My Project

Generated by Doxygen 1.9.1

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 image_preprocessor.scaleDetection.ScaleDetector Class Reference

A ScaleDetector instance is used to find a scale, consisting of n alternating black and white squares, in an image.

Public Member Functions

- def __init__ (self, delta1, delta2, n, phi1, phi2, sigma1, sigma2, m, count, accuracy)

  Initializes a new ScaleDetector instance.
- def analyseLine (self, ends, anchor)

  Calculates the positions of the edges between black and white in the binary gray values of a line and checks on basis of the distances between them, if a line is likely to go through <count> squares of the scale.
- def scanArea (self, center)

  Checks the nearby area of a valid line for another valid line, that is assumed to intersect an edge between a black and a white scale square further left in the image.
- def **adaptBorders** (self, left, right)
- def **detectScale** (self, img)

Static Public Member Functions

- def calculateLineEnds (shape, anchors, vectors)

  Calculates the intersection points between some lines and the border of an image.
- def calculateLine (ends, anchor)

  Calculates the indices of all pixels on a line between two end pixels.

2.1.1 Detailed Description

A ScaleDetector instance is used to find a scale, consisting of n alternating black and white squares, in an image.

To work properly the following conditions must be met.

- The beginning of the scale points to the left.
- One square of the scale in the image is wider than one fortieth of the image width.
- More than 20 squares, starting from the beginning of the scale are fully visible, without a gap.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 __init__()

```
def image_preprocessor.scaleDetection.ScaleDetector.__init__ (
              self,
              delta1,
              delta2,
              n,
              phi1,
              phi2,
              sigma1,
              sigma2,
              m,
              count,
              accuracy )
```

Initializes a new ScaleDetector instance.

Parameters

delta1	Radius step width while the whole image is scanned.
delta2	Orthogonal step width while the nearby area of a valid line is scanned.
n	Number of orthogonal steps done in each direction while scanning the nearby area.
phi1	Angle step width while the whole image is scanned.
phi2	Angle step width while the nearby area of a valid line is scanned.
sigma1	Start angle step witdh for the border adaption.
sigma2	Minimal angle step witdh for the border adaption.
m	Number of angle steps done in each direction while scanning the nearby area.
count	Number of gray value edges with minimum absolute distance and low varity in distance ratio in a row, the line must go trough so it is assumed to go through the scale.
accuracy	The maximum value the ratio between the distances from an edge to the previous edge and the next edge can deviate from one.
min	The minimum distance between two edges.

2.1.3 Member Function Documentation

2.1.3.1 analyseLine()

```
def image_preprocessor.scaleDetection.ScaleDetector.analyseLine (
              self,
              ends,
              anchor )
```

Calculates the positions of the edges between black and white in the binary gray values of a line and checks on basis of the distances between them, if a line is likely to go through <count> squares of the scale.

Parameters

ends	The end points of the line.

Returns

If the check is positive, the positions where it was assumed that the line intersects the edges between white and black squares.

2.1.3.2 calculateLine()

```
def image_preprocessor.scaleDetection.ScaleDetector.calculateLine (
              ends,
              anchor )  [static]
```

Calculates the indices of all pixels on a line between two end pixels.

Parameters

ends	The end pixels of the line.

Returns

Array of pixel indices.

2.1.3.3 calculateLineEnds()

```
def image_preprocessor.scaleDetection.ScaleDetector.calculateLineEnds (
              shape,
              anchors,
              vectors )  [static]
```

Calculates the intersection points between some lines and the border of an image.

Each line is defined by an anchor point on the line and a vector of length one, giving the direction of the line.

Parameters

shape	The shape of the image.
anchors	Array of anchor points.
vectors	Array of line vectors.

Returns

Array of intersection points.

2.1.3.4 scanArea()

```
def image_preprocessor.scaleDetection.ScaleDetector.scanArea (
              self,
              center )
```

Checks the nearby area of a valid line for another valid line, that is assumed to intersect an edge between a black and a white scale square further left in the image.

Returns

The positions where it was assumed that the line with the most left intersection, intersects the edges between white and black squares.

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

- /home/seb/ros2_ws/src/image_preprocessor/image_preprocessor/scaleDetection.py