

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное
учреждение высшего образования
*«Санкт-Петербургский Государственный университет
Аэрокосмического Приборостроения»*

Кафедра №14

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

асс.		П.С. Санкин
_____	_____	_____
должность	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2
Статистическое распределение элементов цифрового изображения
По курсу: «Основы мультимедия технологий».

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГРУППЫ 1441

	А.А. Протасов
_____	_____
подпись, дата	инициалы, фамилия

Санкт-Петербург
2016

1. Цель работы

Неализовать программу выполняющую построение гистограммы для растрового изображения.

2. Постановка задачи

Для графических файлов написать программу выполняющую построение гистограммы для растрового изображения. Программа должна иметь элементы управления отдельными цветовыми каналами изображения согласно заданию. Гистограмма должна быть построена как для исходного, так и для преобразованного изображения. Графические файлы для проверки изображений создать самостоятельно.

3. Задание

Управление яркостью для всех каналов.

4. Краткие теоритические сведения

Гистограмма — способ графического представления табличных данных. Количественные соотношения некоторого показателя представлены в виде прямоугольников, площади которых пропорциональны. Чаще всего для удобства восприятия ширину прямоугольников берут одинаковую, при этом их высота определяет соотношения отображаемого параметра.// Гистограмма в фотографии — это график статистического распределения элементов цифрового изображения с различной яркостью, в котором по горизонтальной оси представлена яркость, а по вертикали — относительное число пикселей с конкретным значением яркости.

5. Описание метода реализации

Программа была написана в среде для разработки Qt с использованием библиотеки компьютерного зрения OpenCV.

Для открытия изображения и разделения цветовой палитры на разные каналы была использована библиотека OpenCV. Так-же из библиотеки OpenCV был использован шаблон `sturate_cast` для преобразования целого значения(исходное значение пикселя плюс значение на которое мы увеличиваем яркость) в понятное для типа `Mat` 8-битное значение.

6. Листинги

6.1. Тело основной функции

```
1  #include "mainwindow.h"
2  #include <QApplication>
3
4  int main(int argc, char *argv[])
5  {
6      QApplication a(argc, argv);
7      MainWindow w;
8      w.show();
9
10     return a.exec();
11 }
```

6.2. Заголовок класса главного окна

```
1  #ifndef MAINWINDOW_H
2  #define MAINWINDOW_H
3
4  #include <QMainWindow>
5  #include <opencv2/highgui/highgui.hpp>
6  #include <opencv2/imgproc/imgproc.hpp>
7
8  typedef enum{
9      RED,
10     GREEN,
11     BLUE
12 }plane;
13
14 using namespace cv;
15
16 namespace Ui {
17 class MainWindow;
18 }
19
20 class MainWindow : public QMainWindow
21 {
22     Q_OBJECT
23
24 public:
25     explicit MainWindow(QWidget *parent = 0);
26     ~MainWindow();
27     void histDisplay(Mat image, plane RGB);
28
29 private slots:
30     void on_horizontalSlider_valueChanged(int value);
31
32 private:
33     Ui::MainWindow *ui;
34 };
35
```

```

36 inline QImage MatToQImage(const Mat &input);
37 inline Mat QImageToMat(const QImage &input);
38
39 #endif // MAINWINDOW_H

```

6.3. Описание методов класса главного окна

```

1  #include "mainwindow.h"
2  #include "ui_mainwindow.h"
3
4  Mat image = imread("/home/toshiki/files/картинки/PNG for LS/kek.png");
5
6  inline QImage MatToQImage(const Mat &input)
7  {
8      QImage image(input.data,
9                  input.cols, input.rows,
10                 static_cast<int>(input.step),
11                 QImage::Format_RGB888);
12     return image.rgbSwapped();
13 }
14
15 QImage img = MatToQImage(image);
16
17 inline Mat QImageToMat(const QImage &input)
18 {
19     QImage swapped;
20     if(input.format() == QImage::Format_RGB32 )
21         swapped = input.convertToFormat( QImage::Format_RGB888 );
22     swapped = input.rgbSwapped();
23     return cv::Mat(swapped.height(), swapped.width(),
24                   CV_8UC3,
25                   const_cast<uchar*>(swapped.bits()),
26                   static_cast<size_t>(swapped.bytesPerLine())
27                   ).clone();
28 }
29
30 void MainWindow::histDisplay(Mat image, plane RGB)
31 {
32     int hist[256];
33     memset(hist, 0, 256*sizeof(*hist));
34     for(int y = 0; y < image.rows; y++)
35         for(int x = 0; x < image.cols; x++)
36             hist[(int)image.at<uchar>(y,x)]++;
37     int hist_w = 256; int hist_h = ui->label_3->size().height();
38     int bin_w = cvRound((double) hist_w/256);
39     Mat histImage(hist_h, hist_w, CV_8UC3, Scalar(50, 50, 50));
40     int max = hist[0];
41     for(int i = 1; i < 256; i++)
42         if(max < hist[i])
43             max = hist[i];
44     for(int i = 0; i < 256; i++)
45         hist[i] = ((double)hist[i]/max)*histImage.rows;
46     switch ( RGB ) {

```

```

47     case RED:
48         for(int i = 0; i < 256; i++)
49             line(histImage, Point(bin_w*(i), hist_h),
50                 Point(bin_w*(i), hist_h - hist[i]),
51                 Scalar(0,0,255));
52         ui->label_3->setPixmap(QPixmap::fromImage(MatToQImage(histImage)));
53         ui->label_3->show();
54         break;
55     case GREEN:
56         for(int i = 0; i < 256; i++)
57             line(histImage, Point(bin_w*(i), hist_h),
58                 Point(bin_w*(i), hist_h - hist[i]),
59                 Scalar(0,255,0));
60         ui->label_4->setPixmap(QPixmap::fromImage(MatToQImage(histImage)));
61         ui->label_4->show();
62         break;
63     case BLUE:
64         for(int i = 0; i < 256; i++)
65             line(histImage, Point(bin_w*(i), hist_h),
66                 Point(bin_w*(i), hist_h - hist[i]),
67                 Scalar(255,0,0));
68         ui->label_5->setPixmap(QPixmap::fromImage(MatToQImage(histImage)));
69         ui->label_5->show();
70         break;
71     default:
72         break;
73 }
74 }
75
76 MainWindow::MainWindow(QWidget *parent) :
77     QMainWindow(parent),
78     ui(new Ui::MainWindow)
79 {
80     ui->setupUi(this);
81
82     ui->verticalLayout->setSpacing(0);
83     ui->verticalLayout->setMargin(0);
84     connect(ui->horizontalSlider, SIGNAL(valueChanged(int)),
85         this, SLOT(on_horizontalSlider_valueChanged(int)));
86     ui->label->setPixmap(QPixmap::fromImage(img).scaled(
87         img.width()/2, img.height()/2, Qt::KeepAspectRatio));
88     ui->label->setMinimumWidth(img.width()/2);
89     ui->label->setMinimumHeight(img.height()/2);
90     ui->label_3->hide();
91     ui->label_4->hide();
92     ui->label_5->hide();
93 }
94
95 QImage brightness(Mat input, int ratio)
96 {
97     Mat new_image = Mat::zeros(input.size(), input.type());
98     for(int y = 0; y < input.rows; y++)
99         for(int x = 0; x < input.cols; x++)
100             for(int c = 0; c < 3; c++)

```

```

101         new_image.at<Vec3b>(y,x)[c] =
102             saturate_cast<uchar>((input.at<Vec3b>(y,x)[c]) + ratio);
103     QImage image = MatToQImage(new_image);
104     return image;
105 }
106
107 MainWindow::~MainWindow()
108 {
109     delete ui;
110 }
111
112 void MainWindow::on_horizontalSlider_valueChanged(int value)
113 {
114     int ratio = value * 2.55;
115     ui->label_2->setText(QString::number(value)+"%");
116     QImage temp(brightness(image,ratio).scaled(
117         img.width()/2, img.height()/2,Qt::KeepAspectRatio));
118     ui->label->setPixmap(QPixmap::fromImage(temp));
119     Mat tmp = QImageToMat(temp);
120     std::vector<Mat> rgb_planes;
121     split(tmp, rgb_planes);
122     if(ui->checkBox->checkState() == Qt::Checked){
123         histDisplay(rgb_planes[2], RED);
124         histDisplay(rgb_planes[1], GREEN);
125         histDisplay(rgb_planes[0], BLUE);
126     } else {
127         ui->label_3->hide();
128         ui->label_4->hide();
129         ui->label_5->hide();
130     }
131 }
132 }

```

6.4. Описание формы главного окна

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <ui version="4.0">
3      <class>MainWindow</class>
4      <widget class="QMainWindow" name="MainWindow">
5          <property name="geometry">
6              <rect>
7                  <x>0</x>
8                  <y>0</y>
9                  <width>653</width>
10                 <height>612</height>
11             </rect>
12         </property>
13         <property name="windowTitle">
14             <string>BrightnessControll</string>
15         </property>
16         <widget class="QWidget" name="centralWidget">
17             <widget class="QWidget" name="verticalLayoutWidget">
18                 <property name="geometry">

```

```
19     <rect>
20         <x>0</x>
21         <y>0</y>
22         <width>221</width>
23         <height>611</height>
24     </rect>
25 </property>
26 <layout class="QVBoxLayout" name="verticalLayout" stretch="0">
27     <property name="spacing">
28         <number>0</number>
29     </property>
30     <property name="sizeConstraint">
31         <enum>QLayout::SetNoConstraint</enum>
32     </property>
33     <item>
34         <widget class="QLabel" name="label">
35             <property name="layoutDirection">
36                 <enum>Qt::LeftToRight</enum>
37             </property>
38             <property name="text">
39                 <string>TextLabel</string>
40             </property>
41             <property name="alignment">
42                 <set>Qt::AlignCenter</set>
43             </property>
44         </widget>
45     </item>
46 </layout>
47 </widget>
48 <widget class="QSlider" name="horizontalSlider">
49     <property name="geometry">
50         <rect>
51             <x>220</x>
52             <y>0</y>
53             <width>160</width>
54             <height>16</height>
55         </rect>
56     </property>
57     <property name="contextMenuPolicy">
58         <enum>Qt::CustomContextMenu</enum>
59     </property>
60     <property name="layoutDirection">
61         <enum>Qt::LeftToRight</enum>
62     </property>
63     <property name="minimum">
64         <number>-100</number>
65     </property>
66     <property name="maximum">
67         <number>100</number>
68     </property>
69     <property name="tracking">
70         <bool>true</bool>
71     </property>
72     <property name="orientation">
```

```
73     <enum>Qt::Horizontal</enum>
74 </property>
75 <property name="tickPosition">
76     <enum>QSlider::NoTicks</enum>
77 </property>
78 </widget>
79 <widget class="QLabel" name="label_2">
80     <property name="geometry">
81         <rect>
82             <x>220</x>
83             <y>20</y>
84             <width>161</width>
85             <height>16</height>
86         </rect>
87     </property>
88     <property name="text">
89         <string>0</string>
90     </property>
91     <property name="alignment">
92         <set>Qt::AlignCenter</set>
93     </property>
94 </widget>
95 <widget class="QCheckBox" name="checkBox">
96     <property name="geometry">
97         <rect>
98             <x>230</x>
99             <y>40</y>
100             <width>181</width>
101             <height>20</height>
102         </rect>
103     </property>
104     <property name="text">
105         <string>Own Implement histogram</string>
106     </property>
107 </widget>
108 <widget class="QWidget" name="verticalLayoutWidget_2">
109     <property name="geometry">
110         <rect>
111             <x>220</x>
112             <y>60</y>
113             <width>431</width>
114             <height>331</height>
115         </rect>
116     </property>
117     <layout class="QVBoxLayout" name="verticalLayout_2">
118         <item>
119             <widget class="QLabel" name="label_3">
120                 <property name="text">
121                     <string>Red</string>
122                 </property>
123             </widget>
124         </item>
125         <item>
126             <widget class="QLabel" name="label_4">
```



```

127     <property name="text">
128         <string>Green</string>
129     </property>
130 </widget>
131 </item>
132 <item>
133     <widget class="QLabel" name="label_5">
134         <property name="text">
135             <string>Blue</string>
136         </property>
137     </widget>
138 </item>
139 </layout>
140 </widget>
141 </widget>
142 </widget>
143 <layoutdefault spacing="6" margin="11"/>
144 <resources/>
145 <connections/>
146 </ui>

```

7. Примеры работы программы



