Государственное образовательное учреждение высшего профессионального образования

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | Информатика и системы управления |
| КАФЕДРА | Информационная безопасность |

**Домашнее Задание №1**

**Простая программа «компьютерного зрения»**

**Задание №9**

**по курсу «Алгоритмические языки программирования»**

Выполнил: Валитов Ришат, гр. ИУ8-22

Проверил: Быков А.Ю.

Москва, 2015

**Цель работы:**

Цель работы состоит в овладении навыками разработки программ на языке С#. Создать простую программу «компьютерного зрения». Программа должна отслеживать в видеопотоке координаты определенного объекта (например, человеческое лицо) и в зависимости от его координат выполнять различные действия (например, отправить e-mail, открыть notepad, открыть в браузере gmail.com, сохранить в файл кадр видеопотока).

**Условие задачи:**

Аудиоплеер с функцией управления жестами. Жесты осуществляются с помощью маркера определенного цвета. (я использовал оранжевый)

В планах следующие жесты:

1. Сверху вниз/ снизу вверх - громкость убавить/прибавить;
2. Справа налево/слева направо - следующая/предыдущая композиция;
3. С середины направо/налево - Пауза/Воспроизведение;

**Ход работы:**

1. Захват видео с веб-камеры(выполнено).
2. Захват изображения не только со встроенной веб-камерой, но и подключенной(выполнено).
3. Разбиение изображения на потоки, фильтрация по цвету(выполнено).
4. Нахождение объекта с помощью метода центра масс(выполнено).
5. Разработка метода анализа движений(выполнено).

подошел метод минимальных квадратов для линейных функций.

1. Реализация многопоточности(выполнено).
2. Аудиоплеер MP3(выполнено).
3. Связать методы анализа с жестами (выполнено).
4. Разработать интуитивный интерфейс(выполнено).

**Захват видео с веб-камеры:**

Для работы с видеопотоком с внешних устройств я использовал технологию OpenCV.

Хочу немного рассказать про технологию OpenCV и её применение на языке программирования C#.   
  
 OpenCV (англ. Open Source Computer Vision Library, библиотека компьютерного зрения с открытым исходным кодом) — библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++, также разрабатывается для Python, Java, Ruby, Matlab, Lua и других языков. Может свободно использоваться в академических и коммерческих целях — распространяется в условиях лицензии BSD.

Я использовал эту технологию, т.к. она не имеет аналогов с близким к ней функционалом и удобством использования.

EmguCV является кросс-платформенным .Net дополнением для библиотеки OpenCV для обработки изображений. Разработано для работы с .NET совместимыми языками, такими как C #, VB, VC ++, IronPython и т.д., может быть использовано в Visual Studio, Xamarin, работает с Windows, Linux, Mac OS X, IOS, Android и Windows Phone.

private Capture \_capture = null; //берет картинку из потока вебки как изображение

public MainForm()

{

InitializeComponent();

CvInvoke.UseOpenCL = false;

radioButtonOrange.Checked = true;

try

{

\_capture = new Capture(0); // 0 - встроенная камера,

\_capture.ImageGrabbed += ProcessFrame; // 1 - подключенная,

} // -1 - любая доступная

catch (NullReferenceException e)

{

MessageBox.Show(e.Message);

}

}

**Разбиение изображения на потоки, фильтрация по цвету**

private Image<Gray, byte> RGBFilter(Image<Bgr, byte> input, //разбиваем на каналы и фильтруем по цвету

Gray Rmin, Gray Rmax,

Gray Gmin, Gray Gmax,

Gray Bmin, Gray Bmax)

{

Image<Gray, byte>[] channels = input.Split();

channels[0] = channels[0].InRange(Bmin, Bmax);

channels[1] = channels[1].InRange(Gmin, Gmax);

channels[2] = channels[2].InRange(Rmin, Rmax);

Image<Gray, byte> result = channels[0].And(channels[1]);

result = result.And(channels[2]);

return result;

}

**Нахождение объекта с помощью метода центра масс:**

private Image<Gray, byte> MassCenter(Image<Gray, byte> input) //ищем центр масс

{

int count = 0;

center\_x = 0;

center\_y = 0;

for (int i = 0; i < input.Height; i++) //пробегаемся по пикселям

{

for (int j = 0; j < input.Width; j++)

{

if (input.Data[i, j, 0] == 255) // суммируем координаты белых пикселей

{

center\_x += j;

center\_y += i;

count++; // и их количество

}

}

}

if (count > 600)

{

\_isDetected = true;

my\_timer.Reset(); //Перезапускаем таймер

center\_x = center\_x / count; //находим координаты центра масс

center\_y = center\_y / count;

labelCenter.Text = "X: " + center\_x.ToString() +

" Y: " + (center\_y).ToString() +

" count: " + count.ToString();

input = RectPaint(input, (int)center\_x, (int)center\_y, count); //рисуем прямоугольник

}

else

{

\_isDetected = false;

my\_timer.Start(); // запуск таймера

labelCenter.Text = " count: " + count.ToString();

}

return input;

}

**Метод минимальных квадратов:**

public class Line\_func

{

private Double a\_;

private Double b\_;

public Double coord\_a\_

{

get { return a\_; }

set { a\_ = value; }

}

public Double coord\_b\_

{

get { return b\_; }

set { b\_ = value; }

}

public Line\_func()

{

coord\_a\_ = 0;

coord\_b\_ = 0;

}

public Line\_func(List<Double> x\_a, List<Double> y\_a)

{

if (x\_a.Count != y\_a.Count)

throw new Exception(" Массивы должны быть одной длинны!");

else

{

Double x\_mid = 0;

Double y\_mid = 0;

Double xx\_mid = 0;

Double xy\_mid = 0;

Int64 x\_sum = 0;

Int64 y\_sum = 0;

Int64 xx\_sum = 0;

Int64 xy\_sum = 0;

Int32 N = x\_a.Count;

Double[] xx\_a = new Double[N];

Double[] xy\_a = new Double[N];

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

{

xx\_a[i] = Math.Pow(x\_a[i], 2);

xy\_a[i] = x\_a[i] \* y\_a[i];

x\_sum += (Int64)x\_a[i];

y\_sum += (Int64)y\_a[i];

xx\_sum += (Int64)xx\_a[i];

xy\_sum += (Int64)xy\_a[i];

}

x\_mid = (Double)x\_sum / N;

y\_mid = (Double)y\_sum / N;

xx\_mid = (Double)xx\_sum / N;

xy\_mid = (Double)xy\_sum / N;

coord\_a\_ = (xy\_mid - (y\_mid \* x\_mid)) / (xx\_mid - Math.Pow(x\_mid, 2));

coord\_b\_ = ((xx\_mid \* y\_mid) - (x\_mid \* xy\_mid)) / (xx\_mid - Math.Pow(x\_mid, 2));

}

}

public void Print()

{

Console.WriteLine(" {0:F} \* X + {1:F}", a\_, b\_);

}

}

**Организация медиа плеера:**

Для создания медиа плеера был выбран объект класса axWindowsMediaPlayer, реализующий стандартный проигрыватель и предоставляющий контроль к его функционалу.

**Связать методы анализа с жестами:**

public partial class FormChooseFile : Form

{

static public MediaPlayer mediaPlayer = new MediaPlayer();

static WMPLib.IWMPPlaylist playlist = mediaPlayer.WindowsMediaPlayer.playlistCollection.newPlaylist("myplaylist");

static WMPLib.IWMPMedia media;

static int numOfMusic = 0;

public FormChooseFile()

{

InitializeComponent();

mediaPlayer.Owner = this;

}

private void buttonBrowse\_Click(object sender, EventArgs e)

{

OpenFileDialog dialog = new OpenFileDialog();

dialog.InitialDirectory = Environment.SpecialFolder.MyMusic.ToString();

dialog.Filter = "MP3 files |\*.mp3| All files {\*.\*}|\*.\*";

dialog.FilterIndex = 1;

dialog.Multiselect = true;

if (dialog.ShowDialog() == DialogResult.OK)

{

if (dialog.FileNames != null)

{

foreach (String str in dialog.FileNames)

{

listBoxPlaylist.Items.Add(str);

}

foreach (String str in listBoxPlaylist.Items)

{

media = mediaPlayer.WindowsMediaPlayer.newMedia(str);

playlist.appendItem(media);

numOfMusic++;

}

mediaPlayer.WindowsMediaPlayer.currentPlaylist = playlist;

}

}

Thread mainThread = new Thread(() => mediaPlayer.Show());

mainThread.IsBackground = true;

mainThread.Start();

mediaPlayer.WindowsMediaPlayer.Ctlcontrols.play();

listBoxPlaylist.SelectedIndex = 0;

}

public void nextSong()

{

mediaPlayer.WindowsMediaPlayer.Ctlcontrols.next();

}

public void prevSong()

{

mediaPlayer.WindowsMediaPlayer.Ctlcontrols.previous();

}

public void playSong()

{

mediaPlayer.WindowsMediaPlayer.Ctlcontrols.play();

}

public void stopSong()

{

mediaPlayer.WindowsMediaPlayer.Ctlcontrols.stop();

}

public void voulumeUp()

{

int vol = mediaPlayer.WindowsMediaPlayer.settings.volume;

if (vol <= 80)

vol = mediaPlayer.WindowsMediaPlayer.settings.volume + 20;

else

vol = 100;

mediaPlayer.WindowsMediaPlayer.settings.volume = vol;

}

public void voulumeDown()

{

int vol = mediaPlayer.WindowsMediaPlayer.settings.volume;

if (vol > 20)

vol = mediaPlayer.WindowsMediaPlayer.settings.volume - 20;

else

vol = 0;

mediaPlayer.WindowsMediaPlayer.settings.volume = vol;

}

**Интерфейс:**

