My Project

Generated by Doxygen 1.7.6.1

Sat May 24 2014 21:42:34

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 DiodeFinder Class Reference

Main DiodeFinder class.

`#include <diodefinder.h>`

Public Member Functions

- DiodeFinder (QLabel *camera_label, QLabel *processed_label, QLabel *output_label)

  DiodeFinder Constructor.
- DiodeFinder ()

  Empty constructor.
- ~DiodeFinder ()

  Destructor.
- bool isOk ()

  Checks whether video capture node is opened successfully.
- void capture ()

  Captures video frame.
- void process ()

  Processing given camera frame.
- void findPoints ()

  Finds points on process_mat and saves them to points vector.
- void findObjects ()

  Finds objects among already found points within the scene.
- void findStableObjects ()

  Finds stable objects (filters found objects on the scene).
- void output ()

  Modifies QPixmap for Qt widget and saves data to node file.

- void enableFIFO (const char *filename)

  Enables FIFO output for visualization.
- Vec3b pickColor (int x, int y)

  Picks color from captured frame.

Public Attributes

- QLabel * camera_label

  QLabel with current camera frame.
- QLabel * processed_label

  QLabel with current processed frame.
- QLabel * output_label

  QLabel with program output (results).
- VideoCapture videoCap

  Video capture class.
- Mat camera_mat

  Matrix for captured frame.
- Mat processed_mat

  Matrix for processed frame.
- Mat output_mat

  Matrix for output (result) frame.
- int hu

  HSL palette's H upper limit.
- int hd

  HSL palette's H down limit.
- int lu

  HSL palette's L upper limit.
- int ld

  HSL palette's L down limit.
- int su

  HSL palette's S upper limit.
- int sd

  HSL palette's S down limit.
- int blur

  Blur filter size.
- int median

  Median filter size.
- int tRatio

  Triangle ratio (1:tRatio).
- float epsilon

  Object finder accuracy (distance ~ +/-epsilon).
- vector< Point > points

  Current points on the scene.

- vector< SceneObject > objects

  Current objects on the scene.
- vector< SceneObject > objectsHistory

  History of recent this->objects vector's content.
- vector< StableObject > stableObjects

  Objects recoginzed as stable on the scene.
- bool saveData

  Whether to save data to data.fifo nod.

Static Public Attributes

- static fstream fifo

  FIFO stream.

2.1.1 Detailed Description

Main DiodeFinder class.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 DiodeFinder::DiodeFinder (QLabel * *camera_label*, QLabel * *processed_label*, QLabel * *output_label*)

DiodeFinder Constructor.

Parameters

camera_pixmap	QPixmap for Qt with current camera frame.
processed_pixmap	QPixmap for Qt with current processed frame.

2.1.3 Member Function Documentation

2.1.3.1 void DiodeFinder::enableFIFO (const char * *filename*)

Enables FIFO output for visualization.

Parameters

filename	FIFO filename.

2.1.3.2 bool DiodeFinder::isOk ( )

Checks whether video capture node is opened successfully.

Returns

True if ok.

2.1.3.3 Vec3b DiodeFinder::pickColor (int *x*, int *y*)

Picks color from captured frame.

Parameters

x	X coord on camera_mat.
y	Y coors on camera_mat.

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

- diodefinder.h
- diodefinder.cpp

2.2 MainWindow Class Reference

Public Slots

- void onStart ()

  For "Start" button being clicked.
- void onSliderChange (int)

  For a slider being moved.
- void onSliderReleased ()

  When slider is released.
- void onClick ()

  For "mouseClickEvent" signal.
- void onVisualizationStart ()

  When "3D" button is clicked.

Signals

- void mouseClickEvent ()

  Signal emitted when user clicks on the main window.

Public Member Functions

- MainWindow (QWidget ∗parent=0)

  Constructor.
- ∼MainWindow ()

  Destructor.
- void closeEvent (QCloseEvent ∗event)

  Closing event.
- void refreshResults ()

  Refreshing DiodeFinder results.

Public Attributes

- bool isClicked

  Flag whether mouse btn has been clicked.
- bool isOpen

  Flag whether the app is opened.
- DiodeFinder df

  DiodeFinderr instance.

Protected Member Functions

- void **mousePressEvent** (QMouseEvent ∗e)
- void **mouseReleaseEvent** (QMouseEvent ∗e)

2.2.1 Member Function Documentation

2.2.1.1 void MainWindow::closeEvent (QCloseEvent ∗ *event*)

Closing event.

Parameters

event	

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

- mainwindow.h
- mainwindow.cpp

2.3 SceneObject Class Reference

The SceneObject class. It's a container for objects found on scene, based on filtered diodes.

`#include <sceneobject.h>`

Public Member Functions

- SceneObject ()

  Empty constructor.
- SceneObject (Point basePoint, Point shortPoint, Point longPoint)

  Constructor.

Static Public Member Functions

- static float distance (Point *p1, Point *p2)

  Calculate distance between points.
- static float calcRotation (SceneObject *obj)

  Calculate SceneObject's rotation.
- static float calcSize (SceneObject *obj)

  Calculate SceneObject's size.
- static float diff (SceneObject *obj1, SceneObject *obj2)

  Calculates difference between given objects.

Public Attributes

- Point basePoint

  Base point of the triangle.
- Point longPoint

  Point near to basePoint.
- Point shortPoint

  Point far from basePoint.
- float rotation

  Calculated rotation.
- float size

  Size of the object based on distance between points.

2.3.1 Detailed Description

The SceneObject class. It's a container for objects found on scene, based on filtered diodes.

2.3.2 Constructor & Destructor Documentation

2.3.2.1 SceneObject::SceneObject (Point *basePoint,* Point *shortPoint,* Point *longPoint*)

Constructor.

Parameters

basePoint	Base point of the triangle.
shortPoint	Point near to basePoint.
longPoint	Point far from basePoint.

2.3.3 Member Function Documentation

2.3.3.1 float SceneObject::calcRotation (SceneObject * *obj*) [static]

Calculate SceneObject's rotation.

Parameters

obj	

Returns

2.3.3.2 float SceneObject::calcSize (SceneObject * *obj*) [static]

Calculate SceneObject's size.

Parameters

obj	

Returns

2.3.3.3 float SceneObject::diff (SceneObject * *obj1,* SceneObject * *obj2*) [static]

Calculates difference between given objects.

Parameters

obj1	
obj2	

Returns

2.3.3.4 float SceneObject::distance (Point * *p1*, Point * *p2*) [static]

Calculate distance between points.

Parameters

p1	
p2	

Returns

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

- sceneobject.h
- sceneobject.cpp

2.4 StableObject Class Reference

The StableObject class. Container for scene objects that has been recognized as stable (filtered).

`#include <stableobject.h>`

Public Member Functions

- StableObject ()

  Empty constructor.
- StableObject (SceneObject sceneObject)

  Standard constructor.

Public Attributes

- SceneObject sceneObject

  Standard scene object.
- float speed

  Current speed.

2.4.1 Detailed Description

The StableObject class. Container for scene objects that has been recognized as stable (filtered).

2.4.2 Constructor & Destructor Documentation

2.4.2.1 StableObject::StableObject (SceneObject *sceneObject*)

Standard constructor.

Parameters

sceneObject	Standard scene object.

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

- stableobject.h
- stableobject.cpp