Niblack Binarization

v1.0.1.

Generated by Doxygen 1.9.7

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 NiblackBinarization Class Reference

Namespace for std::filesystem methods.

`#include <niblack.hpp>`

Public Member Functions

- NiblackBinarization ()=default
  Default constructor for the NiblackBinariation class.
- NiblackBinarization (cv::Mat &rhs)
  Copy constructor for the NiblackBinariation class.
- NiblackBinarization (cv::Mat &src, const int window_size, const double k, const double &scale)
  Copy constructor for the NiblackBinariation class.
- ~NiblackBinarization ()=default
  Destructor for the NiblackBinariation class.
- NiblackBinarization & operator= (const NiblackBinarization &rhs)
  Access operator for the NiblackBinarization class.
- bool check_the_image (cv::Mat image)
  A method that checks the correctness of the image format.
- cv::Mat niblackThreshold (const cv::Mat &src, int window_size, double k, double &scale)
  Method that performs image binarization using the Niblack method.
- void demonstrateNiblack (const cv::Mat &src, int window_size, double k, double scale, int selected_row, std::string executable_path)
  Method that demonstrates the source and from the binarized image obtained by calling the niblackThreshold method.
- void drawGraph (std::ofstream &file, const std::vector< double > &values, const std::string &color, const std::string &label)
  a method that stores the coordinates of points in a .tex file for visualization
- void plotValues (const std::string &filePath, const std::vector< double > &localIntensity, const std::vector< double > &meanValues, const std::vector< double > &varianceValues, const std::vector< double > &thresholdValues, int selectedRow)
  The method in which the structure of the .tex file is formed, as well as its contents, by calling the drawGraph method.
- void setWindowSize (int window_size)
  window_size setter
- void setK (double k)
  K setter.
- void setTargetRow (int target_row)
  target_row setter
- void setInputImage (const cv::Mat &image)
  image setter

Private Attributes

- cv::Mat image_
- int window_size_ = 1
- double k_ = 0.2
- int target_row_ = 1
- double scale_ = 1.0

3.1.1 Detailed Description

Namespace for std::filesystem methods.

A class for calculating, demonstrating and visualizing the threshold value using the Niblack method

3.1.2 Constructor & Destructor Documentation

3.1.2.1 NiblackBinarization() [1/3]

```
NiblackBinarization::NiblackBinarization ( )  [default]
```

Default constructor for the NiblackBinariation class.

3.1.2.2 NiblackBinarization() [2/3]

```
NiblackBinarization::NiblackBinarization (
            cv::Mat & rhs )
```

Copy constructor for the NiblackBinariation class.

Parameters

in	*cv::Mat&*	rhs - input image

3.1.2.3 NiblackBinarization() [3/3]

```
NiblackBinarization::NiblackBinarization (
            cv::Mat & src,
            const int window_size,
            const double k,
            const double & scale )
```

Copy constructor for the NiblackBinariation class.

Parameters

in	*cv::Mat&*	src - input image
in	*window_size,determining*	the window size for the algorithm
in	*k*	- coefficient affecting the threshold value
in	*scale*	- the coefficient with which the graph will be scaled (default = 1)

3.1.2.4 ∼NiblackBinarization()

```
NiblackBinarization::∼NiblackBinarization ( )  [default]
```

Destructor for the NiblackBinariation class.

3.1.3 Member Function Documentation

3.1.3.1 check_the_image()

```
bool NiblackBinarization::check_the_image (
            cv::Mat image )
```

A method that checks the correctness of the image format.

Parameters

in	*cv::Mat*	image - input image

Returns

true/false

3.1.3.2 demonstrateNiblack()

```
void NiblackBinarization::demonstrateNiblack (
            const cv::Mat & src,
            int window_size,
            double k,
            double scale,
            int selected_row,
            std::string executable_path )
```

Method that demonstrates the source and from the binarized image obtained by calling the niblackThreshold method.

Parameters

in	*cv::Mat&*	src - input image
in	*window_size,determining*	the window size for the algorithm
in	*k*	- coefficient affecting the threshold value
in	*scale*	- the coefficient with which the graph will be scaled (default = 1)
in	*selected_row*	- the row for which the visualization will be performed
in	*executable_path*	- the path for the executable file to pass what to plotValues-method for visualization

3.1.3.3 drawGraph()

```
void NiblackBinarization::drawGraph (
            std::ofstream & file,
            const std::vector< double > & values,
            const std::string & color,
            const std::string & label )
```

a method that stores the coordinates of points in a .tex file for visualization

Parameters

in	*file*	- file for recording coordinates
in	*values*	- array of coordinates of points
in	*color*	- line color on the chart
in	*label*	- the inscription in the legend of the graph

3.1.3.4 niblackThreshold()

```
cv::Mat NiblackBinarization::niblackThreshold (
            const cv::Mat & src,
            int window_size,
            double k,
            double & scale )
```

Method that performs image binarization using the Niblack method.

Parameters

in	*cv::Mat&*	src - input image
in	*window_size,determining*	the window size for the algorithm
in	*k*	- coefficient affecting the threshold value
in	*scale*	- the coefficient with which the graph will be scaled (default = 1)

Returns

cv::Mat object after binarization

3.1.3.5 operator=()

```
NiblackBinarization & NiblackBinarization::operator= (
            const NiblackBinarization & rhs )
```

Access operator for the NiblackBinarization class.

Parameters

in	*rhs*	NiblackBinarization - copyied object

Returns

NiblackBinarization&

3.1.3.6 plotValues()

```
void NiblackBinarization::plotValues (
            const std::string & filePath,
            const std::vector< double > & localIntensity,
            const std::vector< double > & meanValues,
            const std::vector< double > & varianceValues,
            const std::vector< double > & thresholdValues,
            int selectedRow )
```

The method in which the structure of the .tex file is formed, as well as its contents, by calling the drawGraph method.

Parameters

filePath	- the path to the executable file next to which the .tex file will be generated
localIntensity	- array of vectors with local intensity values
meanValues	- array of vectors with mean brightness values
varianceValues	- array of vectors with standart deviation values
thresholdValues	- array of threshold value vectors
selectedRow	- the number of the row selected for visualization

3.1.3.7 setInputImage()

```
void NiblackBinarization::setInputImage (
            const cv::Mat & image )  [inline]
```

image setter

Parameters

in	*image*	- input image

3.1.3.8 setK()

```
void NiblackBinarization::setK (
            double k )  [inline]
```

K setter.

Parameters

in	*k*	- coefficient affecting the threshold value

3.1.3.9 setTargetRow()

```
void NiblackBinarization::setTargetRow (
            int target_row )  [inline]
```

target_row setter

Parameters

in	*target_row*	- the row for which the visualization will be performed

3.1.3.10 setWindowSize()

```
void NiblackBinarization::setWindowSize (
            int window_size )  [inline]
```

window_size setter

Parameters

in	*window_size,determining*	the window size for the algorithm

3.1.4 Member Data Documentation

3.1.4.1 image_

```
cv::Mat NiblackBinarization::image_  [private]
```

3.1.4.2 k_

```
double NiblackBinarization::k_ = 0.2  [private]
```

3.1.4.3 scale_

```
double NiblackBinarization::scale_ = 1.0  [private]
```

3.1.4.4 target_row_

```
int NiblackBinarization::target_row_ = 1  [private]
```

3.1.4.5 window_size_

```
int NiblackBinarization::window_size_ = 1  [private]
```

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

- C:/Projects_C++/OOP_2023/prj.cw/niblack/include/niblack/niblack.hpp
- C:/Projects_C++/OOP_2023/prj.cw/niblack/niblack.cpp

Chapter 4

File Documentation

4.1 C:/Projects_C++/OOP_↩ 2023/prj.cw/niblack/include/niblack/niblack.hpp File Reference

```
#include <opencv2/opencv.hpp>
#include <opencv2/core/utils/logger.hpp>
#include <cmath>
#include <iostream>
#include <vector>
#include <fstream>
#include <filesystem>
```

Classes

- class NiblackBinarization

  Namespace for std::filesystem methods.

4.2 niblack.hpp

Go to the documentation of this file.

```
00001 #ifndef NIBLACKBINARIZATION_H
00002 #define NIBLACKBINARIZATION_H
00003
00004 #include <opencv2/opencv.hpp>
00005 #include <opencv2/core/utils/logger.hpp>
00006
00007 #include <cmath>
00008 #include <iostream>
00009 #include <vector>
00010 #include <fstream>
00011 #include <filesystem>
00012
00016 namespace fs = std::filesystem;
00017
00021 class NiblackBinarization {
00022 public:
00026     NiblackBinarization() = default;
00027
00032     NiblackBinarization(cv::Mat& rhs);
00033
00041     NiblackBinarization(cv::Mat& src, const int window_size, const double k, const double& scale);
00042
```

```
00046     ~NiblackBinarization() = default;
00047
00053     NiblackBinarization& operator=(const NiblackBinarization& rhs);
00054
00060     bool check_the_image(cv::Mat image);
00061
00070     cv::Mat niblackThreshold(const cv::Mat& src, int window_size, double k, double& scale);
00071
00081     void demonstrateNiblack(const cv::Mat& src, int window_size, double k, double scale, int
      selected_row, std::string executable_path);
00082
00090     void drawGraph(std::ofstream& file, const std::vector<double>& values, const std::string& color,
      const std::string& label);
00091
00101     void plotValues(const std::string& filePath, const std::vector<double>& localIntensity, const
      std::vector<double>& meanValues, const std::vector<double>& varianceValues, const std::vector<double>&
      thresholdValues, int selectedRow);
00102
00107     void setWindowSize(int window_size) {
00108         window_size_ = window_size;
00109     }
00110
00115     void setK(double k) {
00116         k_ = k;
00117     }
00118
00123     void setTargetRow(int target_row) {
00124         target_row_ = target_row;
00125     }
00126
00131     void setInputImage(const cv::Mat& image) {
00132         image_ = image.clone();
00133     }
00134
00135 private:
00136     cv::Mat image_; /*image_ - input image */
00137     int window_size_ = 1; /*window_size_, determining the window size for the algorithm */
00138     double k_ = 0.2; /*k - coefficient affecting the threshold value*/
00139     int target_row_ = 1; /*target_row - the row for which the visualization will be performed*/
00140     double scale_ = 1.0; /*scale - the coefficient with which the graph will be scaled (default = 1)*/
00141 };
00142
00143 #endif NIBLACKBINARIZATION_H
```

4.3 C:/Projects_C++/OOP_2023/prj.cw/niblack/niblack.cpp File Reference

```
#include <niblack/niblack.hpp>
```

4.3.1 Detailed Description

Copyright

Copyright 2023 Shestakov Nikolai Licensed under GPL-3.0-or-later

Index