**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 ЛАБОРАТОРНОЙ РАБОТЫ № 5**

«ВЫСОКОУРОВНЕВАЯ РАБОТА С ПЕРИФЕРИЙНЫМИ  
УСТРОЙСТВАМИ»

студента (ки) \_\_\_\_2\_\_\_\_\_ курса, \_\_21212\_\_ группы

**Олимпиева Юрия Юрьевича**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

**А. А. Артюхов**

2022 год

**Цели**

1. Ознакомиться с программированием периферийных устройств на примере ввода данных с web-камеры с использованием библиотеки OpenCV. **Задание**

1. Реализовать программу с использованием OpenCV, которая получает поток видеоданных с камеры и выводит его на экран.

2. Выполнить произвольное преобразование изображения.

3. Измерить количество кадров, обрабатываемое программой в секунду. Оценить долю времени, затрачиваемого процессором на обработку (ввод, преобразование, показ) видеоданных, получаемых с камеры.

4. Составить отчет по лабораторной работе.

**Описание работы**

* Подключил библиотеку openCV к проекту через Cmake файл.
* Написал программу для получения видео с встроенной web-камеры и вывода видео на экран (приложение 1).
* Реализовал преобразование Лапласа для выводимого видео (приложение 1).
* Измерил посредством библиотеки chrono количество кадров, обрабатываемое программой в секунду.
* Составил отчёт о выполненной работе.

**Заключение**

Изучил основы работы с библиотекой openCV, познакомился с программированием периферийных устройств.

**Приложение 1**

#include <opencv2/opencv.hpp>

#include <opencv2/highgui/highgui.hpp>

#include <iostream>

#include <ctime>

#include <chrono>

using namespace cv;

int main(int argc, char \*\*argv) {

Mat frame;

VideoCapture cap;

cap.open(0);

std::chrono::time\_point<std::chrono::system\_clock> begin, end;

begin = std::chrono::system\_clock::now();

while(cap.isOpened() && cap.read(frame)) {

// if escape(‘Esc’) was pressed

if (waitKey(1)==27) break;

// Frame processing

GaussianBlur( frame, frame, Size(3, 3), 0, 0, BORDER\_DEFAULT );

Laplacian(frame, frame, CV\_8U, 3, 1, BORDER\_DEFAULT);

// Calc time from prev frame to current

end = std::chrono::system\_clock::now();

std::chrono::duration<double> elapsed\_time = (end - begin);

std::string FPSInfo = "FPS: " + std::to\_string(1.0 /

elapsed\_time.count());

begin = end;

// Put FPS data on frame

putText(frame,

FPSInfo,

Point(frame.cols / 10, frame.rows / 10),

FONT\_HERSHEY\_COMPLEX,

0.5 ,

Scalar(255,255,255));

// load frame to video stream

imshow("lab5", frame);

}

cap.release();

destroyAllWindows();

}

**CmakeLists.txt**

cmake\_minimum\_required(VERSION 3.14)

project( lab5 )

find\_package( OpenCV REQUIRED )

include\_directories( ${OpenCV\_INCLUDE\_DIRS} )

add\_executable( lab5 lab5.cpp )

target\_link\_libraries(lab5 ${OpenCV\_LIBS} )

**Build on Linux Ubuntu:**

cmake -S . && make