**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Телевидения и Видеотехники**

**курсовой проект**

**по дисциплине «Информационные технологии»**

**Тема: Разработка редактора изображений**

|  |  |  |
| --- | --- | --- |
| Студент гр. 2106 |  | Петров В.Е. |
| Преподаватель |  | Чиркунова А.А. |

Санкт-Петербург

2023

**ЗАДАНИЕ**

**на курсовой проект**

|  |  |  |
| --- | --- | --- |
| Студент Петров В.Е. | | |
| Группа 2106 | | |
| Тема проекта: Разработка редактора изображений | | |
| Исходные данные:  Изображения формата .bmp; 1.0(Перемещение прямоугольной области);  1.2(Ластик);2.12Уменьшение масштаба всего изображения); 3.5(Гамма-коррекция); 4.2(Нормализация гистограммы); 5.16(пороговая сегментация инверсный цвета YCbCr). | | |
| Содержание пояснительной записки:  Содержание, Введение, 1. Теоретическая часть,2. Программные документы, Заключение, Список использованных источников | | |
| Предполагаемый объем пояснительной записки:  Не менее 15 страниц. | | |
| Дата выдачи задания:24.03.2023 | | |
| Дата сдачи реферата: | | |
| Дата защиты реферата: | | |
| Студент |  | Петров В.Е. |
| Преподаватель |  | Чиркунова А.А. |

**Аннотация**

Мое курсовое задание заключается в обработке изображений. В нем я загружаю изображение, использую фильтры и применяю разные методы работы с ним. Курсовое задние заключается в создании или загрузки изображения и изменении его визуальных свойств. Используются такие фильтры как:

-Перемещение прямоугольной области

-Ластик

- Уменьшение масштаба всего изображения

- Гамма-коррекция

- Нормализация гистограммы

- Пороговая сегментация

**содержание**

|  |  |  |
| --- | --- | --- |
|  | Введение | 6 |
| 1. | Теоретическая часть | 7 |
| 1.0. | Фильтр «Перемещение прямоугольной области» | 7 |
| 1.1. | Фильтр «Ластик» |  |
| 1.2. | Фильтр «Уменьшение масштаба всего изображения» | 7 |
| 1.3. | Фильтр «Гамма-коррекция» | 7 |
| 1.4. | Фильтр «Нормализация гистограммы» | 7 |
| 1.5. | Фильтр «Пороговая сегментация» | 7 |
| 2. | Программные документы | 8 |
| 2.1. | Спецификация | 8 |
| 2.2. | Описание программы | 8 |
| 2.2.1. | Общие сведения | 0 |
| 2.2.2. | Функциональное назначение | 0 |
| 2.2.3. | Описание логической структуры | 0 |
| 2.2.4. | Используемые технические средства | 0 |
| 2.2.5. | Вызов и загрузка | 0 |
| 2.2.6. | Выходные данные | 0 |
| 2.3. | Текст программы | 0 |
| 2.4. | Эксплуатационные документы | 0 |
| 2.4.1. | Ведомость эксплуатационных документов | 0 |
| 2.4.2. | Описание применения | 0 |
| 2.4.2.1. | Назначение программы | 0 |
| 2.4.2.2. | Условия применения | 0 |
| 2.4.2.3. | Описание задачи | 0 |
| 2.4.2.4 | Входные и выходные данные | 0 |
| 2.4.3. | **Руководство оператора** | 0 |
| 2.4.3.1. | Назначение программы | 0 |
| 2.4.3.2. | Условия выполнения программы | 0 |
| 2.4.3.3. | Выполнение программы | 0 |
| 2.4.3.4. | Выполнение программы | 0 |
| 2.4.3.5. | Сообщения оператору | 0 |
| 2.4.4. | **Описание языка** | 0 |
| 2.4.4.1. | Общие сведения | 0 |
| 2.4.4.2. | Элементы языка | 0 |
| 2.4.4.3. | Средства обмена данными | 0 |
| 2.4.4.4. | Встроенные элементы | 0 |
| 2.4.4.5. | Средства отладки программы | 0 |
|  | Заключение | 0 |
|  | Список использованных источников | 0 |
|  | Приложение А. Ссылочные нормативные документы | 18 |

**введение**

Цель работы: разработка программного обеспечения для редактирования изображений, отладка программного обеспечения.

Требования: перенос части изображения, реализация ластика с помощью ползунков; масштабирование изображения(уменьшения размера всего изображения); предыскажения яркости каналов составляющих изображение(гамма-коррекция) нормализация гистограммы; пороговая сегментация цветных изображений(YCrCb инверсные цвета).

**1. тЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**1.0. Фильтр** «Перемещение прямоугольной области»

**1.1. Фильтр** «Ластик»

**1.2. Фильтр** «Уменьшение масштаба всего изображения»

**1.3. Фильтр** «Гамма-коррекция»

**1.4. Фильтр** «Нормализация гистограммы»

**1.5. Фильтр** «Пороговая сегментация изображений»

**2. пРограммная документация**

**2.1. Спецификация**

- Компоненты – windows 10, visual studio 22, c++(opencv)

|  |  |  |
| --- | --- | --- |
| Обозначение | Наименование | Примечание |
|  |  |  |
|  |  |  |
|  |  |  |

**2.2. Описание программы**

2.2.1. Общие сведения: windows 10, visual studio 22, c++(opencv)

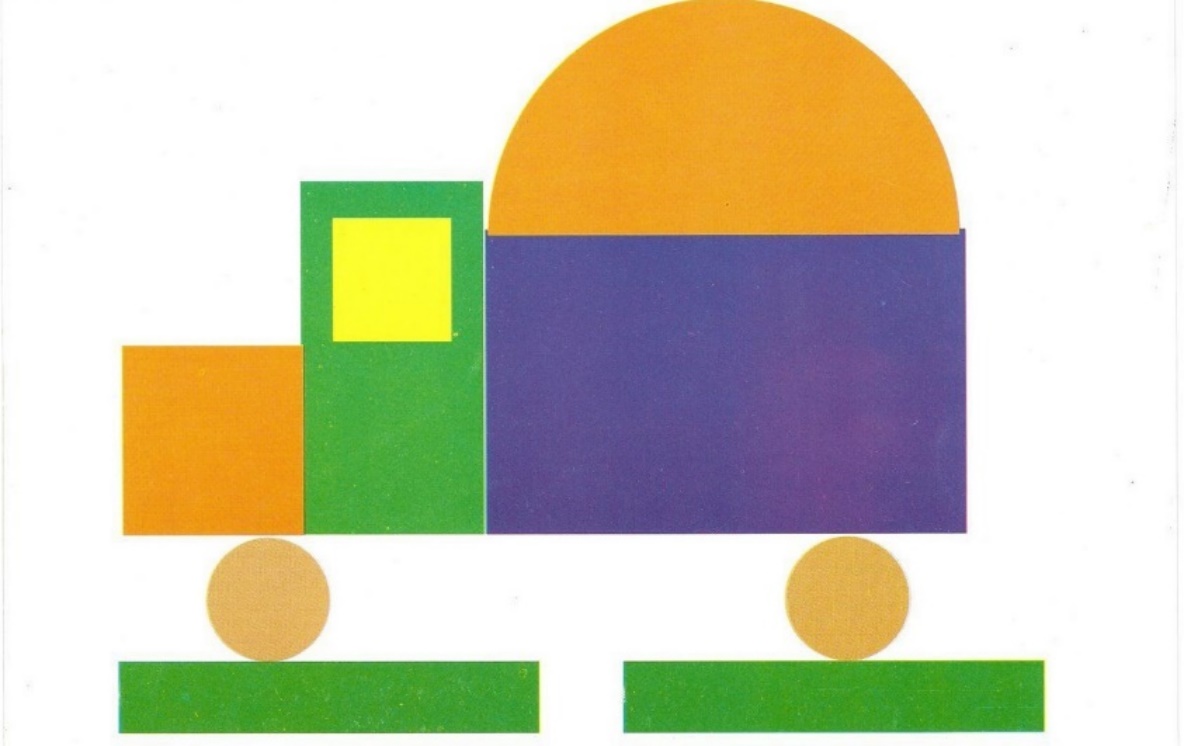
2.2.2. Функциональное назначение: обработка изображений, ограничение: обработка текстовых файлов.

2.2.3. Описание логической структуры блок-схема алгоритма программы,

2.2.4. Используемые технические средства: компьютер на windows 10 с 8 гб ОЗУ и четырехъядерным процессором.

2.2.5. Вызов и загрузка способ вызова программы: запуск программы нажатием клавиши F5, объем оперативной памяти 8 гб, объем программы: 355 строчки кода(32.1 озу)

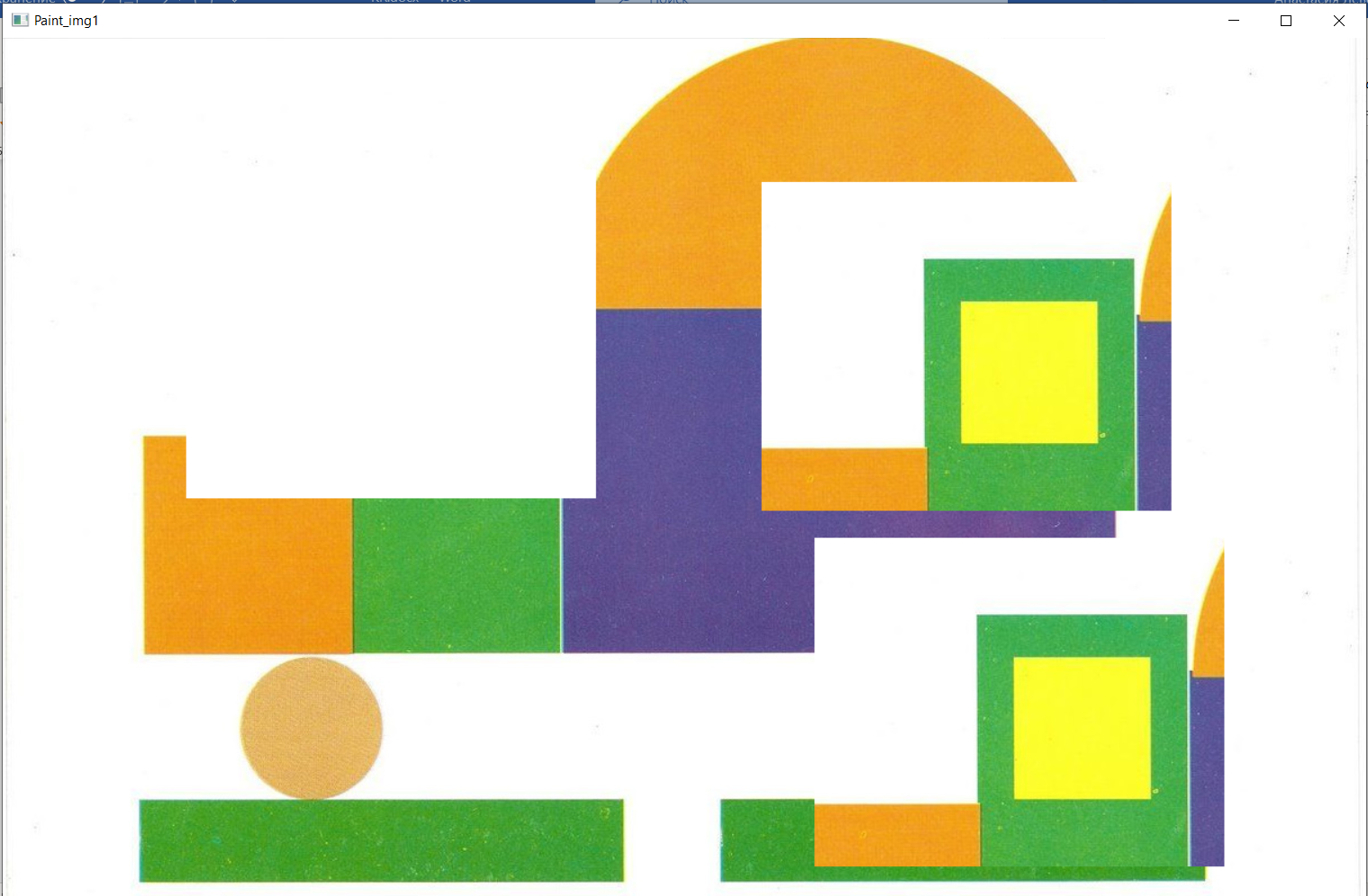
Входные данные:

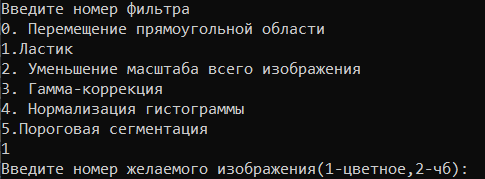


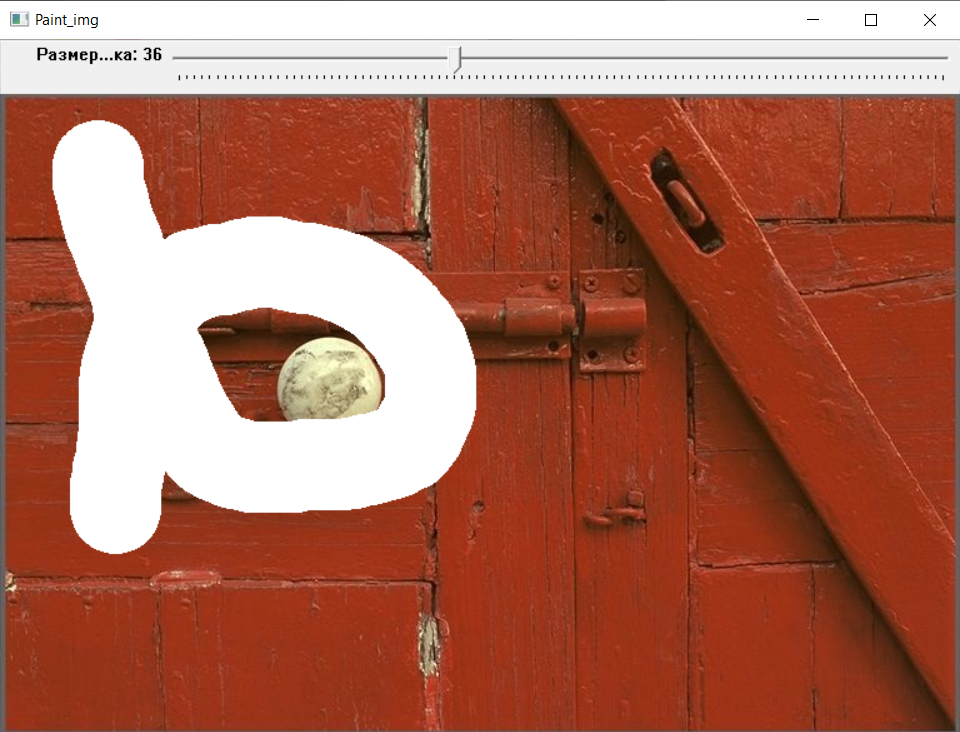


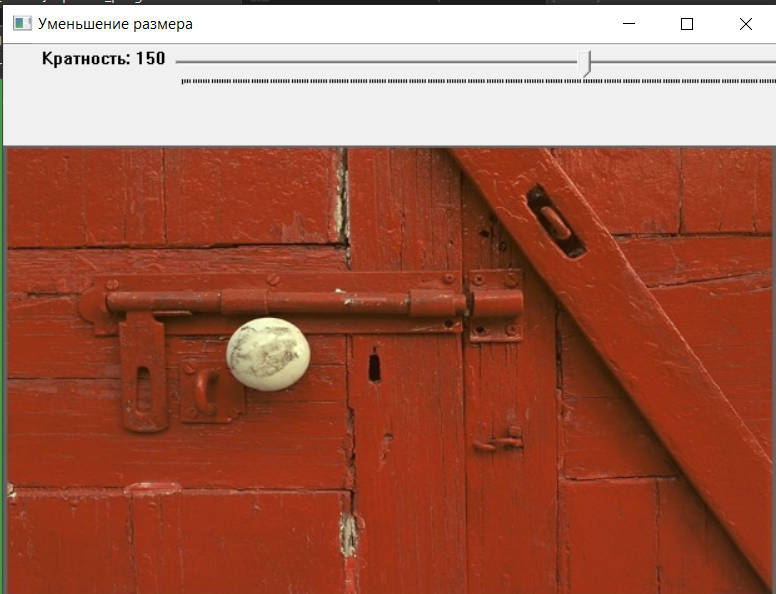


2.2.6. Выходные данные

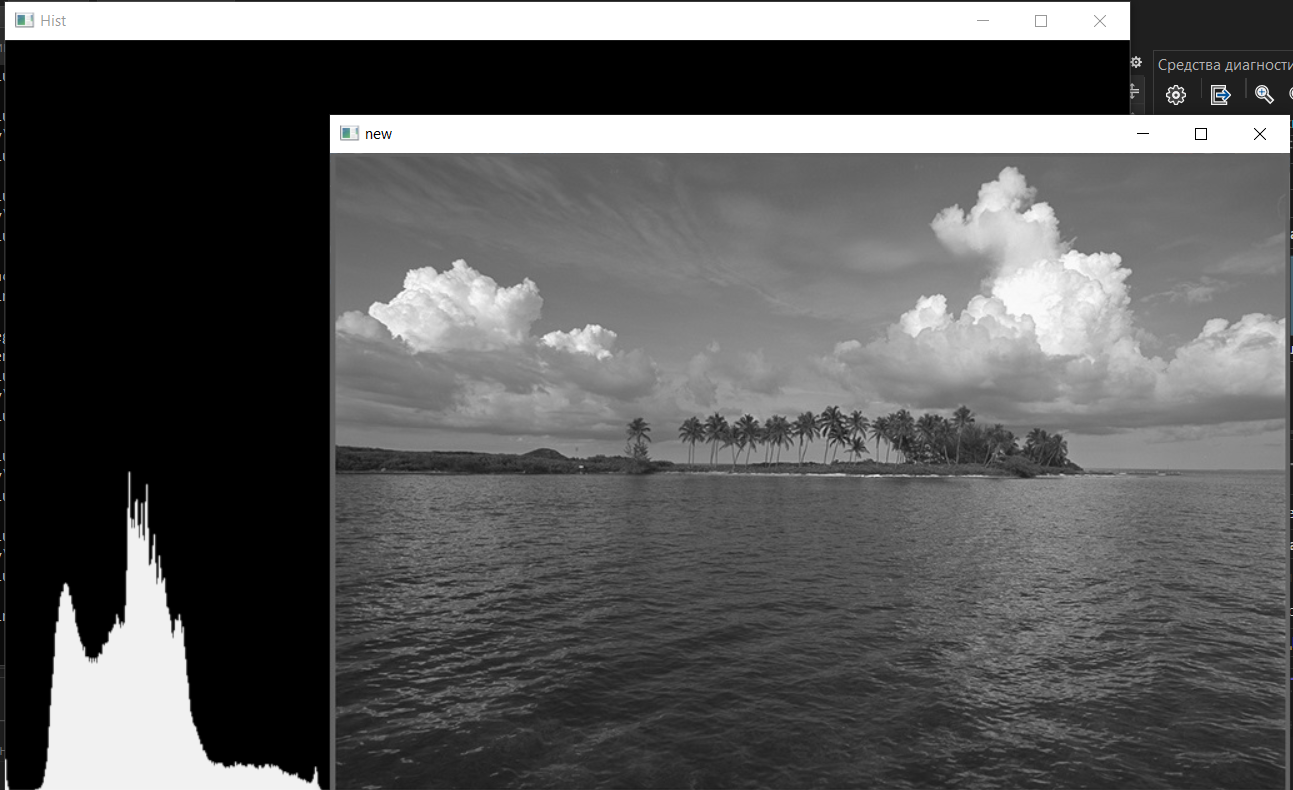


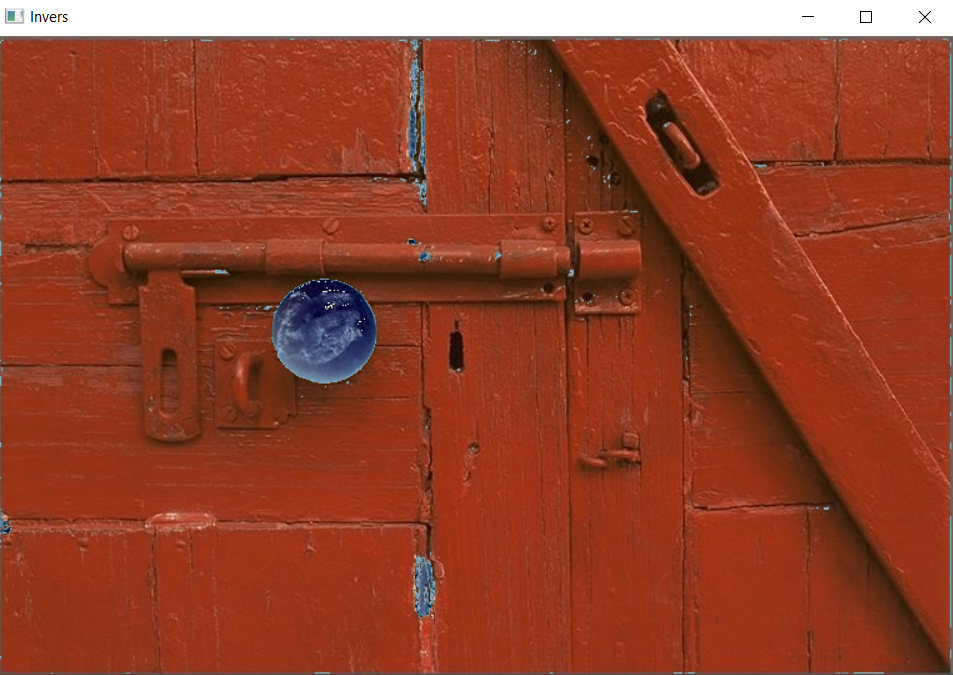












- **2.3. Текст программы**

#include <opencv2/opencv.hpp>

#include <iostream>

using namespace cv;

void post

#include <iostream>

#include <opencv2/core.hpp>

#include <opencv2/imgproc.hpp>

#include <opencv2/highgui.hpp>

#include<vector>

#include<algorithm>

#include<string>

using namespace std;

using namespace cv;

Mat img, tmpImg, tmpImg1;

int k = 0;

int s = 0;

void on\_mouse(int EVENT, int x, int y, int flags, void\* ustc)

{

switch (EVENT)

{

case EVENT\_LBUTTONDOWN:

{ k++;

break;

}

case EVENT\_MOUSEMOVE:

{if (k == 1) {

circle(tmpImg, Point(x, y), s, Scalar(255, 255, 255), FILLED, 0);

imshow("Paint\_img", tmpImg);

}

break; }

case EVENT\_LBUTTONUP: {

k--;

break; }

}

}

Mat img3(1000, 1000, CV\_8UC3, Scalar(255, 255, 255));

vector <int> b;

vector <int> b1;

void on\_mouse1(int EVENT, int x, int y, int flags, void\* ustc)

{

switch (EVENT)

{

case EVENT\_LBUTTONDOWN:

{

b.push\_back(x);

b.push\_back(y);

break;

}

case EVENT\_LBUTTONUP:

{

b1.push\_back(x);

b1.push\_back(y);

vector<Mat> channels{};

split(tmpImg1, channels);

vector<Mat> channels1{};

split(img3, channels1);

for (int i = b[b.size() - 2], i1 = 0; i < x; i++, i1++) {

for (int j = b[b.size() - 1], j1 = 0; j < y; j++, j1++) {

channels1.at(0).at<uchar>(j1, i1) = channels.at(0).at<uchar>(j, i);

channels1.at(1).at<uchar>(j1, i1) = channels.at(1).at<uchar>(j, i);

channels1.at(2).at<uchar>(j1, i1) = channels.at(2).at<uchar>(j, i);

channels.at(0).at<uchar>(j, i) = 255;

channels.at(1).at<uchar>(j, i) = 255;

channels.at(2).at<uchar>(j, i) = 255;

}

}

merge(channels, tmpImg1);

imshow("Paint\_img1", tmpImg1);

merge(channels1, img3);

break;

}

case EVENT\_RBUTTONDOWN: {

vector<Mat> channels{};

split(tmpImg1, channels);

vector<Mat> channels1{};

split(img3, channels1);

for (int i = x, i1 = 0; i < (x + b1[b1.size() - 2] - b[b.size() - 2]); i++, i1++) {

for (int j = y, j1 = 0; j < (y + b1[b1.size() - 1] - b[b.size() - 1]); j++, j1++) {

channels.at(0).at<uchar>(j, i) = channels1.at(0).at<uchar>(j1, i1);

channels.at(1).at<uchar>(j, i) = channels1.at(1).at<uchar>(j1, i1);

channels.at(2).at<uchar>(j, i) = channels1.at(2).at<uchar>(j1, i1);

}

}

merge(channels, tmpImg1);

imshow("Paint\_img1", tmpImg1);

break;

}

}

}

int main()

{

setlocale(LC\_ALL,"rus");

int n;

cout << "Введите номер фильтра" << endl;

cout << "0. Перемещение прямоугольной области" << endl << "1.Ластик" << endl << "2. Уменьшение масштаба всего изображения"<<endl;

cout << "3. Гамма-коррекция" << endl << "4. Нормализация гистограммы" << endl << "5.Пороговая сегментация" << endl;

cin >> n;

switch (n)

{

case 0:

{

img = imread("16.jpg");

tmpImg1 = img.clone();

imshow("Paint\_img1", tmpImg1);

while (true)

{

setMouseCallback("Paint\_img1", on\_mouse1, (void\*)(&tmpImg1)); // Вызов функции обратного вызова

waitKey(0);

}

destroyWindow("Paint\_img1");

break;

}

case 1:

{

cout << "Введите номер желаемого изображения(1-цветное,2-чб):" << endl;

int l;

cin >> l;

if(l==1)

img = imread("16(c).jpg");

else

img = imread("16(cb).jpg");

tmpImg = img.clone();

imshow("Paint\_img", tmpImg);

createTrackbar("Размер ластика", "Paint\_img", &s, 100);

while (true)

{

setMouseCallback("Paint\_img", on\_mouse, (void\*)(&tmpImg)); // Вызов функции обратного вызова

waitKey(0);

}

destroyWindow("Paint\_img");

break;

}

case 2:

{

cout << "Введите номер желаемого изображения(1-цветное,2-чб):" << endl;

int l;

cin >> l;

if (l == 1)

img = imread("16(c).jpg");

else

img = imread("16(cb).jpg");

Mat img1;

int k = 0;

namedWindow("Уменьшение размера",WINDOW\_AUTOSIZE);

createTrackbar("Кратность", "Уменьшение размера", &k, min(img.size().width, img.size().height));

while (true)

{

resize(img, img1, Size(img.size().width - k, img.size().height - k),0,0, INTER\_LINEAR);

if ((img.size().width - k) <= 0 or (img.size().height - k) <= 0)

break;

imshow("Уменьшение размера", img1);

waitKey(1);

}

break;

}

case 3:

{

cout << "Введите номер желаемого изображения(1-цветное,2-чб):" << endl;

int l;

cin >> l;

if (l == 1)

{

img = imread("16(c).jpg");

Mat img2 = img.clone();

namedWindow("Гамма-Коррекция");

vector<Mat> channels{};

split(img, channels);

int qr = 10, qg = 10, qb = 10;

createTrackbar("Синий", "Гамма-Коррекция", &qr, 20);

createTrackbar("Зелёный", "Гамма-Коррекция", &qg, 20);

createTrackbar("Красный", "Гамма-Коррекция", &qb, 20);

while (true)

{

double qrr = static\_cast<double>(qr) / 10;

double qgg = static\_cast<double>(qg) / 10;

double qbb = static\_cast<double>(qb) / 10;

vector<Mat> channels{};

split(img2, channels);

vector<Mat> channelss{};

split(img, channelss);

for (int i = 0; i < img.rows; i++)

{

for (int j = 0; j < img.cols; j++)

{

if (pow(channelss.at(0).at<uchar>(i, j), qrr) < 255)

{

channelss.at(0).at<uchar>(i, j) = pow(channels.at(0).at<uchar>(i, j), qrr);

}

else

channelss.at(0).at<uchar>(i, j) = 255;

if (pow(channelss.at(1).at<uchar>(i, j), qgg) < 255)

{

channelss.at(1).at<uchar>(i, j) = pow(channels.at(1).at<uchar>(i, j), qgg);

}

else

channelss.at(1).at<uchar>(i, j) = 255;

if (pow(channelss.at(2).at<uchar>(i, j), qrr) < 255)

{

channelss.at(2).at<uchar>(i, j) = pow(channels.at(2).at<uchar>(i, j), qbb);

}

else

channelss.at(2).at<uchar>(i, j) = 255;

}

}

merge(channelss, img);

imshow("Гамма-Коррекция", img);

waitKey(1);

}

}

else

{

img = imread("16(cb).jpg",0);

Mat img2 = img.clone();

namedWindow("Гамма-Коррекция");

int qr = 10;

createTrackbar("Гамма", "Гамма-Коррекция", &qr, 20);

while (true)

{

double qrr = static\_cast<double> (qr) / 10;

for (int i = 0; i < img.rows; i++)

{

for (int j = 0; j < img.cols; j++)

{

if (pow(img2.at<uchar>(i, j), qrr) < 255)

{

img.at<uchar>(i, j) = pow(img2.at<uchar>(i, j), qrr);

}

else

img.at<uchar>(i, j) = 255;

}

}

imshow("Гамма-Коррекция", img);

waitKey(1);

}

}

break;

}

case 4:

{

cout << "Введите номер желаемого изображения(1-цветное,2-чб):" << endl;

int l;

cin >> l;

if (l == 1)

{

img = imread("16(c).jpg");

Mat histR,histG,histB,hist,img2;

int channels[] = { 0 };

int histSize[] = { 256 };

float hr1[] = { 0,255 };

const float\* ranges[] = { hr1 };

vector<Mat> ch;

split(img, ch);

calcHist(&ch.at(2), 1, channels, Mat(), histR, 1, histSize, ranges, true, false);

calcHist(&ch.at(1), 1, channels, Mat(), histG, 1, histSize, ranges, true, false);

calcHist(&ch.at(0), 1, channels, Mat(), histB, 1, histSize, ranges, true, false);

Mat histimgR(600, 900, CV\_8UC3, Scalar(0, 0, 0));

Mat histimgG(600, 900, CV\_8UC3, Scalar(0, 0, 0));

Mat histimgB(600, 900, CV\_8UC3, Scalar(0, 0, 0));

normalize(histR, histR, 0, 255, NORM\_MINMAX, -1, Mat());

normalize(histG, histG, 0, 255, NORM\_MINMAX, -1, Mat());

normalize(histB, histB, 0, 255, NORM\_MINMAX, -1, Mat());

for (int i = 0; i < 256; i++)

{

line(histimgR, Point(i, 600), Point(i, 600 - cvRound(histR.at<float>(i))), Scalar(255, 255, 255), 1, LINE\_AA);

line(histimgG, Point(i, 600), Point(i, 600 - cvRound(histG.at<float>(i))), Scalar(255, 255, 255), 1, LINE\_AA);

line(histimgB, Point(i, 600), Point(i, 600 - cvRound(histB.at<float>(i))), Scalar(255, 255, 255), 1, LINE\_AA);

}

normalize(img, img, 0, 255, NORM\_MINMAX);

imshow("img", img);

imshow("HistR", histimgR);

imshow("HistG", histimgG);

imshow("HistB", histimgB);

waitKey(0);

}

else

{

img = imread("16(cb).jpg", IMREAD\_GRAYSCALE);

Mat img4 = img.clone();

Mat hist;

int channels[] = { 0 };

int histSize[] = { 256 };

float hr1[] = { 0,255 };

const float\* ranges[] = { hr1 };

calcHist(&img, 1, channels, Mat(), hist, 1, histSize, ranges, true, false);

Mat histimg(600, 900, CV\_8UC3, Scalar(0, 0, 0));

normalize(hist, hist, 0, 255, NORM\_MINMAX, -1, Mat());

for (int i = 0; i < 256; i++)

{

line(histimg, Point(i, 600), Point(i, 600 - cvRound(hist.at<float>(i))), Scalar(255, 255, 255), 1, LINE\_AA);

}

imshow("Hist", histimg);

normalize(img, img4, 0, 255, NORM\_MINMAX);

imshow("new", img4);

waitKey(0);

}

break;

}

case 5:

{

cout << "Введите номер желаемого изображения(1-цветное,2-чб):" << endl;

int l;

cin >> l;

if (l == 1)

img = imread("16(c).jpg");

else

img = imread("16(cb).jpg");

Mat img1;

Mat img2(Size(img.size()), CV\_8UC3, Scalar(0));

cvtColor(img, img1, COLOR\_RGB2YCrCb);

int y\_min = 0, y\_max = 0, Cr\_min = 0,Cr\_max = 0,Cb\_min = 0, Cb\_max=0;

namedWindow("Invers", WINDOW\_AUTOSIZE);

createTrackbar("y\_min", "Invers", &y\_min, 1000);

createTrackbar("y\_max", "Invers", &y\_max, 1000);

createTrackbar("Cr\_min", "Invers", &Cr\_min, 1000);

createTrackbar("Cr\_max", "Invers", &Cr\_max, 1000);

createTrackbar("Cb\_min", "Invers", &Cb\_min, 1000);

createTrackbar("Cb\_max", "Invers", &Cb\_max, 1000);

while (true)

{

Mat img1;

img1 = img.clone();

cvtColor(img1, img1, COLOR\_BGR2YCrCb);

vector<Mat> channels{};

split(img1, channels);

for (int i = 0; i < img1.rows; i++)

{

for (int j = 0; j < img1.cols; j++)

{

if (channels.at(0).at<uchar>(i, j) > y\_min && channels.at(0).at<uchar>(i, j) < y\_max &&

channels.at(1).at<uchar>(i, j) > Cr\_min && channels.at(1).at<uchar>(i, j) < Cr\_max &&

channels.at(2).at<uchar>(i, j) > Cb\_min && channels.at(2).at<uchar>(i, j) < Cb\_max)

{

channels.at(0).at<uchar>(i, j) = 255 - channels.at(0).at<uchar>(i, j);

channels.at(1).at<uchar>(i, j) = 255 - channels.at(1).at<uchar>(i, j);

channels.at(2).at<uchar>(i, j) = 255 - channels.at(2).at<uchar>(i, j);

}

}

}

merge(channels, img1);

cvtColor(img1, img1, cv::COLOR\_YCrCb2BGR);

imshow("Invers", img1);

waitKey(1);

}

}

}

}

**2.4. Эксплуатационные документы**

**2.4.1. Ведомость эксплуатационных документов**

|  |  |  |  |
| --- | --- | --- | --- |
| Обозначение | Наименование | Кол.  Экз | Местонахождение |
|  | Документы на программу |  |  |
| А.В.ХХХХХ-ХХ 31 ХХ-ХХ | Обработка изображений  Описание применения | 1 | Каф. ТВ |
| А.В.ХХХХХ-ХХ 34 ХХ-ХХ | Обработка изображений  Руководство оператора | 1 | Каф. ТВ |
| А.В.ХХХХХ-ХХ 35 ХХ-ХХ | Обработка изображений  Описание языка | 1 | Каф. ТВ |

**2.4.2. Описание применения**

2.4.2.1. Назначение программы назначение: обработка изображений, возможности ПО: создание изображения и обработка, ограничения: создание и обработка текстовых файлов.

2.4.2.2. Условия применения загрузка файлов в формате png, jpeg. Минимальный объем оперативной памяти ПК 4гб, четырехъядерный процессор, windows 10,11, место на жестком диске не менее 1 гб.

2.4.2.3. Описание задачи: изменить изображения применяя различные фильтры. Метод решения - использовать библиотеку opencv.

2.4.2.4. Входные и выходные данные: изображения.

**2.4.3. Руководство оператора**

2.4.3.1. Назначение программы назначение: обработка изображений, возможности ПО: создание изображения и обработка, ограничения: создание и обработка текстовых файлов.

2.4.3.2. Условия выполнения программы ПК с 4 гб ОЗУ, установленный visual studio, установленный c++ с библиотекой Open CV.

2.4.3.3. Выполнение программы последовательность действий оператора , обеспечивающих загрузку, запуск, выполение и завершение программы, описание функций, формата, возможных вариантов команд и ответы программы на эти команды

2.4.3.4. Сообщения оператору: при запуске программы представлен выбор фильтра и изображения для использования.В случае загрузки изображения не того формата компилятор выдаст ошибку открытия изображения.

**2.4.4 Описание языка**

2.4.4.1. Общие сведения предназначен для решения различных задач, основными особенностями являются: использование условных операторов, различные шаблоны и функции, возможность пользования различными библиотеками, с++ применяется для создания и обработки изображений, используется для работы с большими объемами данных.

2.4.4.2. Элементы языка: в программах используется базовый синтаксис с++ с применением библиотек Open CV.

2.4.4.3. Средства обмена данными: стандартные файлы ввода и вывода(#include <iostream>).

2.4.4.4. Встроенные элементы: подключены библиотеки(opencv, iostream, vector), библиотека opencv - открытая библиотека для работы с алгоритмами компьютерного зрения, машинным обучением и обработкой изображений, iostream - Заголовочный файл с классами, функциями и переменными для организации ввода-вывода в языке программирования C++,vector-библиотека для работы с векторами(специальными типами данных с++, включающих в себя возможность записи данных).

2.4.4.5. Средства отладки программы: Пошаговое перемещение по коду и просмотр значений, хранящихся в переменных, задавать контрольные значения для переменных , чтобы отслеживать изменение значений, изучать путь выполнения кода и тд.

**заключение**

В ходе выполнения курсовой работы было разработано программное обеспечение для редактирования изображений. Проведен анализ и отладка программного обеспечения, предназначенного для изменения свойств и параметров изображений. Была запущена программа, написанная на языке программирования С++ с применением фильтров и использовании библиотеки opencv.

**список использованных источников**

***Ниже представлены примеры библиографического описания, В качестве названия источника в примерах приводится вариант, в котором применяется то или иное библиографическое описание.***

1. Книга двух авторов/ О.Д. Суарес, Г.Б. Гарсия. : Издательство visilab, 2012. 206 с.

2. Книга двух авторов / А.Кэлер, Г. Брэдски. : Издательство ДМК, 2017. 827 с.

3. Сайт /https://docs.opencv.org/4.x/d9/df8/tutorial\_root.html