

HOG library

Создано системой Doxygen 1.9.7



# Глава 1

## HOG-feature-descriptor

Course work for “ООП” 2023 class in NUST MISIS



## Глава 2

# Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

<a href="#">HOGDescriptor</a>	Класс для работы с дескриптором HOG (Гистограмма направленных градиентов)	??
<a href="#">HOGSettings</a>	.....	??
<a href="#">texHOG</a>	Класс для создания .tex файлов для визуализации процесса	??



## Глава 3

# Список файлов

### 3.1 Файлы

Полный список файлов.

src/example/ <a href="#">main.cpp</a>	...	??
src/lib/hogdescriptor/ <a href="#">hogdescriptor.cpp</a>	...	??
src/lib/hogdescriptor/include/hogdescriptor/ <a href="#">hogdescriptor.hpp</a>	...	??
src/lib/texvisualization/ <a href="#">texvisualization.cpp</a>	...	??
src/lib/texvisualization/include/texvisualization/ <a href="#">texvisualization.hpp</a>	...	??
src/tests/ <a href="#">hogtest.cpp</a>	...	??





## Глава 4

# Классы

### 4.1 Класс HOGDescriptor

Класс для работы с дескриптором HOG (Гистограмма направленных градиентов)

```
#include <hogdescriptor.hpp>
```

Открытые члены

- [HOGDescriptor](#) ()  
Конструктор для инициализации объекта класса [HOGDescriptor](#) с параметрами по умолчанию
- [HOGDescriptor](#) (const size\_t blockSize, const size\_t cellSize, const size\_t stride, const size\_t binNumber, const size\_t gradType)  
Конструктор для класса [HOGDescriptor](#) со всеми параметрами
- [HOGDescriptor](#) (const size\_t blockSize, const size\_t cellSize)  
Конструктор для создания нового объекта [HOGDescriptor](#).
- [~HOGDescriptor](#) ()  
Деструктор для класса [HOGDescriptor](#).
- void [visualizeHOG](#) (float scale, bool imposed)  
Метод для визуализации гистограмм ячеек [HOGDescriptor](#) в виде стрелок внутри каждой ячейки на изображении
- void [HOGgrid](#) (cv::Mat &image, float thickness, int cellSize)  
Метод для отображения сетки ячеек на изображении
- void [computeHOG](#) (cv::Mat &image)  
Метод для вычисления гистограмм HOG.
- std::vector< float > [getHOGFeatureVector](#) ()  
Метод для получения вектора гистограммы HOG.
- std::vector< float > [getCellHistogram](#) (int y, int x)  
Получение гистограммы ячейки
- std::vector< std::vector< float > > [getBlockHistogram](#) (int y, int x)  
Получение гистограммы блока
- void [saveVectorData](#) (const std::string &executablePath, const std::string &vectorName)  
Сохранение вектора HOG в файл

## Статические открытые данные

- static const size\_t `GRADIENT_SIGNED` = 360  
Разброс градиента на 360 градусов
- static const size\_t `GRADIENT_UNSIGNED` = 180  
Разброс градиента на 180 градусов

## Закрытые члены

- void `computeGradientFeatures` (cv::Mat &image)  
Функция для вычисления амплитуды и ориентации градиента каждого пикселя
- std::vector< std::vector< std::vector< float > > > `computeCellHistograms` (cv::Mat magnitude, cv::Mat orientation, std::vector< std::vector< std::vector< float > > > &cell\_histograms)  
Вычисление гистограмм HOG для каждой ячейки изображения.
- std::vector< float > `cellHistogram` (const cv::Mat &cellMagnitude, const cv::Mat &cellOrientation)  
Метод для вычисления гистограммы для данной ячейки
- void `normalizeBlockHistogram` (std::vector< float > &block\_histogram)  
Функция для нормализации значений из гистограмм HOG для объединенных ячеек из блока
- const std::vector< float > `calculateHOGVector` (const std::vector< std::vector< std::vector< float > > > &cell\_histograms)  
Метод для вычисления вектора гистограммы HOG.

## Закрытые данные

- int `blockSize_`  
Размер блока скользящего окна в пикселях
- int `cellSize_`  
Размер ячейки в пикселях
- int `binNumber_`  
Количество корзин в гистограмме каждой ячейки
- int `binWidth_`  
Ширина корзин в гистограмме каждой ячейки
- int `stride_`  
Шаг скользящего окна в пикселях
- int `gradType_`  
Тип вычисления градиента (беззнаковый или со знаком)
- bool `hogFlag_` = false  
Флаг, указывающий, был ли вычислен вектор гистограммы HOG.
- cv::Mat `imageMagnitude_`  
Амплитуда градиента изображения
- cv::Mat `imageOrientation_`  
Ориентация градиента изображения
- std::vector< std::vector< std::vector< float > > > `cellHistograms_`  
Вектор гистограмм ячеек
- std::vector< float > `hogFeatureVector_`  
Финальный вектор гистограммы HOG.

## 4.1.1 Подробное описание

Класс для работы с дескриптором HOG (Гистограмма направленных градиентов)

## 4.1.2 Конструктор(ы)

## 4.1.2.1 HOGDescriptor() [1/3]

HOGDescriptor::HOGDescriptor ( )

Конструктор для инициализации объекта класса [HOGDescriptor](#) с параметрами по умолчанию

## 4.1.2.2 HOGDescriptor() [2/3]

```
HOGDescriptor::HOGDescriptor (
    const size_t blockSize,
    const size_t cellSize,
    const size_t stride,
    const size_t binNumber,
    const size_t gradType )
```

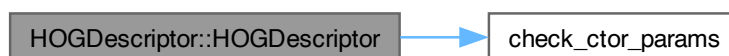
Конструктор для класса [HOGDescriptor](#) со всеми параметрами

Конструктор для класса [HOGDescriptor](#) со всеми параметрами

Аргументы

blockSize	Размер блока скользящего окна в пикселях
cellSize	Размер ячейки в пикселях
stride	Шаг скользящего окна в пикселях
binNumber	Количество корзин в гистограмме каждой ячейки
gradType	Тип вычисления градиента (беззнаковый или со знаком)

Граф вызовов:



## 4.1.2.3 HOGDescriptor() [3/3]

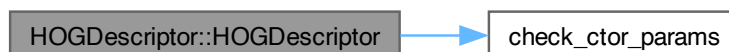
```
HOGDescriptor::HOGDescriptor (
    const size_t blockSize,
    const size_t cellSize )
```

Конструктор для создания нового объекта [HOGDescriptor](#).

Аргументы

blockSize	Размер блока скользящего окна в пикселях
cellSize	Размер ячейки в пикселях

Граф вызовов:



#### 4.1.2.4 ~HOGDescriptor()

HOGDescriptor::~~HOGDescriptor ( )

Деструктор для класса [HOGDescriptor](#).

### 4.1.3 Методы

#### 4.1.3.1 calculateHOGVector()

```
const std::vector< float > HOGDescriptor::calculateHOGVector (
    const std::vector< std::vector< std::vector< float > > > & cell_histograms ) [private]
```

Метод для вычисления вектора гистограммы HOG.

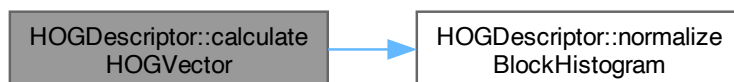
Аргументы

cell_histograms	3D-вектор гистограмм ячеек
-----------------	----------------------------

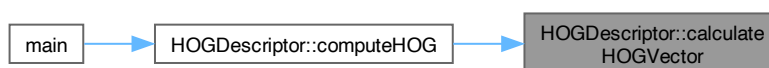
Возвращает

Финальный вектор

Граф вызовов:



Граф вызова функции:



#### 4.1.3.2 cellHistogram()

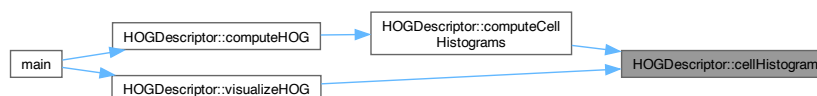
```
std::vector< float > HOGDescriptor::cellHistogram (
    const cv::Mat & cellMagnitude,
    const cv::Mat & cellOrientation ) [private]
```

Метод для вычисления гистограммы для данной ячейки

Аргументы

cellMagnitude	Матрица амплитуд ячейки
cellOrientation	Матрица ориентаций ячейки

Граф вызова функции:



#### 4.1.3.3 computeCellHistograms()

```
std::vector< std::vector< std::vector< float > > > HOGDescriptor::computeCellHistograms (
    cv::Mat magnitude,
    cv::Mat orientation,
    std::vector< std::vector< std::vector< float > > > & cell_histograms ) [private]
```

Вычисление гистограмм HOG для каждой ячейки изображения.

Аргументы

magnitude	Амплитуда градиента, вычисленная с помощью computeGradient()
orientation	Ориентация градиента, вычисленная с помощью computeGradient()
cell_histograms	Выходной вектор гистограмм HOG для каждой ячейки

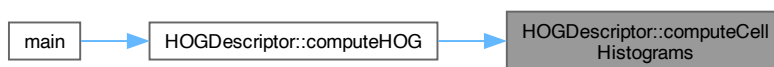
Возвращает

3D-вектор гистограмм ячеек

Граф вызовов:



Граф вызова функции:



#### 4.1.3.4 computeGradientFeatures()

```
void HOGDescriptor::computeGradientFeatures (
    cv::Mat & image ) [private]
```

Функция для вычисления амплитуды и ориентации градиента каждого пикселя

## Аргументы

image	Входное изображение
-------	---------------------

Граф вызова функции:



## 4.1.3.5 computeHOG()

```
void HOGDescriptor::computeHOG (  
    cv::Mat & image )
```

Метод для вычисления гистограмм HOG.

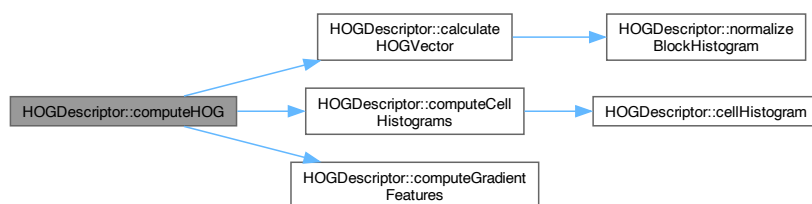
## Аргументы

image	Входное изображение
-------	---------------------

Возвращает

`std::vector<float>` - Вектор гистограммы HOG для изображения

Граф вызовов:



Граф вызова функции:



#### 4.1.3.6 getBlockHistogram()

```
std::vector< std::vector< float > > HOGDescriptor::getBlockHistogram (
    int y,
    int x )
```

Получение гистограммы блока

Аргументы

y	Позиция первой ячейки блока по вертикали
x	Позиция первой ячейки блока по горизонтали

Возвращает

```
std::vector<std::vector<float>>>
```

Граф вызовов:



Граф вызова функции:





## 4.1.3.7 getCellHistogram()

```
std::vector< float > HOGDescriptor::getCellHistogram (
    int y,
    int x )
```

Получение гистограммы ячейки

Аргументы

y	Позиция ячейки по вертикали
x	Позиция ячейки по горизонтали

Возвращает

std::vector<float> Вектор гистограммы для ячейки

Граф вызова функции:



## 4.1.3.8 getHOGFeatureVector()

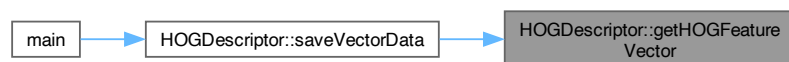
```
std::vector< float > HOGDescriptor::getHOGFeatureVector ( )
```

Метод для получения вектора гистограммы HOG.

Возвращает

std::vector<float>

Граф вызова функции:



## 4.1.3.9 HOGgrid()

```
void HOGDescriptor::HOGgrid (
    cv::Mat & image,
    float thickness,
    int cellSize )
```

Метод для отображения сетки ячеек на изображении

Аргументы

thickness	Толщина линий сетки
cellSize	Размер ячейки в пикселях

Граф вызова функции:



#### 4.1.3.10 normalizeBlockHistogram()

```
void HOGDescriptor::normalizeBlockHistogram (
    std::vector< float > & block_histogram ) [private]
```

Функция для нормализации значений из гистограмм HOG для объединенных ячеек из блока

Аргументы

block	Вектор значений, представляющих гистограммы ячеек для всего блока
-------	-------------------------------------------------------------------

Граф вызова функции:



#### 4.1.3.11 saveVectorData()

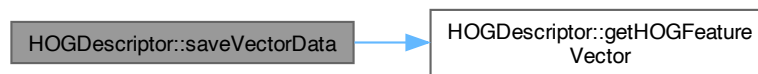
```
void HOGDescriptor::saveVectorData (
    const std::string & executablePath,
    const std::string & vectorName )
```

Сохранение вектора HOG в файл

Аргументы

executablePath	Путь, где будет сохранен файл
vectorName	Имя вектора

Граф вызовов:



Граф вызова функции:



#### 4.1.3.12 visualizeHOG()

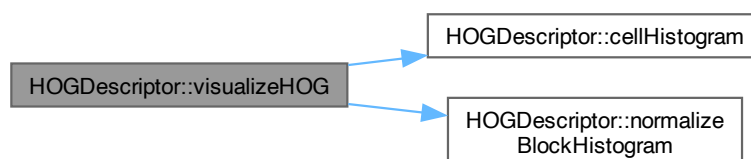
```
void HOGDescriptor::visualizeHOG (
    float scale,
    bool imposed )
```

Метод для визуализации гистограмм ячеек [HOGDescriptor](#) в виде стрелок внутри каждой ячейки на изображении

Аргументы

scale	Масштаб стрелок
imposed	Изображение фона для справки по амплитуде

Граф вызовов:



Граф вызова функции:



#### 4.1.4 Данные класса

##### 4.1.4.1 binNumber\_

```
int HOGDescriptor::binNumber_ [private]
```

Количество корзин в гистограмме каждой ячейки

##### 4.1.4.2 binWidth\_

```
int HOGDescriptor::binWidth_ [private]
```

Ширина корзин в гистограмме каждой ячейки

##### 4.1.4.3 blockSize\_

```
int HOGDescriptor::blockSize_ [private]
```

Размер блока скользящего окна в пикселях

##### 4.1.4.4 cellHistograms\_

```
std::vector<std::vector<std::vector<float> > > > HOGDescriptor::cellHistograms_ [private]
```

Вектор гистограмм ячеек

##### 4.1.4.5 cellSize\_

```
int HOGDescriptor::cellSize_ [private]
```

Размер ячейки в пикселях

##### 4.1.4.6 GRADIENT\_SIGNED

```
const size_t HOGDescriptor::GRADIENT_SIGNED = 360 [static]
```

Разброс градиента на 360 градусов

## 4.1.4.7 GRADIENT\_UNSIGNED

```
const size_t HOGDescriptor::GRADIENT_UNSIGNED = 180 [static]
```

Разброс градиента на 180 градусов

## 4.1.4.8 gradType\_

```
int HOGDescriptor::gradType_ [private]
```

Тип вычисления градиента (беззнаковый или со знаком)

## 4.1.4.9 hogFeatureVector\_

```
std::vector<float> HOGDescriptor::hogFeatureVector_ [private]
```

Финальный вектор гистограммы HOG.

## 4.1.4.10 hogFlag\_

```
bool HOGDescriptor::hogFlag_ = false [private]
```

Флаг, указывающий, был ли вычислен вектор гистограммы HOG.

## 4.1.4.11 imageMagnitude\_

```
cv::Mat HOGDescriptor::imageMagnitude_ [private]
```

Амплитуда градиента изображения

## 4.1.4.12 imageOrientation\_

```
cv::Mat HOGDescriptor::imageOrientation_ [private]
```

Ориентация градиента изображения

## 4.1.4.13 stride\_

```
int HOGDescriptor::stride_ [private]
```

Шаг скользящего окна в пикселях

Объявления и описания членов классов находятся в файлах:

- `src/lib/hogdescriptor/include/hogdescriptor/hogdescriptor.hpp`
- `src/lib/hogdescriptor/hogdescriptor.cpp`

## 4.2 Структура HOGSettings

Открытые атрибуты

- int [blockSize](#)
- int [cellSize](#)
- int [stride](#)
- int [binNumber](#)
- int [gradType](#)
- int [binWidth](#)
- std::string [folderPath](#)
- std::string [saveVectorData](#)

### 4.2.1 Данные класса

#### 4.2.1.1 binNumber

int HOGSettings::binNumber

#### 4.2.1.2 binWidth

int HOGSettings::binWidth

#### 4.2.1.3 blockSize

int HOGSettings::blockSize

#### 4.2.1.4 cellSize

int HOGSettings::cellSize

#### 4.2.1.5 folderPath

std::string HOGSettings::folderPath

#### 4.2.1.6 gradType

int HOGSettings::gradType

#### 4.2.1.7 saveVectorData

std::string HOGSettings::saveVectorData

## 4.2.1.8 stride

int HOGSettings::stride

Объявления и описания членов структуры находятся в файле:

- src/example/[main.cpp](#)

## 4.3 Класс texHOG

Класс для создания .tex файлов для визуализации процесса

```
#include <texvisualization.hpp>
```

Открытые члены

- [texHOG](#) ()=default  
Конструктор класса
- void [cellHistogramPlot](#) (std::vector< float > cellHistogram, int binWidth, const std::string &executablePath)  
Метод для создания .tex файла с графиком гистограммы ячейки
- void [blockHistogramPlot](#) (std::vector< std::vector< float > > blockHistogram, int binWidth, const std::string &executablePath)  
Метод для создания файла .tex с гистограммами ячеек в заданном блоке

## 4.3.1 Подробное описание

Класс для создания .tex файлов для визуализации процесса

## 4.3.2 Конструктор(ы)

## 4.3.2.1 texHOG()

```
texHOG::texHOG ( ) [default]
```

Конструктор класса

## 4.3.3 Методы

## 4.3.3.1 blockHistogramPlot()

```
void texHOG::blockHistogramPlot (
    std::vector< std::vector< float > > blockHistogram,
    int binWidth,
    const std::string & executablePath )
```

Метод для создания файла .tex с гистограммами ячеек в заданном блоке

## Аргументы

blockHistogram	Матрица гистограмм ячеек
blockWidth	Ширина блока гистограммы
executablePath	Путь к файлу .tex

Граф вызова функции:



## 4.3.3.2 cellHistogramPlot()

```

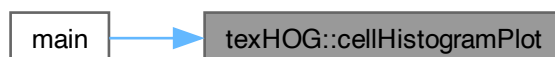
void texHOG::cellHistogramPlot (
    std::vector< float > cellHistogram,
    int binWidth,
    const std::string & executablePath )
  
```

Метод для создания .tex файла с графиком гистограммы ячейки

## Аргументы

cellHistogram	Гистограмма ячейки
blockWidth	Ширина блока гистограммы
executablePath	Путь к файлу .tex

Граф вызова функции:



Объявления и описания членов классов находятся в файлах:

- [src/lib/texvisualization/include/texvisualization/texvisualization.hpp](#)
- [src/lib/texvisualization/texvisualization.cpp](#)



## Глава 5

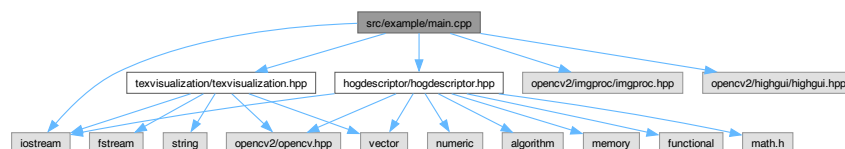
# Файлы

### 5.1 Файл README.md

### 5.2 Файл src/example/main.cpp

```
#include <hogdescriptor/hogdescriptor.hpp>
#include <texvisualization/texvisualization.hpp>
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include <iostream>
```

Граф включаемых заголовочных файлов для main.cpp:



### Классы

- struct [HOGSettings](#)

### Функции

- void [saveSettingsToFile](#) (const [HOGSettings](#) &settings)
- [HOGSettings loadSettingsFromFile](#) ()
- int [main](#) (int argc, char \*\*argv)

## 5.2.1 Функции

### 5.2.1.1 loadSettingsFromFile()

`HOGSettings` loadSettingsFromFile ( )

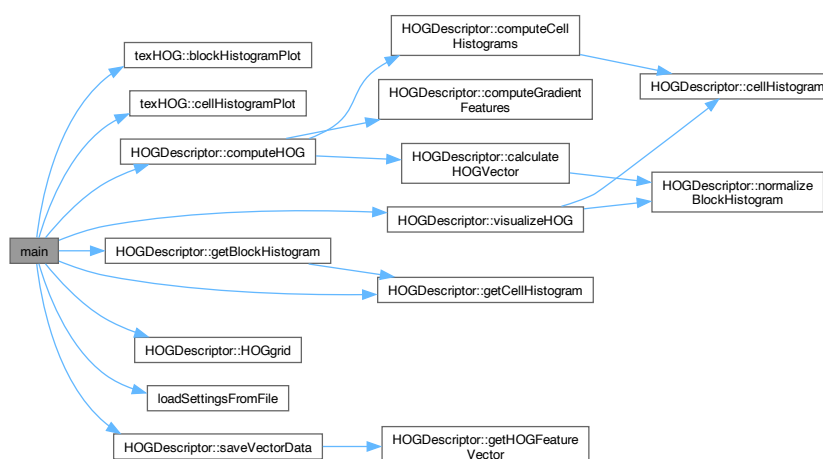
Граф вызова функции:



### 5.2.1.2 main()

```
int main (
    int argc,
    char ** argv )
```

Граф вызовов:



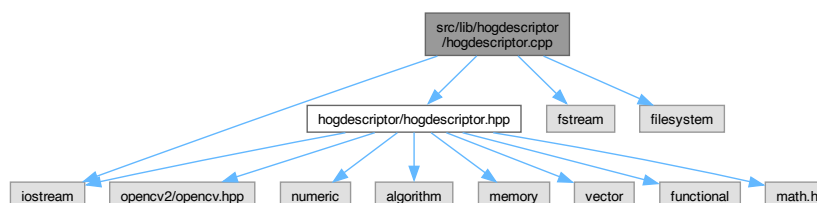
### 5.2.1.3 saveSettingsToFile()

```
void saveSettingsToFile (
    const HOGSettings & settings )
```

### 5.3 Файл src/lib/hogdescriptor/hogdescriptor.cpp

```
#include <hogdescriptor/hogdescriptor.hpp>
#include <iostream>
#include <fstream>
#include <filesystem>
```

Граф включаемых заголовочных файлов для hogdescriptor.cpp:



#### Функции

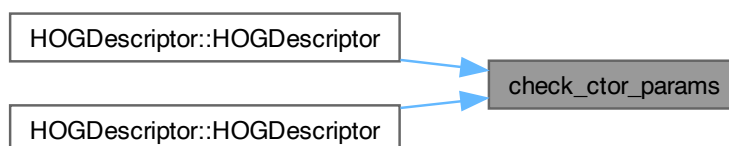
- void [check\\_ctor\\_params](#) (size\_t blockSize, size\_t cellSize, size\_t stride, size\_t binNumber, size\_t gradType)

#### 5.3.1 Функции

##### 5.3.1.1 check\_ctor\_params()

```
void check_ctor_params (
    size_t blockSize,
    size_t cellSize,
    size_t stride,
    size_t binNumber,
    size_t gradType )
```

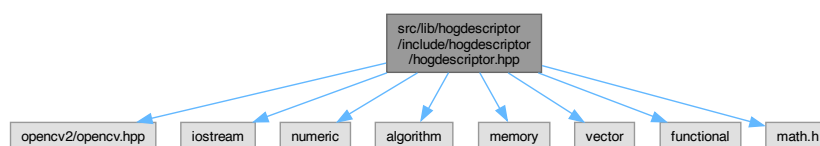
Граф вызова функции:



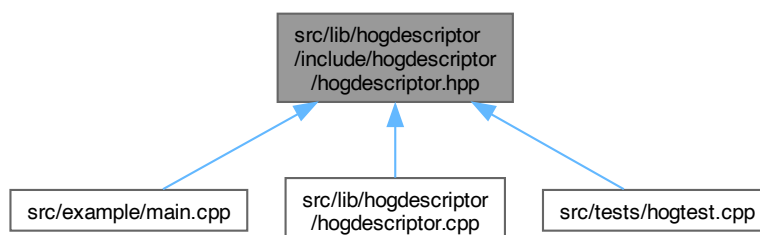
## 5.4 Файл src/lib/hogdescriptor/include/hogdescriptor/hogdescriptor.hpp

```
#include <opencv2/opencv.hpp>
#include <iostream>
#include <numeric>
#include <algorithm>
#include <memory>
#include <vector>
#include <functional>
#include <math.h>
```

Граф включаемых заголовочных файлов для hogdescriptor.hpp:



Граф файлов, в которые включается этот файл:



### Классы

- class [HOGDescriptor](#)

Класс для работы с дескриптором HOG (Гистограмма направленных градиентов)

## 5.5 hogdescriptor.hpp

[См. документацию.](#)

```
00001 #ifndef HOGDESCRIPTOR_H
00002 #define HOGDESCRIPTOR_H
00003
00004 #include <opencv2/opencv.hpp>
00005 #include <iostream>
00006 #include <numeric>
00007 #include <algorithm>
00008 #include <memory>
00009 #include <vector>
00010 #include <functional>
```

```

00011 #include <math.h>
00012
00016 class HOGDescriptor {
00017 public:
00021     HOGDescriptor();
00032     HOGDescriptor(const size_t blockSize, const size_t cellSize,
00033         const size_t stride, const size_t binNumber, const size_t gradType);
00041     HOGDescriptor(const size_t blockSize, const size_t cellSize);
00042
00046     ~HOGDescriptor();
00047
00048 public: // Публичные методы для визуализации
00055     void visualizeHOG(float scale, bool imposed);
00056
00063     void HOGgrid(cv::Mat& image, float thickness, int cellSize);
00064
00065 public:
00066     static const size_t GRADIENT_SIGNED = 360;
00067     static const size_t GRADIENT_UNSIGNED = 180;
00068
00075     void computeHOG(cv::Mat& image);
00076
00082     std::vector<float> getHOGFeatureVector();
00083
00091     std::vector<float> getCellHistogram(int y, int x);
00092
00100     std::vector<std::vector<float>> getBlockHistogram(int y, int x);
00101
00108     void saveVectorData(const std::string& executablePath, const std::string& vectorName);
00109
00110 private:
00116     void computeGradientFeatures(cv::Mat& image);
00117
00126     std::vector<std::vector<std::vector<float>>> computeCellHistograms(cv::Mat magnitude, cv::Mat orientation,
std::vector<std::vector<std::vector<float>>>& cell_histograms);
00127
00134     std::vector<float> cellHistogram(const cv::Mat& cellMagnitude, const cv::Mat& cellOrientation);
00135
00141     void normalizeBlockHistogram(std::vector<float>& block_histogram);
00142
00149     const std::vector<float> calculateHOGVector(const std::vector<std::vector<std::vector<float>>>& cell_histograms);
00150
00151 private:
00152     int blockSize_;
00153     int cellSize_;
00154     int binNumber_;
00155     int binWidth_;
00156     int stride_;
00157     int gradType_;
00158
00159     bool hogFlag_ = false;
00160
00161     cv::Mat imageMagnitude_;
00162     cv::Mat imageOrientation_;
00163
00164     std::vector<std::vector<std::vector<float>>> cellHistograms_;
00165     std::vector<float> hogFeatureVector_;
00166
00167 };
00168
00169 #endif //HOGDESCRIPTOR_H

```

## 5.6 Файл

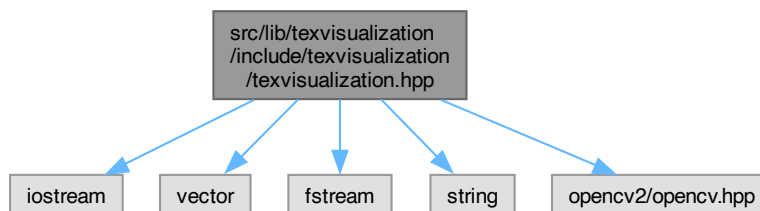
src/lib/texvisualization/include/texvisualization/texvisualization.hpp

```

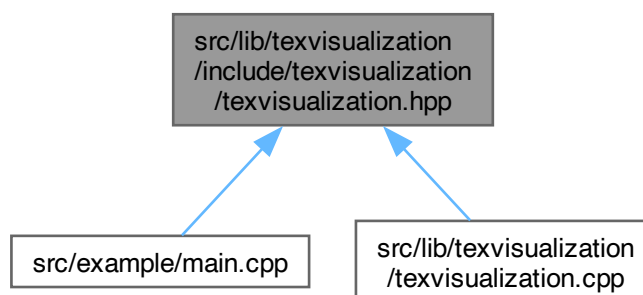
#include <iostream>
#include <vector>
#include <fstream>
#include <string>
#include <opencv2/opencv.hpp>

```

Граф включаемых заголовочных файлов для texvisualization.hpp:



Граф файлов, в которые включается этот файл:



## Классы

- class `texHOG`

Класс для создания .tex файлов для визуализации процесса

## 5.7 texvisualization.hpp

См. документацию.

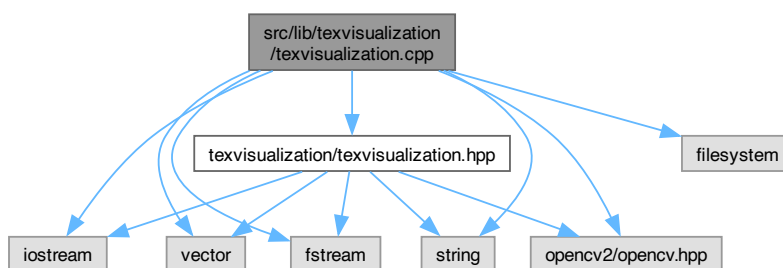
```

00001 #ifndef TEXHOG
00002 #define TEXHOG
00003
00004 #include <iostream>
00005 #include <vector>
00006 #include <fstream>
00007 #include <string>
00008 #include <opencv2/opencv.hpp>
00009
00013 class texHOG{
00014 public:
00019     texHOG() = default;
00020
00028     void cellHistogramPlot(std::vector<float> cellHistogram, int binWidth, const std::string& executablePath);
00029
00037     void blockHistogramPlot(std::vector<std::vector<float>> blockHistogram, int binWidth, const std::string&
executablePath);
00038 };
00039
00040 #endif
  
```

## 5.8 Файл src/lib/texvisualization/texvisualization.cpp

```
#include <texvisualization/texvisualization.hpp>
#include <opencv2/opencv.hpp>
#include <string>
#include <iostream>
#include <vector>
#include <fstream>
#include <filesystem>
```

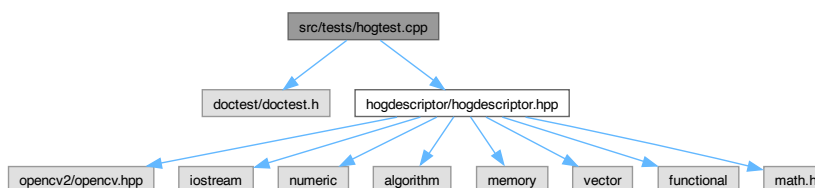
Граф включаемых заголовочных файлов для texvisualization.cpp:



## 5.9 Файл src/tests/hogtest.cpp

```
#include <doctest/doctest.h>
#include <hogdescriptor/hogdescriptor.hpp>
```

Граф включаемых заголовочных файлов для hogtest.cpp:



### Макросы

- `#define DOCTEST_CONFIG_IMPLEMENT_WITH_MAIN`

### 5.9.1 Макросы

#### 5.9.1.1 DOCTEST\_CONFIG\_IMPLEMENT\_WITH\_MAIN

```
#define DOCTEST_CONFIG_IMPLEMENT_WITH_MAIN
```

