



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий
Кафедра Информатики и информационных технологий

направление подготовки
09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 1

Дисциплина: «Распознавание образов в информационных и автоматизированных системах копия 1»

Тема: «Использование библиотеки OpenCV в среде разработки Visual Studio»

Выполнил: студент группы 211-723

Сергеев Станислав Олегович

Дата, подпись _____
(Дата) (Подпись)

Проверил: _____
(Фамилия И.О., степень, звание) **(Оценка)**

Дата, подпись _____
(Дата) (Подпись)

Замечания: _____

Москва

2022

Использование библиотеки OpenCV в среде разработки Visual Studio.

Цель:

Целью данной работы является проверка остаточных навыков программирования на языке высокого уровня C#, а также навыков использования среды разработки Microsoft Visual Studio.

Постановка задачи:

Необходимо разработать приложение WindowsForms, способное:

1. осуществлять выбор и загрузку изображения;
2. отображать изображение на экране;
3. обрабатывать изображение при помощи эффекта Cell Shading.

Использование библиотеки OpenCV в среде разработки Visual Studio

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Structure;
using Emgu.CV.Util;
```

```
namespace _1
{
```

```
    public partial class Form1 : Form
    {
```

```
        private VideoCapture capture;
        private Image<Bgr, byte> sourceImage; //глобальная переменная
        double _cannyThreshold = 80.0, _cannyThresholdLinking = 50.0;
        int porog = 50;
        double frameCount;
```

```
        public Form1()
```

```
        {
            InitializeComponent();
        }
```

```
        public void Filtratsiya()
```

```
        {
            Image<Gray, byte> grayImage = sourceImage.Convert<Gray, byte>();
            var tempImage = grayImage.PyrDown();
            var destImage = tempImage.PyrUp();
            double cannyThreshold = _cannyThreshold;
            double cannyThresholdLinking = _cannyThresholdLinking;
            Image<Gray, byte> cannyEdges = destImage.Canny(cannyThreshold, cannyThresholdLinking);
            var cannyEdgesBgr = cannyEdges.Convert<Bgr, byte>();
```

```

var resultImage = sourceImage.Sub(cannyEdgesBgr); // попиксельное вычитание
imageBox2.Image = resultImage.Resize(400, 400, Inter.Linear);

for (int channel = 0; channel < resultImage.NumberOfChannels; channel++) //обход по каналам
    for (int x = 0; x < resultImage.Width; x++)
        for (int y = 0; y < resultImage.Height; y++) // обход по пикселям
        {
            // получение цвета пикселя
            byte color = resultImage.Data[y, x, channel];
            if (color <= porog)
                color = 0;
            else if (color <= porog + porog)
                color = 25;
            else if (color <= porog + porog*2)
                color = 180;
            else if (color <= porog + porog*3)
                color = 210;
            else
                color = 255;
            resultImage.Data[y, x, channel] = color; // изменение цвета пикселя
        }
imageBox3.Image = resultImage.Resize(400, 400, Inter.Linear);
}

private void imageBox1_Click_1(object sender, EventArgs e)
{

}

private void Form1_Load(object sender, EventArgs e)
{

}

private void button1_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    var result = openFileDialog.ShowDialog(); // открытие диалога выбора файла
    if (result == DialogResult.OK) // открытие выбранного файла
    {
        string fileName = openFileDialog.FileName;
        sourceImage = new Image<Bgr, byte>(fileName);
    }
    imageBox1.Image = sourceImage.Resize(400, 400, Inter.Linear);
}

private void imageBox2_Click(object sender, EventArgs e)
{

}

private void textBox1_TextChanged(object sender, EventArgs e)
{

}

private void textBox3_TextChanged(object sender, EventArgs e)
{

}

private void button2_Click(object sender, EventArgs e)
{
    _cannyThreshold = 160.0;
}

private void button4_Click(object sender, EventArgs e)

```

```
{
    Filtratsiya();
}

private void button3_Click(object sender, EventArgs e)
{
    _cannyThresholdLinking = 100.0;
}

private void button5_Click(object sender, EventArgs e)
{
    _cannyThreshold = 120.0;
}

private void button7_Click(object sender, EventArgs e)
{
    _cannyThreshold = 80.0;
}

private void button9_Click(object sender, EventArgs e)
{
    _cannyThreshold = 40.0;
}

private void button11_Click(object sender, EventArgs e)
{
    _cannyThreshold = 0.0;
}

private void button6_Click(object sender, EventArgs e)
{
    _cannyThresholdLinking = 75.0;
}

private void button8_Click(object sender, EventArgs e)
{
    _cannyThresholdLinking = 50.0;
}

private void button10_Click(object sender, EventArgs e)
{
    _cannyThresholdLinking = 25.0;
}

private void button12_Click(object sender, EventArgs e)
{
    _cannyThresholdLinking = 0.0;
}

private void button13_Click(object sender, EventArgs e)
{
    porog = 100;
}

private void button14_Click(object sender, EventArgs e)
{
    porog = 75;
}

private void button15_Click(object sender, EventArgs e)
{
    porog = 50;
}

private void button16_Click(object sender, EventArgs e)
{
    porog = 25;
}
```

```

}

private void button17_Click(object sender, EventArgs e)
{
    порог = 0;
}

private void ProcessFrame(object sender, EventArgs e)
{
    var frame = new Mat();
    capture.Retrieve(frame); // получение текущего кадра
    Image<Bgr, byte> image = frame.ToImage<Bgr, byte>();
    imageBox1.Image = image.Resize(400, 400, Inter.Linear);
}

private void button18_Click(object sender, EventArgs e)
{
    capture = new VideoCapture();
    capture.ImageGrabbed += ProcessFrame;
    capture.Start();

    button18.Visible = false;
    button18.Enabled = false;
    button20.Visible = true;
    button20.Enabled = true;
}

private void button20_Click(object sender, EventArgs e)
{
    capture.Stop();
    button18.Visible = true;
    button18.Enabled = true;
    button20.Visible = false;
    button20.Enabled = false;
}

private void ProcessFrameForVideo(object sender, EventArgs e)
{
    if (frameCount == 0)
    {
        capture.Stop();
    }
    else
    {
        frameCount--;
        var frame = new Mat();
        capture.Retrieve(frame);
        Image<Bgr, byte> image = frame.ToImage<Bgr, byte>();
        imageBox1.Image = image.Resize(400, 400, Inter.Linear);

        Image<Gray, byte> grayImage = image.Convert<Gray, byte>();
        var tempImage = grayImage.PyrDown();
        var destImage = tempImage.PyrUp();
        double cannyThreshold = _cannyThreshold;
        double cannyThresholdLinking = _cannyThresholdLinking;
        Image<Gray, byte> cannyEdges = destImage.Canny(cannyThreshold, cannyThresholdLinking);
        var cannyEdgesBgr = cannyEdges.Convert<Bgr, byte>();
        var resultImage = image.Sub(cannyEdgesBgr); // попиксельное вычитание
        imageBox2.Image = resultImage.Resize(400, 400, Inter.Linear);

        for (int channel = 0; channel < resultImage.NumberOfChannels; channel++) //обход по каналам
            for (int x = 0; x < resultImage.Width; x++)
                for (int y = 0; y < resultImage.Height; y++) // обход по пикселям
                {
                    // получение цвета пикселя
                    byte color = resultImage.Data[y, x, channel];
                }
            }
        }
    }

```

```

        if (color <= porog)
            color = 0;
        else if (color <= porog + porog)
            color = 25;
        else if (color <= porog + porog * 2)
            color = 180;
        else if (color <= porog + porog * 3)
            color = 210;
        else
            color = 255;
        resultImage.Data[y, x, channel] = color; // изменение цвета пикселя
    }
    imageBox3.Image = resultImage.Resize(400, 400, Inter.Linear);
}

}

private void button19_Click(object sender, EventArgs e)
{
    button4.Visible = false;
    button4.Enabled = false;
    button19.Visible = false;
    button19.Enabled = false;
    button21.Visible = true;
    button21.Enabled = true;
    button22.Visible = true;
    button22.Enabled = true;
    OpenFileDialog openFileDialog = new OpenFileDialog();
    var result = openFileDialog.ShowDialog(); // открытие диалога выбора файла
    if (result == DialogResult.OK) // открытие выбранного файла
    {
        string fileName = openFileDialog.FileName;
        capture = new VideoCapture(fileName);
    }
    frameCount = capture.GetCaptureProperty(CapProp.FrameCount);

    capture.ImageGrabbed += ProcessFrameForVideo;
    capture.Start();
}

private void button21_Click(object sender, EventArgs e)
{
    capture.Stop();
    button22.Visible = false;
    button22.Enabled = false;
    button19.Visible = true;
    button19.Enabled = true;
    button21.Visible = false;
    button21.Enabled = false;
    button4.Visible = true;
    button4.Enabled = true;
}

private void button22_Click(object sender, EventArgs e)
{
}

private void textBox2_TextChanged(object sender, EventArgs e)
{
}

}

```

}

Form1

Фильтр Кэни

Пороговый фильтр

Загрузить изображение

Веб-камера

Загрузить видео

cannyThreshold	cannyThresholdLinking	Порог
160	100	100
120	75	75
80	50	50
40	25	25
0	0	0

Фильтрация

Изображение:

Form1

Фильтр Кэни

Пороговый фильтр

Загрузить изображение

Веб-камера

Загрузить видео

cannyThreshold	cannyThresholdLinking	Порог
160	100	100
120	75	75
80	50	50
40	25	25
0	0	0

Фильтрация

Form1

Фильтр Кэни

Пороговый фильтр

Загрузить изображение

Веб-камера

Загрузить видео

cannyThreshold	cannyThresholdLinking	Порог
160	100	100
120	75	75
80	50	50
40	25	25
0	0	0

Фильтрация

Веб-камера:

Form1

Загрузить изображение

Прекратить

Загрузить видео

Фильтр Кэнни

Пороговый фильтр

cannyThreshold

160

120

80

40

0

cannyThresholdLinking

100

75

50

25

0

Порог

100

75

50

25

0

Фильтрация

Загрузить изображение

Прекратить

Загрузить видео

Фильтр Кэнни

Пороговый фильтр

cannyThreshold

160

120

80

40

0

cannyThresholdLinking

100

75

50

25

0

Порог

100

75

50

25

0

Фильтрация

Видео:

Form1

Загрузить изображение

Веб-камера

Стоп

Фильтр Кэнни

Пороговый фильтр

cannyThreshold

160

120

80

40

0

cannyThresholdLinking

100

75

50

25

0

Порог

100

75

50

25

0

Фильтрация

Загрузить изображение

Веб-камера

Стоп

Фильтр Кэнни

Пороговый фильтр

cannyThreshold

160

120

80

40

0

cannyThresholdLinking

100

75

50

25

0

Порог

100

75

50

25

0

Фильтрация

Form1

Загрузить изображение

Веб-камера

Стоп

Фильтр Кэнни

Пороговый фильтр

cannyThreshold

160

120

80

40

0

cannyThresholdLinking

100

75

50

25

0

Порог

100

75

50

25

0

Фильтрация

Загрузить изображение

Веб-камера

Стоп

Фильтр Кэнни

Пороговый фильтр

cannyThreshold

160

120

80

40

0

cannyThresholdLinking

100

75

50

25

0

Порог

100

75

50

25

0

Фильтрация