

HOG library

Создано системой Doxygen 1.9.7

1 Алфавитный указатель классов	1
1.1 Классы	1
2 Список файлов	3
2.1 Файлы	3
3 Классы	5
3.1 Класс HOGDescriptor	5
3.1.1 Подробное описание	6
3.1.2 Конструктор(ы)	7
3.1.2.1 HOGDescriptor() [1/3]	7
3.1.2.2 HOGDescriptor() [2/3]	7
3.1.2.3 HOGDescriptor() [3/3]	7
3.1.2.4 ~HOGDescriptor()	8
3.1.3 Методы	8
3.1.3.1 calculateHOGVector()	8
3.1.3.2 cellHistogram()	9
3.1.3.3 computeCellHistograms()	10
3.1.3.4 computeGradientFeatures()	10
3.1.3.5 computeHOG()	11
3.1.3.6 getBlockHistogram()	12
3.1.3.7 getCellHistogram()	13
3.1.3.8 getHOGFeatureVector()	13
3.1.3.9 HOGgrid()	13
3.1.3.10 normalizeBlockHistogram()	14
3.1.3.11 saveVectorData()	14
3.1.3.12 visualizeHOG()	15
3.1.4 Данные класса	16
3.1.4.1 binNumber_	16
3.1.4.2 binWidth_	16
3.1.4.3 blockSize_	16
3.1.4.4 cellHistograms_	16
3.1.4.5 cellSize_	16
3.1.4.6 GRADIENT_SIGNED	16
3.1.4.7 GRADIENT_UNSIGNED	17
3.1.4.8 gradType_	17
3.1.4.9 hogFeatureVector_	17
3.1.4.10 hogFlag_	17
3.1.4.11 imageMagnitude_	17
3.1.4.12 imageOrientation_	17
3.1.4.13 stride_	17
3.2 Структура HOGSettings	18
3.2.1 Подробное описание	18
3.2.2 Данные класса	18

3.2.2.1 binNumber	18
3.2.2.2 binWidth	18
3.2.2.3 blockSize	18
3.2.2.4 cellSize	18
3.2.2.5 folderPath	18
3.2.2.6 gradType	18
3.2.2.7 saveVectorData	19
3.2.2.8 stride	19
3.3 Класс texHOG	19
3.3.1 Подробное описание	19
3.3.2 Конструктор(ы)	19
3.3.2.1 texHOG()	19
3.3.3 Методы	19
3.3.3.1 blockHistogramPlot()	19
3.3.3.2 cellHistogramPlot()	20
4 Файлы	21
4.1 Файл /Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/example/main.cpp	21
4.1.1 Подробное описание	22
4.1.2 Функции	22
4.1.2.1 loadSettingsFromFile()	22
4.1.2.2 main()	22
4.2 Файл /Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/hogdescriptor/hogdescriptor.cpp	
4.2.1 Функции	24
4.2.1.1 check_ctor_params()	24
4.3 Файл /Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/hogdescriptor/include/hogdescriptor.h	
4.4 hogdescriptor.hpp	25
4.5 Файл /Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/texvisualization/include/texvisualization.h	
4.6 texvisualization.hpp	27
4.7 Файл /Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/texvisualization/texvisualization.cpp	
4.8 Файл /Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/tests/hogtest.cpp	28
4.8.1 Макросы	28
4.8.1.1 DOCTEST_CONFIG_IMPLEMENT_WITH_MAIN	28

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

HOGDescriptor	Класс для работы с дескриптором НОГ (Гистограмма направленных градиентов)	5
texHOG	Класс для создания .tex файлов для визуализации процесса	19

Глава 2

Список файлов

2.1 Файлы

Полный список файлов.

`/Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/hogdescriptor/hogdescriptor.cpp`
23

`/Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/hogdescriptor/include/hogdescriptor/hogdescriptor.h`
24

`/Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/texvisualization/texvisualization.cpp`
27

`/Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/texvisualization/include/texvisualization/texvisualization.h`
26

Глава 3

Классы

3.1 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.

3.1.1 Подробное описание

Класс для работы с дескриптором HOG (Гистограмма направленных градиентов)

3.1.2 Конструктор(ы)

3.1.2.1 HOGDescriptor() [1/3]

HOGDescriptor::HOGDescriptor ()

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

3.1.2.2 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 )
```

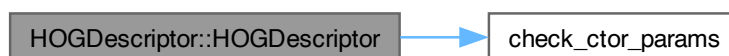
Конструктор для класса [HOGDescriptor](#) со всеми параметрами

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:



3.1.2.3 HOGDescriptor() [3/3]

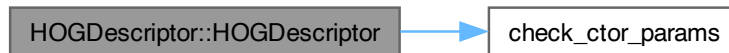
```
HOGDescriptor::HOGDescriptor (
    const size_t blockSize,
    const size_t cellSize )
```

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

Аргументы

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

Граф вызовов:



3.1.2.4 ~HOGDescriptor()

HOGDescriptor::~~HOGDescriptor ()

Деструктор для класса [HOGDescriptor](#).

3.1.3 Методы

3.1.3.1 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
-----------------	----------------------------

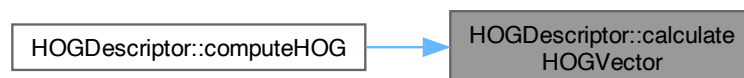
Возвращает

Финальный вектор

Граф вызовов:



Граф вызова функции:



3.1.3.2 cellHistogram()

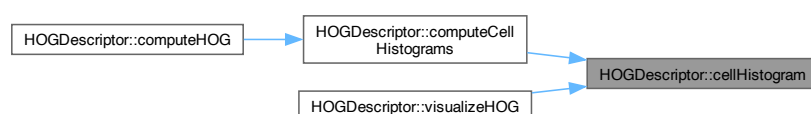
```
std::vector< float > HOGDescriptor::cellHistogram (
    const cv::Mat & cellMagnitude,
    const cv::Mat & cellOrientation ) [private]
```

Метод для вычисления гистограммы для данной ячейки

Аргументы

cellMagnitude	Матрица амплитуд ячейки
cellOrientation	Матрица ориентаций ячейки

Граф вызова функции:



3.1.3.3 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.

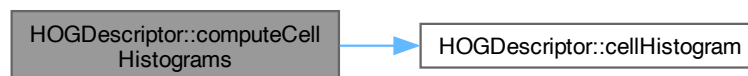
Аргументы

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

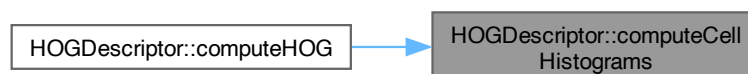
Возвращает

3D-вектор гистограмм ячеек

Граф вызовов:



Граф вызова функции:



3.1.3.4 computeGradientFeatures()

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

Функция для вычисления амплитуды и ориентации градиента каждого пикселя

Аргументы

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

Граф вызова функции:



3.1.3.5 computeHOG()

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

Метод для вычисления гистограмм HOG.

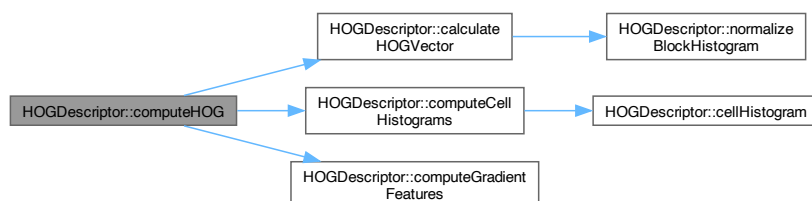
Аргументы

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

Возвращает

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

Граф вызовов:



3.1.3.6 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:



3.1.3.7 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:



3.1.3.8 getHOGFeatureVector()

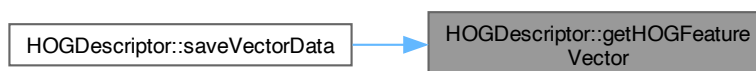
```
std::vector< float > HOGDescriptor::getHOGFeatureVector ( )
```

Метод для получения вектора гистограммы HOG.

Возвращает

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

Граф вызова функции:



3.1.3.9 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

3.1.3.10 normalizeBlockHistogram()

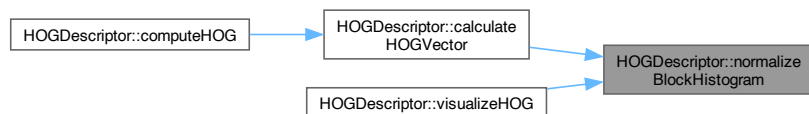
```
void HOGDescriptor::normalizeBlockHistogram (
    std::vector< float > & block_histogram ) [private]
```

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:



3.1.3.11 saveVectorData()

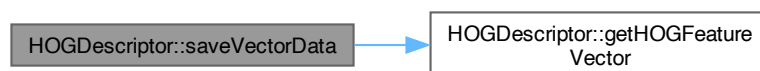
```
void HOGDescriptor::saveVectorData (
    const std::string & executablePath,
    const std::string & vectorName )
```

Сохранение вектора HOG в файл

Аргументы

executablePath	Путь, где будет сохранен файл
vectorName	Имя вектора

Граф вызовов:



3.1.3.12 visualizeHOG()

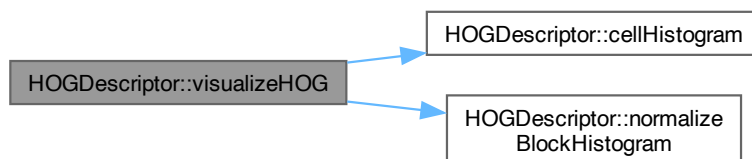
```
void HOGDescriptor::visualizeHOG (
    float scale,
    bool imposed )
```

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:



3.1.4 Данные класса

3.1.4.1 binNumber_

int HOGDescriptor::binNumber_ [private]

Количество корзин в гистограмме каждой ячейки

3.1.4.2 binWidth_

int HOGDescriptor::binWidth_ [private]

Ширина корзин в гистограмме каждой ячейки

3.1.4.3 blockSize_

int HOGDescriptor::blockSize_ [private]

Размер блока скользящего окна в пикселях

3.1.4.4 cellHistograms_

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

Вектор гистограмм ячеек

3.1.4.5 cellSize_

int HOGDescriptor::cellSize_ [private]

Размер ячейки в пикселях

3.1.4.6 GRADIENT_SIGNED

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

Разброс градиента на 360 градусов

3.1.4.7 GRADIENT_UNSIGNED

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

Разброс градиента на 180 градусов

3.1.4.8 gradType_

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

Тип вычисления градиента (беззнаковый или со знаком)

3.1.4.9 hogFeatureVector_

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

Финальный вектор гистограммы HOG.

3.1.4.10 hogFlag_

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

Флаг, указывающий, был ли вычислен вектор гистограммы HOG.

3.1.4.11 imageMagnitude_

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

Амплитуда градиента изображения

3.1.4.12 imageOrientation_

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

Ориентация градиента изображения

3.1.4.13 stride_

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

Шаг скользящего окна в пикселях

Объявления и описания членов классов находятся в файлах:

- `/Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/hogdescriptor/include/hogdescriptor/hogdescriptor.h`
- `/Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/hogdescriptor/hogdescriptor.cpp`

3.2 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

3.2.1 Подробное описание

Класс для создания .tex файлов для визуализации процесса

3.2.2 Конструктор(ы)

3.2.2.1 texHOG()

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

Конструктор класса

3.2.3 Методы

3.2.3.1 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

3.2.3.2 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

Объявления и описания членов классов находятся в файлах:

- /Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/texvisualization/include/texvisualization
- /Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/texvisualization/[texvisualization.cpp](#)

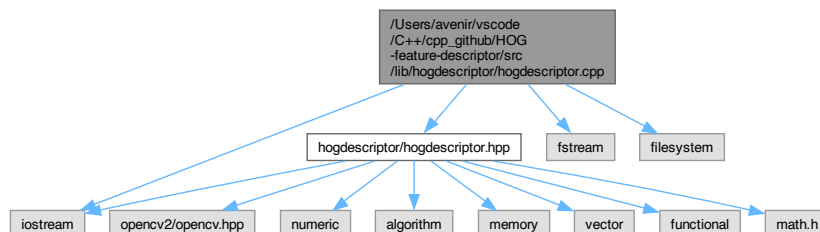
Глава 4

Файлы

4.1 Файл /Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/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)

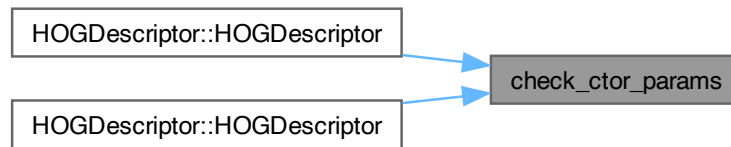
4.1.1 Функции

4.1.1.1 check_ctor_params()

```
void check_ctor_params (
    size_t blockSize,
    size_t cellSize,
    size_t stride,
```

```
size_t binNumber,
size_t gradType )
```

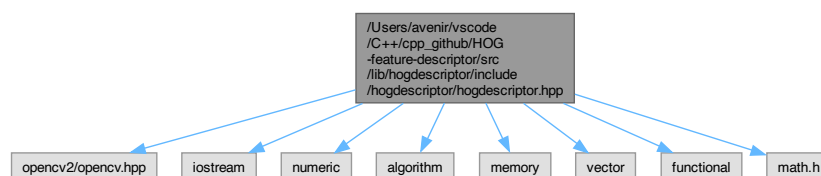
Граф вызова функции:



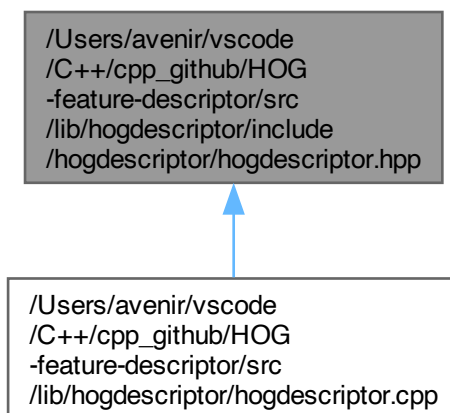
4.2 Файл /Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/hogdescriptor/include/hogdescriptor/hogdescriptor.hpp

```
#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)

4.3 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();
00022
00033     HOGDescriptor(const size_t blockSize, const size_t cellSize,
00034                 const size_t stride, const size_t binNumber, const size_t gradType);
00035
00043     HOGDescriptor(const size_t blockSize, const size_t cellSize);
00044
00048     ~HOGDescriptor();
00049
00050 public: // Публичные методы для визуализации
00057     void visualizeHOG(float scale, bool imposed);
00058
00065     void HOGgrid(cv::Mat& image, float thickness, int cellSize);
00066
00067 public:
00068     static const size_t GRADIENT_SIGNED = 360;
00069     static const size_t GRADIENT_UNSIGNED = 180;
00070
00077     void computeHOG(cv::Mat& image);
00078
00084     std::vector<float> getHOGFeatureVector();
  
```

```

00085
00093     std::vector<float> getCellHistogram(int y, int x);
00094
00102     std::vector<std::vector<float>> getBlockHistogram(int y, int x);
00103
00110     void saveVectorData(const std::string& executablePath, const std::string& vectorName);
00111
00112 private:
00118     void computeGradientFeatures(cv::Mat& image);
00119
00128     std::vector<std::vector<std::vector<float>>> computeCellHistograms(cv::Mat magnitude, cv::Mat orientation,
std::vector<std::vector<std::vector<float>>>& cell_histograms);
00129
00136     std::vector<float> cellHistogram(const cv::Mat& cellMagnitude, const cv::Mat& cellOrientation);
00137
00143     void normalizeBlockHistogram(std::vector<float>& block_histogram);
00144
00151     const std::vector<float> calculateHOGVector(const std::vector<std::vector<std::vector<float>>>& cell_histograms);
00152
00153 private:
00154     int blockSize_;
00155     int cellSize_;
00156     int binNumber_;
00157     int binWidth_;
00158     int stride_;
00159     int gradType_;
00160
00161     bool hogFlag_ = false;
00162
00163     cv::Mat imageMagnitude_;
00164     cv::Mat imageOrientation_;
00165
00166     std::vector<std::vector<std::vector<float>>> cellHistograms_;
00167     std::vector<float> hogFeatureVector_;
00168
00169 };
00170
00171 #endif //HOGDESCRIPTOR_H

```

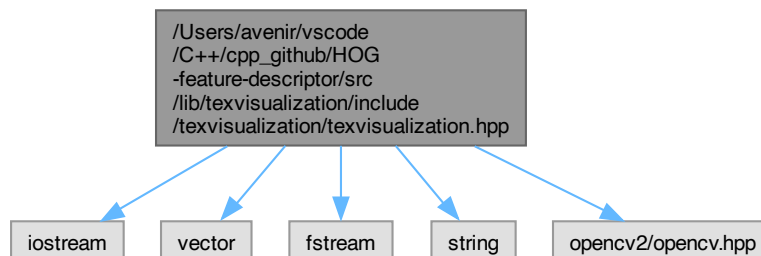
4.4 Файл /Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/src/lib/texvisualization/include/texvisualization/texvisualization.hpp

```

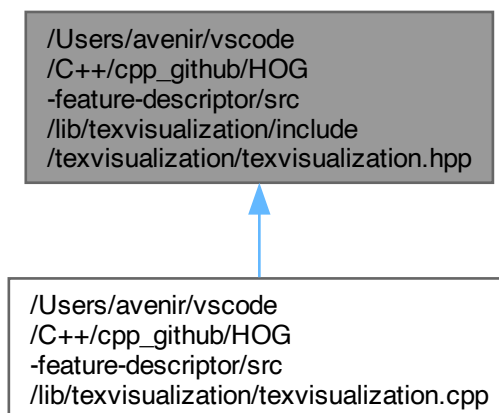
#include <iostream>
#include <vector>
#include <fstream>
#include <string>
#include <opencv2/opencv.hpp>

```

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

4.5 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
  
```

4.6 Файл /Users/avenir/vscode/C++/cpp_github/HOG-feature-descriptor/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:

