

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**
**НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**
Факультет информационных технологий
Кафедра параллельных вычислений

ОТЧЕТ

О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОЙ РАБОТЫ

«Преобразование изображения камеры и работа с USB-устройствами»

студента 2 курса, группы 20205

Муратова Максима Александровича

Направление 09.03.01 – «Информатика и вычислительная техника»

**Преподаватель:
Доцент
Власенко А. Ю.**

Новосибирск 2021

СОДЕРЖАНИЕ

ЦЕЛЬ.....	3
ЗАДАНИЕ.....	3
ОПИСАНИЕ РАБОТЫ.....	4
ЗАКЛЮЧЕНИЕ.....	7
Приложение 1. <i>main.cpp</i>	8
Приложение 2. <i>transform.h</i>	10
Приложение 3. <i>initial.cpp</i>	11
Приложение 4. <i>negative.cpp</i>	12
Приложение 5. <i>usb.cpp</i>	13
Приложение 6. <i>Makefile</i>	16
Приложение 7. <i>tester.sh</i>	17
Приложение 8. <i>initial-report.txt</i>	18
Приложение 9. <i>negative-report.txt</i>	19
Приложение 10. <i>usb-report.txt</i>	20
Приложение 11. <i>Источники</i>	21

ЦЕЛИ

- Научиться работать с OpenCV: получать изображение с камеры, обрабатывать его и находить частоту кадров (FPS)
- Научиться получать и обрабатывать данные о USB-устройствах через библиотеку libusb

ЗАДАНИЕ

1. Написать программу, которая бы выводила необработанное изображение на экран;
2. Написать программу, которая бы выводила негатив изображения на экран;
3. При помощи утилиты `std::chrono::system_clock::now()` из `<chrono>` измерять FPS каждые 20 кадров, найти среднее значение для обеих программ и сравнить их;
4. Получить следующие данные о USB-устройствах: ID класса, подкласса, устройства, его производителя, серийный номер, название устройства и производителя.

ОПИСАНИЕ РАБОТЫ

Как уже говорилось, для работы с изображением требуется две программы: для вывода изображения без обработки и обработкой. Для этого я написал 4 файла:

1. `main.cpp` – основная программа, служит для вывода изображения на экран (ПРИЛОЖЕНИЕ 1);
2. `transform.h` – заголовочный файл с функцией `transformImage()`, которое обрабатывает изображение. Файл используется для корректной связи между основной программой и реализациями функции преобразования изображения (ПРИЛОЖЕНИЕ 2);
3. `initial.cpp` – реализация отсутствия обработки для получения исходного изображения (ПРИЛОЖЕНИЕ 3);
4. `negative.cpp` – реализация наложения негатива на изображение (ПРИЛОЖЕНИЕ 4).

Для компиляции программ для работы как с изображениями, так и с USB, был написан файл конфигурации `Makefile` (ПРИЛОЖЕНИЕ 6), программы конкретно для работы с OpenCV собираются командой `$ make opencv`

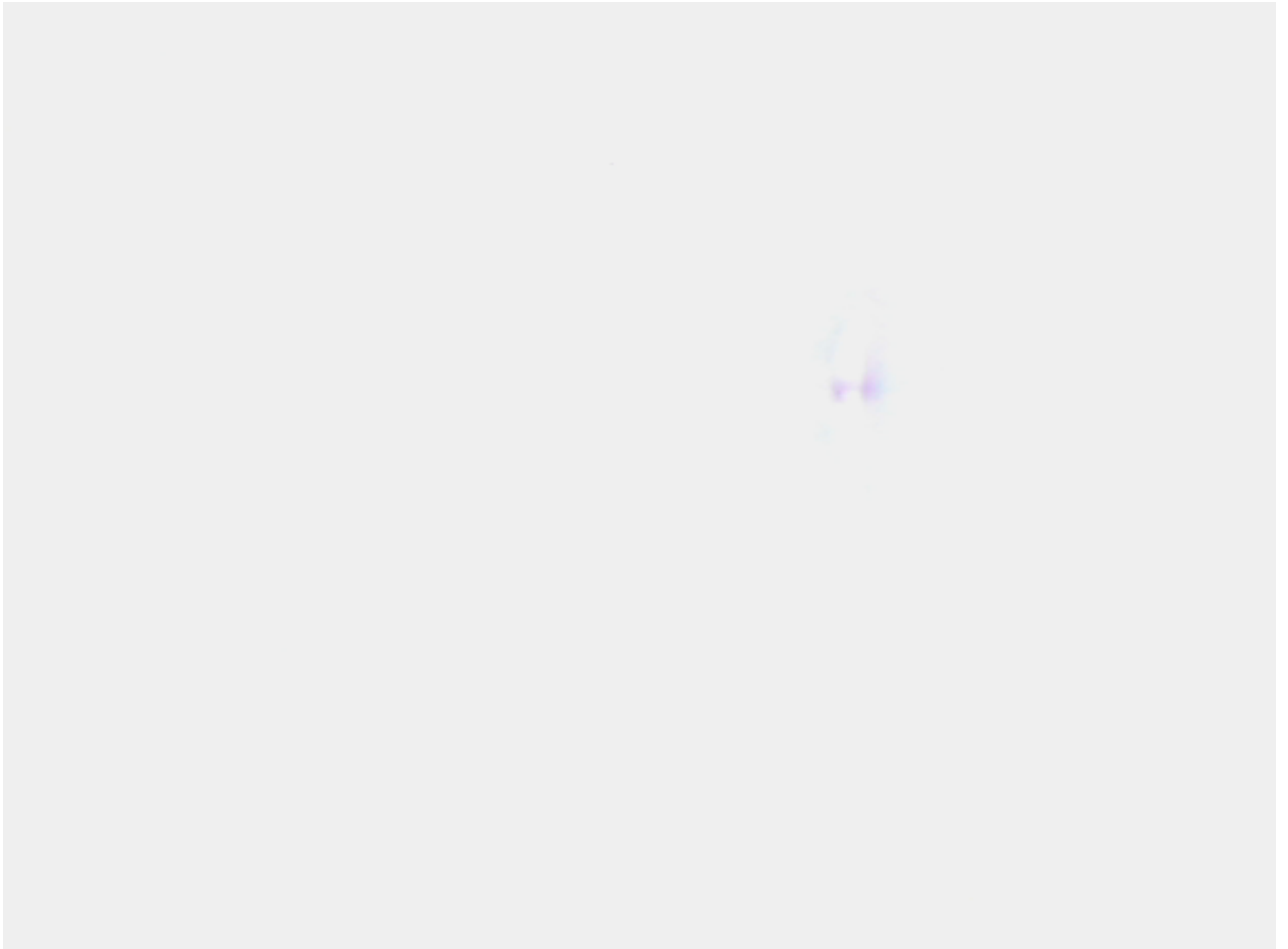
На выходе мы получим две программы: `initial` и `negative` для вывода необработанного изображения и изображения в негативе соответственно.

Для теста программы необходимо воспользоваться скриптом тестирования `tester.sh`: `$./tester.sh program`, где `program` – название программы – `initial` или `negative`, – без начального `./`

До:



После:



У необработанного изображения среднее FPS составило 1.86, у изображения в негативе – 2.25. Есть предположение, что FPS такое низкое из-за того, что параллельно шли другие процессы, потому что пробные тесты показывали FPS, равный 4.

Таким, среднее время на обработку одного изображения составило 0.45 и 0.54 секунд соответственно.

Теперь можно поработать с USB. Тут программа всего одна, поэтому нет необходимости применять элементы модульности, и вся программа содержится в файле `usb.cpp` (ПРИЛОЖЕНИЕ 5). Собрать данную программу можно при помощи того же файла конфигурации `Makefile`, командой

```
$ make usb
```

На выходе мы получили программу `usb`. Чтобы её запустить, следует ввести команду

```
# ./usb >usb-report.txt
```

Обратите внимание – программу нужно запускать с правами суперпользователя.

usb-report.txt – протокол работы данной программы
(ПРИЛОЖЕНИЕ 10)

Было обнаружено 5 устройств, вот их характеристики:

Номер	ID класса	ID устройства	ID производителя	Серийный номер
1	0x09	3	7531	0000:00:15.0
2	0xe0	2081	3034	00e04c000001
3	0xef	46551	1266	0x0001
4	0x00	9488	2362	Не найден
5	0x09	2	7531	0000:00:15.0

Немного поподробней о каждом устройстве:

1. Что это: порт USB 3.0 (xHCI Host Controller)
 1. Значение ID класса 0x09: порт
 2. Производитель: Linux 5.4.0-91-generic xhci-hcd
2. Что это: Bluetooth модуль (Bluetooth Radio)
 1. Значение ID класса 0xe0: беспроводной контроллер
 2. Производитель: Realtek
3. Что это: встроенная камера компьютера (EasyCamera)
 1. Значение ID класса 0xef: прочее
 2. Производитель: Chicony Electronics Co.,Ltd.
4. Что это: оптическая мышь (USB Optical Mouse)
 1. Значение ID класса 0x00: интерфейс
 2. Производитель: PixArt
5. То же, что и первое устройство, то есть второй USB порт

Ссылка на документации по OpenCV и libusb можно найти в ПРИЛОЖЕНИИ 11.

ЗАКЛЮЧЕНИЕ

1. Обработка изображения средствами библиотеки OpenCV идёт довольно медленно, из-за чего FPS был приблизительно равен 2.
2. Было обнаружено 5 USB-устройств: 2 USB-порта, Bluetooth модуль, встроенная камера и компьютерная мышь.

Приложение 1. *main.cpp* программа отображения изображения с камеры

```
#include <opencv2/highgui/highgui.hpp>
#include <iostream>
#include <chrono>
#include "transform.h"

int main(int argc, char *argv[]){
    CvCapture *capture = cvCreateCameraCapture(0);
    if (!capture) {
        std::cerr << "failed to capture an image\n";
        return 1;
    }

    auto begin = std::chrono::system_clock::now();
    for(int iter = 0;; iter++) {
        if(iter % 20 == 0)
            begin = std::chrono::system_clock::now();
        IplImage *frame = cvQueryFrame(capture);
        if(!frame) break;
        cv::Mat mtr = cv::cvarrToMat(frame);

        transformImage(mtr);

        imshow("test", mtr);
        char c = cvWaitKey(33);
        if(c == 27) break;

        if(iter % 20 == 19) {
            std::chrono::duration<double> dif =
                std::chrono::system_clock::now() - begin;
            std::cout << 20. / dif.count() << '\t' << dif.count() /
```



```
                20 << '\n';  
            }  
        }  
        cvReleaseCapture(&capture);  
        cvDestroyWindow("test");  
        return 0;  
    }
```

Приложение 2. *transform.h*
заголовочный файл, в котором объявлена функция преобразования
изображения

```
#ifndef TRANSFORM_H
#define TRANSFORM_H

#include <opencv2/highgui/highgui.hpp>

void transformImage(cv::Mat &mtr);

#endif //TRANSFORM_H
```

Приложение 3. *initial.cpp*
**реализация отсутствия преобразования изображения, чтобы
можно было вывести изображение без изменений**

```
#include <opencv2/core/core.hpp>

#include "transform.h"

void transformImage(cv::Mat &mtr) { }
```

Приложение 4. *negative.cpp*
преобразования изображения в негатив

```
#include <opencv2/core/core.hpp>
#include "transform.h"

void transformImage(cv::Mat &mtr) {
    cv::Mat neg = cv::Scalar(255, 255, 255) - mtr;
    mtr = std::move(neg);
}
```

Приложение 5. *usb.cpp*
программа для получения данных USB устройств

```
#include <iostream>
#include <libusb.h>
#include <stdio.h>
using namespace std;

void printdev(libusb_device *dev);

int main() {
    libusb_device **devs;
    libusb_context *ctx = NULL;
    int r;
    ssize_t cnt;
    ssize_t i;
    r = libusb_init(&ctx);
    if(r < 0) {
        fprintf(stderr,
            "Ошибка: инициализация не выполнена, код: %d.\n", r);
        return 1;
    }
    libusb_set_debug(ctx, 3);
    cnt = libusb_get_device_list(ctx, &devs);
    if(cnt < 0){
        fprintf(stderr,
            "Ошибка %d: список USB устройств не получен.\n", r);
        return 1;
    }
    printf("found devices: %ld\n", cnt);

    printf("=====\n");
    printf("class\tsubcl\tvendor\tproduct\tserial\tname\tvendor\n");
    printf("ID\tID\tID\tID\tnumber\n");
    for(i = 0; i < cnt; i++) {
```

```

        printdev(devs[i]);
    }

printf("=====\n");

    libusb_free_device_list(devs, 1);
    libusb_exit(ctx);
    return 0;
}

void printdev(libusb_device *dev) {
    libusb_device_descriptor desc;
    libusb_device_handle* hand = NULL;
    int r = libusb_get_device_descriptor(dev, &desc);
    if (r < 0) {
        fprintf(stderr, "Ошибка: дескриптор устройства не получен,
код: %d.\n", r);
        return;
    }

    r = libusb_open(dev, &hand);
    if(r < 0) {
        std::cerr << "error: handle not opened\n";
        return;
    }

    unsigned char serial_number[256], manufacturer[256],
product[256];

    int sn_ret = libusb_get_string_descriptor_ascii(hand,
desc.iSerialNumber, serial_number, 256);

    int pr_ret = libusb_get_string_descriptor_ascii(hand,
desc.iProduct, product, 256);

    int mn_ret = libusb_get_string_descriptor_ascii(hand,
desc.iManufacturer, manufacturer, 256);

    printf("0x%02x\t0x%02x\t%d\t%d\t%s\t%s\t%s\n",
        (int)desc.bDeviceClass,
        (int)desc.bDeviceSubClass,

```

```
        desc.idVendor,  
        desc.idProduct,  
        (sn_ret < 0 ? ((unsigned char *) "----") :  
serial_number),  
        (mn_ret < 0 ? ((unsigned char *) "----") : product),  
        (pr_ret < 0 ? ((unsigned char *) "----") : manufacturer)  
    );  
    libusb_close(hand);  
}
```

Приложение 6. Makefile

Файл сборки программы

```
CC=g++
CFLAGS=-std=c++11 -Wall

all: usb opencv

opencv: negative initial

usb: usb.cpp
    $(CC) $(CFLAGS) -o usb -I/usr/include/libusb-1.0 usb.cpp -lusb-1.0

negative: main.o negative.o
    $(CC) main.o negative.o -o negative -lopencv_core -lopencv_highgui

initial: main.o initial.o
    $(CC) main.o initial.o -o initial -lopencv_core -lopencv_highgui

main.o: main.cpp
    $(CC) -c $(CFLAGS) main.cpp -o main.o

negative.o: negative.cpp
    $(CC) -c $(CFLAGS) negative.cpp -o negative.o

initial.o: initial.cpp
    $(CC) -c $(CFLAGS) initial.cpp -o initial.o
```


Приложение 7. *tester.sh*
скрипт для работы с программой с OpenCV

```
#!/bin/bash

if [[ $# -ne 1 ]]; then
    echo "count of arguments must be equal to 1" >2
    exit
fi

echo "making the programs"
make opencv
if [[ $? -ne 0 ]]; then
    echo "error: build failed" >2
    exit
fi
echo "programs are ready"

out="$1-report.txt"
echo -e "FPS\tsecs per one pic" >$out

"./$1" | tee -a $out
```

Приложение 8. *initial.txt*

Протокол замера FPS при отсутствии обработки изображения

FPS	Seconds per picture
1.81049	0.552337
1.91198	0.523018
2.24047	0.446335
1.51801	0.658757
1.99291	0.501779
2.22956	0.448518
2.34286	0.426829
1.38221	0.723478
1.50867	0.662837
2.11089	0.473733
1.3973	0.715665

Приложение 9. *negative-report.txt*

Протокол замера FPS при преобразования изображения в негатив

FPS	Seconds per picture
2.21102	0.45228
1.7748	0.563444
2.04037	0.490106
2.6633	0.375474
2.62606	0.380799
3.0443	0.328483
3.54558	0.282042
3.82799	0.261234
2.9641	0.337371
1.40729	0.710588
1.9061	0.524632
1.70855	0.585291
1.78274	0.560935
2.52453	0.396113
2.52766	0.395623
2.28101	0.438402
1.89129	0.52874
1.7412	0.574316
1.28528	0.77804
1.50203	0.665765
1.94046	0.51534

Приложение 10. *usb-report.txt*
Протокол работы с USB устройствами

found devices: 5

```
=====
class subclass vendor    product    serial    name vendor
ID      ID          ID      ID          number
0x09  0x00 7531 3      0000:00:15.0  xHCI Host Controller
Linux 5.4.0-91-generic xhci-hcd
0xe0  0x01 3034 2081 00e04c000001  Bluetooth Radio
Realtek
0xef  0x02 1266 46551      0x0001  EasyCamera    Chicony
Electronics Co.,Ltd.
0x00  0x00 2362 9488 ---- USB Optical Mouse  PixArt
0x09  0x00 7531 2      0000:00:15.0  xHCI Host Controller
Linux 5.4.0-91-generic xhci-hcd
=====
```

Приложение 11. Источники

- Сайт OpenCV:
<https://opencv.org/>
- Документация libusb:
https://libusb.sourceforge.io/api-1.0/libusb_api.html