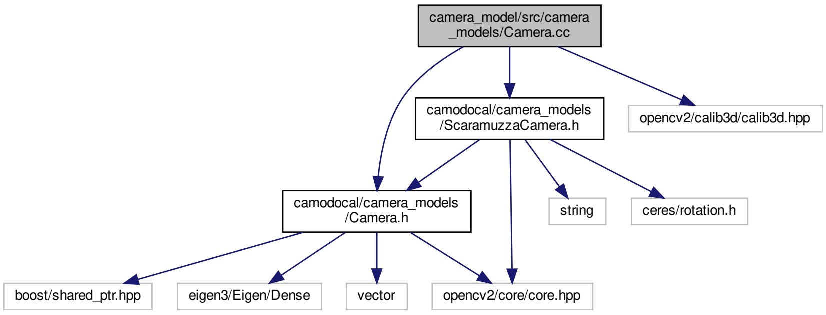
Copyright (c) 2019

6.2 camera_model/src/camera_models/Camera.cc File Reference

base class of all other camera class

```
#include "camodocal/camera_models/Camera.h"
#include "camodocal/camera_models/ScaramuzzaCamera.h"
#include <opencv2/calib3d/calib3d.hpp>
```

Include dependency graph for Camera.cc:



6.2.1 Detailed Description

base class of all other camera class

Author

yuntianli (yuntianlee91@hotmail.com)

Version

0.1

Date

2019-08-18

Copyright

Copyright (c) 2019