INF552

Generated by Doxygen 1.8.13

Contents

Chapter 1

Projet-Inf552

But du projet INPUT => image de gauche + disparité => à partir de deux images, créer le nuage de point 3D => pour chaque pixel, extraire les coordonnées x, y, z correspondantes => en déduire l'altitude des pixels (détecter le sol, objets verticaux, ...)

K deux images matrices rotation, translation Considérer origine Cam1

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Class Documentation

4.1 Plan Class Reference

```
#include <plan.hpp>
```

Public Member Functions

- **Plan** ()
- **Plan** (double a, double b, double c, double d)
- **Plan** (Vec3d p1, Vec3d p2, Vec3d p3)
- double **distance** (Vec3d p)
- void **regression** (**point3dCloud** pointcloud)

Friends

- ostream & **operator**<< (ostream &os, const **Plan** &p)

4.1.1 Constructor & Destructor Documentation

4.1.1.1 Plan() [1/3]

```
Plan::Plan ( )
```

Constructor for the class.

4.1.1.2 Plan() [2/3]

```
Plan::Plan (
             double a,
             double b,
             double c,
             double d )
```

Constructor for the class.

Parameters

a	1st parameter of a∗x + b∗y + c∗z + d = 0
b	2nd parameter of a∗x + b∗y + c∗z + d = 0
c	3rd parameter of a∗x + b∗y + c∗z + d = 0
d	4th parameter of a∗x + b∗y + c∗z + d = 0

4.1.1.3 Plan() [3/3]

```
Plan::Plan (
             Vec3d p1,
             Vec3d p2,
             Vec3d p3 )
```

Constructor for the class.

Parameters

p1	1st point of the plan
p2	2nd point of the plan
p3	3rd point of the plan

4.1.2 Member Function Documentation

4.1.2.1 distance()

```
double Plan::distance (
             Vec3d p )
```

Find the distance between the plan and a point.

Parameters

p	The considered point.

4.1.2.2 regression()

```
void Plan::regression (
              point3dCloud pointcloud )
```

Find the closest plan to the given point cloud.

Parameters

pointcloud	The considered point cloud.

Returns

The **Plan** (p. 7) of linear regression.

4.1.3 Friends And Related Function Documentation

4.1.3.1 operator<<

```
ostream& operator<< (
            ostream & os,
            const  Plan & p )  [friend]
```

Overloads ofstream.

Parameters

os	Considered stream.
p	Considered plan.

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

- **plan.hpp**
- **plan.cpp**

4.2 point3d Class Reference

```
#include <point3d.hpp>
```

Public Member Functions

- **point3d** (Vec3d position, Vec3b color)
- Vec3d **getPosition** ()
- Vec3b **getColor** ()

4.2.1 Constructor & Destructor Documentation

4.2.1.1 point3d()

```
point3d::point3d (
            Vec3d position,
            Vec3b color )
```

Constructor for the class.

Parameters

position	The position of the 3dPoint.
color	The color of the 3dPoint.

4.2.2 Member Function Documentation

4.2.2.1 getColor()

```
Vec3b point3d::getColor ( )
```

4.2.2.2 getPosition()

```
Vec3d point3d::getPosition ( )
```

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

- **point3d.hpp**
- **point3d.cpp**

4.3 point3dCloud Class Reference

```
#include <point3dCloud.hpp>
```

Public Member Functions

- **point3dCloud** ()
- void **push_back** (**point3d** point)
- **point3d** **operator[ ]** (int i)
- int **size** ()

4.3.1 Constructor & Destructor Documentation

4.3.1.1 point3dCloud()

```
point3dCloud::point3dCloud ( )
```

Constructor for the class.

4.3.2 Member Function Documentation

4.3.2.1 operator[]()

```
 point3d point3dCloud::operator[] (
             int i )
```

4.3.2.2 push_back()

```
void point3dCloud::push_back (
              point3d point )
```

4.3.2.3 size()

```
int point3dCloud::size ( )
```

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

- **point3dCloud.hpp**
- **point3dCloud.cpp**

4.4 Ransac Class Reference

```
#include <ransac.hpp>
```

Public Member Functions

- **Ransac** (int n_iterations, double epsilon)
- **point3dCloud fit** (**point3dCloud** pointCloud)

4.4.1 Constructor & Destructor Documentation

4.4.1.1 Ransac()

```
Ransac::Ransac (
            int n_iterations,
            double epsilon )
```

Constructor for the class.

Parameters

n_iterations	The number of iterations for the algorithm.
epsilon	Threshold.

4.4.2 Member Function Documentation

4.4.2.1 fit()

```
point3dCloud Ransac::fit (
            point3dCloud pointCloud )
```

Extract the most correlated points (plane model).

Parameters

pointCloud	The considered point cloud.

Returns

List of points that correlate the most (plane model) as vector<pair<Vec3d, Vec3b>>.

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

- **ransac.hpp**
- **ransac.cpp**

Chapter 5

File Documentation

5.1 plan.cpp File Reference

```
#include "plan.hpp"
```

Functions

- ostream & **operator**<< (ostream &os, const **Plan** &p)

5.1.1 Function Documentation

5.1.1.1 operator<<()

```
ostream& operator<< (
            ostream & os,
            const Plan & p )
```

Overloads ofstream.

Parameters

os	Considered stream.
p	Considered plan.

5.2 plan.hpp File Reference

```
#include <iostream>
#include <opencv2/highgui/highgui.hpp>
```

```
#include <opencv2/features2d/features2d.hpp>
#include <opencv2/calib3d/calib3d.hpp>
#include <opencv2/imgproc.hpp>
#include "point3dCloud.hpp"
```

Classes

- class **Plan**

5.3 point3d.cpp File Reference

```
#include "point3d.hpp"
```

5.4 point3d.hpp File Reference

```
#include <iostream>
#include <windows.h>
#include <fstream>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/features2d/features2d.hpp>
#include <opencv2/calib3d/calib3d.hpp>
#include <opencv2/imgproc.hpp>
```

Classes

- class **point3d**

5.5 point3dCloud.cpp File Reference

```
#include "point3dCloud.hpp"
```

5.6 point3dCloud.hpp File Reference

```
#include <iostream>
#include <windows.h>
#include <fstream>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/features2d/features2d.hpp>
#include <opencv2/calib3d/calib3d.hpp>
#include <opencv2/imgproc.hpp>
#include "point3d.hpp"
```

Classes

- class **point3dCloud**

5.7 projet.cpp File Reference

```
#include <iostream>
#include <windows.h>
#include <fstream>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/features2d/features2d.hpp>
#include <opencv2/calib3d/calib3d.hpp>
#include <opencv2/imgproc.hpp>
#include "include\json.hpp"
#include "ransac.hpp"
```

Typedefs

- using **json** = nlohmann::json

Functions

- void **setcursor** (bool visible, DWORD size)
- bool **hasToBeTreated** (int i, int j, double d, const Mat &left_image)
- void **pointCloud2ply** (**point3dCloud** pointcloud)
- **point3dCloud** **pointCloudFromImages** (Mat &left_image, const Mat &disparity, Matx33d N)
- int **main** ()

5.7.1 Typedef Documentation

5.7.1.1 json

```
using json = nlohmann::json
```

This program intents to reconstruct a 3D scene from two images taken from a car in a street and to detect elements such as the road or the vertical objects

Author

Lucas Broux & Romain Loiseau

5.7.2 Function Documentation

5.7.2.1 hasToBeTreated()

```
bool hasToBeTreated (
            int i,
            int j,
            double d,
            const Mat & left_image )  [inline]
```

Determines if the pixel should be treated.

Parameters

d	The disparity between left/right images.

Returns

Whether the pixel should be considered.

5.7.2.2 main()

```
int main ( )
```

5.7.2.3 pointCloud2ply()

```
void pointCloud2ply (
              point3dCloud pointcloud )
```

Generates .ply file from point cloud values.

Parameters

poincloud	The corresponding point cloud.

5.7.2.4 pointCloudFromImages()

```
 point3dCloud pointCloudFromImages (
             Mat & left_image,
             const Mat & disparity,
             Matx33d N )
```

Generates a 3d point cloud from left image + disparity + transformation matrix. Exports the result as .ply file.

Parameters

left_image	The left image.
disparity	The disparity.
N	The matrix of correspondence : it can transform the disparity into 3d point.

Returns

The point cloud as vector<pair<Vec3d, Vec3b>>.

5.7.2.5 setcursor()

```
void setcursor (
            bool visible,
            DWORD size )
```

Function for hiding/showing cursor : hiding with setcursror(0, 0); reinitialisation with setcursor(1, 10).

Parameters

visible	Whether the cursor should be visible.
size	The size of the cursor.

5.8 ransac.cpp File Reference

```
#include "ransac.hpp"
```

5.9 ransac.hpp File Reference

```
#include <iostream>
#include <windows.h>
#include <fstream>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/features2d/features2d.hpp>
#include <opencv2/calib3d/calib3d.hpp>
#include <opencv2/imgproc.hpp>
#include "plan.hpp"
```

Classes

- class **Ransac**

5.10 README.md File Reference

Index