Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Дисциплина: Интерфейсы и устройства вычислительных машин

Лабораторная работа №4

«Веб-камера»

Выполнил: Проверил:

Ст. гр. 150504 Селезнев А. И.

Горбачевский К. В.

Минск 2023

1. Постановка задачи

Вывести информацию об установленной веб-камере.

Осуществить захват изображения (фото) с последующим  сохранением в файл.

По вариантам (вариант 1):

- предусмотреть скрытый вариант фотонаблюдения, когда на мониторе  и на панели задач не отображается информация о Вашем работающем  приложении.

1. Краткие теоретические сведения

Веб-камера – сканер, выполняющий мгновенное считывание оригинального сигнала (в данном случае изображения) для выполнения различных действий над изображением.

Состав: объектив, светофильтр, матрица, схемы цифровой обработки.

Дисплей – основное устройство вывода информации, с помощью которого осуществляется интерфейс «человек-машина».

Объектив – оптическое устройство, формирующее поток лучей, направленный на матрицу.

Основные характеристики объектива:

- фокусное расстояние (угол обзора объектива);

- светосила (уровень ослабления объективом светового потока);

- наличие автофокуса.

Матрица – это «сердце» веб-камеры, преобразующее потоки света, поступающие из объектива, в цифровые сигналы. Все цифровые матрицы делятся на два типа: ПЗС и КМОП.

ПЗС матрицы (или CCD) – дороже в производстве, однако обладают лучшей цветопередачей, динамическим диапазоном и низким уровнем «шума». Основная область применения ПЗС – матриц это медицинское оборудование, реже зеркальные фотокамеры. В веб-камерах встречаются очень редко.

Матрицы КМОП (CMOS) завоевали более 90% мирового рынка. Преимущества КМОП-технологии: низкое энергопотребление в статическом состоянии, сразу выдаёт цифровой сигнал, который не требует дополнительного преобразования, дешевизна производства.

Принцип работы матрицы представлен тремя этапами: до начала записи выполняется сигнал сброса; в ходе съёмки происходит накопление фотодиодов; при считывании предоставляется возможность выбора уровня напряжения на конденсаторе.

Методы получения цветного изображения.

1) Аддитивные метод. Суть метода в сложении цветов с учётом особенности зрения человека относительно трёх основных цветов: красного, зелёного, синего. Цвета получаются за счёт слияния цветов.

2) Субтрактивный метод. Цвета получаются за счёт вычитания цветов. Метод воспроизводит цвета дополнительные к основным: жёлтый, пурпурный и голубой.

1. Код программы

#include <format>

#include <mutex>

#include <thread>

#include <windows.h>

#include <cstdio>

#include <iostream>

#include <setupapi.h>

#include <devguid.h>

#include <direct.h>

#include <opencv2\opencv.hpp>

#include <Windows.h>

#include <winuser.h>

#pragma comment(lib, "setupapi.lib")

using namespace cv;

using namespace std;

#define EXT\_KEY VK\_MENU

#define CAPTURE 0x43

#define EXIT 0x51

#define INFO 0x58

void printCameraInfo()

{

SP\_DEVINFO\_DATA DeviceInfoData;

ZeroMemory(&DeviceInfoData, sizeof(SP\_DEVINFO\_DATA));

DeviceInfoData.cbSize = sizeof(SP\_DEVINFO\_DATA);

HDEVINFO DeviceInfoSet = SetupDiGetClassDevs(&GUID\_DEVCLASS\_CAMERA, L"USB", NULL, DIGCF\_PRESENT);

if (DeviceInfoSet == INVALID\_HANDLE\_VALUE) {

wcout << L"Error: Unable to get device information set." << endl;

return;

}

DeviceInfoData.cbSize = sizeof(SP\_DEVINFO\_DATA);

if (!SetupDiEnumDeviceInfo(DeviceInfoSet, 0, &DeviceInfoData))

{

wcout << L"Error: Unable to enumerate device information." << endl;

SetupDiDestroyDeviceInfoList(DeviceInfoSet);

return;

}

const int PropertyBufferSize = 1024;

wchar\_t deviceID[PropertyBufferSize], deviceName[PropertyBufferSize], companyName[PropertyBufferSize];

ZeroMemory(&deviceID, sizeof(deviceID));

ZeroMemory(&deviceName, sizeof(deviceName));

ZeroMemory(&companyName, sizeof(companyName));

if (!SetupDiGetDeviceInstanceId(DeviceInfoSet, &DeviceInfoData, deviceID, sizeof(deviceID), NULL))

{

wcout << L"Error: Unable to get device instance ID." << endl;

wcout << L"Error code: " << GetLastError() << endl;

SetupDiDestroyDeviceInfoList(DeviceInfoSet);

return;

}

if (!SetupDiGetDeviceRegistryProperty(DeviceInfoSet, &DeviceInfoData, SPDRP\_DEVICEDESC, NULL, (PBYTE)deviceName, sizeof(deviceName), NULL))

{

wcout << L"Error: Unable to get device description." << endl;

SetupDiDestroyDeviceInfoList(DeviceInfoSet);

return;

}

if (!SetupDiGetDeviceRegistryProperty(DeviceInfoSet, &DeviceInfoData, SPDRP\_MFG, NULL, (PBYTE)companyName, sizeof(companyName), NULL))

{

wcout << L"Error: Unable to get manufacturer information." << endl;

SetupDiDestroyDeviceInfoList(DeviceInfoSet);

return;

}

wstring venAndDevId(deviceID);

wcout << L"Name: " << deviceName << endl;

wcout << L"Manufacture: " << companyName << endl;

wcout << L"Vendor ID: " << venAndDevId.substr(8, 4).c\_str() << endl;

wcout << L"Device ID: " << venAndDevId.substr(17, 4).c\_str() << endl << endl;

SetupDiDestroyDeviceInfoList(DeviceInfoSet);

}

void capture()

{

Mat frame;

SYSTEMTIME time;

try

{

VideoCapture cap(0);

if (cap.isOpened())

{

cap.set(CAP\_PROP\_FRAME\_WIDTH, 1920);

cap.set(CAP\_PROP\_FRAME\_HEIGHT, 1080);

cap.read(frame);

GetSystemTime(&time);

}

}

catch (...) {}

if (!frame.empty())

{

string filePath = "D:/Education/BGUIR\_\_LABS/5\_term/ИиУВМ/lab4/photos/";

string fileName = "camera-" + std::to\_string(time.wDay) + std::to\_string(time.wMinute) + std::to\_string(time.wMilliseconds) + ".png";

imwrite(filePath + fileName, frame);

cout << "Photo saved successfully in " << filePath << endl << endl;

}

}

bool shift = false;

bool ctrl = false;

void GetKEY()

{

if (GetAsyncKeyState(0x10)) shift = true;

else shift = false;

if (GetAsyncKeyState(0x11)) ctrl = true;

else ctrl = false;

}

int main() {

setlocale(LC\_ALL, "Russian");

HWND hWnd = GetConsoleWindow();

bool isSecretMode = false;

char answer = 'n';

cout << "Enter to secret mode (y/n): ";

cin >> answer;

if (answer == 'y') {

isSecretMode = true;

ShowWindow(hWnd, SW\_HIDE);

}

if (!isSecretMode) {

cout << "\n\nPress shift to make photo!\n";

cout << "Press ctrl to get camera info!\n\n";

}

while (1) {

Sleep(100);

GetKEY();

if (shift) {

if(isSecretMode) ShowWindow(hWnd, SW\_HIDE);

capture();

}

if (ctrl) {

if (isSecretMode) ShowWindow(hWnd, SW\_SHOW);

printCameraInfo();

}

}

return 0;

}

1. **Тестирование программы**

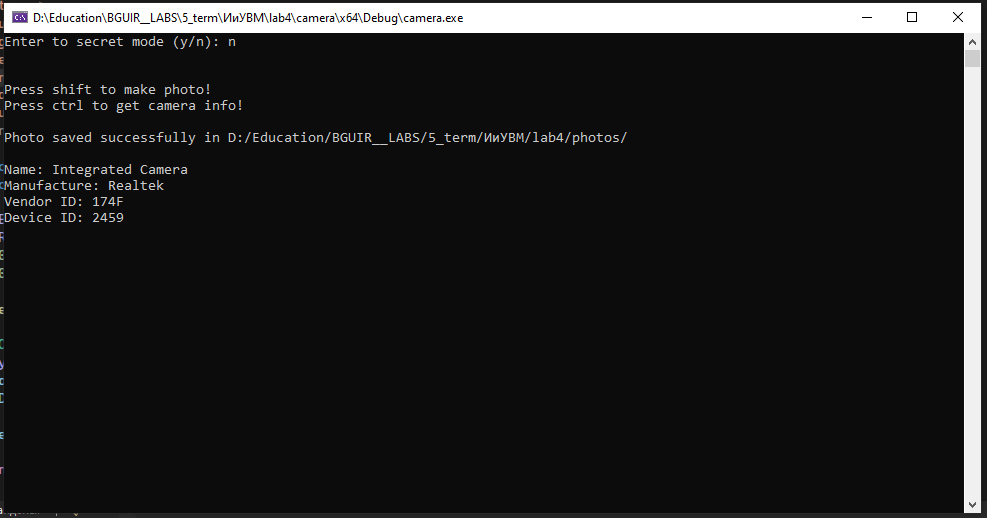


Рисунок 4.1 – Результат работы программы

1. **Вывод**

В ходе выполнения данной лабораторной работы были изучены теоретические основы работы веб-камеры, возможности работы с ней в операционной системе Windows 10 и способы доступа к информации о камере.

Программа написана для операционной системы Windows 10, использованные инструменты – язык программирования C++, Microsoft Visual Studio 2022, библиотека для работы с камерой “opencv”.