

# Urządzenia peryferyjne - laboratorium

Karol Kulawiec 241281

Jakub Kalina 241346

13.11.2019

## 1 Wstęp

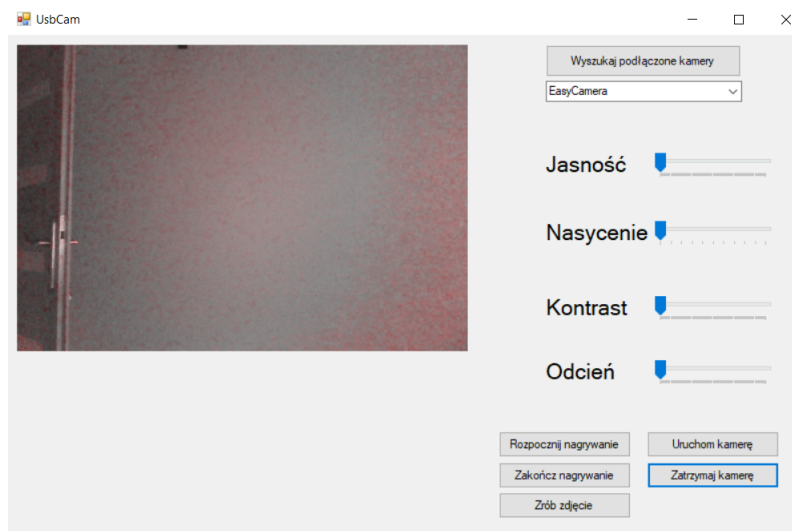
Celem wykonanego ćwiczenia było napisanie programu, za pomocą którego będzie można wyświetlić oraz przetworzyć obraz nagrywany kamerą USB.

Kamera wykorzystana na zajęciach do komunikacji z komputerem wykorzystywała port USB 2.0. Była to kamera cyfrowa, która może transmitować obrazy statyczne lub transmisja może odbywać się w sposób ciągły. Najczęściej obraz rzutowany na matrycę jest typu CCD (Charge coupled devices) lub CMOS (Complementary metal-oxide semiconductor). Obraz był nagrywany w rozdzielczości 640x480 pikseli w 30 klatkach na sekundę (Zmiana kamery na kamerkę zamontowaną w laptopie zmieniała rozdzielczość).

## 2 Opis aplikacji

Program został napisany w technologii *C#*, z wykorzystaniem ogólnodostępnej platformy AForge.NET, której głównym przeznaczeniem jest przetwarzanie obrazu, budowanie sieci neuronowych, pisanie algorytmów genetycznych, rozmytej logiki, algorytmów uczenia maszynowego oraz innych, podobnych. Dla naszych potrzeb, korzystaliśmy głównie z bibliotek AForge.Video oraz AForge.Imaging, służących do przetwarzania i filtrowania obrazów oraz wideo.

Aplikacja prezentuje się w następujący sposób:



Rysunek 1: Interface aplikacji

W lewym górnym rogu widoczny jest bieżący obraz nagrywany przez kamerę. Obraz pojawia się po wybraniu kamery oraz naciśnięciu przycisku "Uruchom kamerę". Po prawej stronie aplikacji, zaczynając od góry, kolejno znajdują się:

1. Przycisk do wyszukania kamer dostępnych na komputerze.
2. Rozwijana lista, w której znajdują się wszystkie znalezione kamery.
3. Cztery suwaki do ustawiania:

- jasności obrazu.
  - nasycenia obrazu.
  - kontrastu obrazu.
  - odcieni obrazu.
4. Pięć przycisków, których funkcje są następujące:
- Przycisk "Rozpocznij nagrywanie" w momencie naciśnięcia, sprawdza, czy aktualnie któraś z kamer rejestruje obraz, jeżeli tak, to wybieramy plik (w formacie mvi), do którego zapiszemy nagranie.
  - Przycisk "Zakończ nagrywanie" w momencie naciśnięcia, sprawdza, czy aktualnie trwa jakieś nagranie, jeżeli trwa, to zatrzymuje je i zapisuje plik, podany w wcześniejszym podpunkcie.
  - Przycisk "Zrób zdjęcie" zapisuje bieżącą klatkę do pliku.
  - Przycisk "Uruchom kamerę" powoduje, że obraz z wybranej kamery zaczyna być rejestrowany i wyświetlany w polu obok.
  - Przycisk "Zatrzymaj kamerę" powoduje zaprzestanie rejestrowania obrazu z wybranej kamery.

### 3 Opis ważniejszych funkcji

- Wyświetlenie przechwyconego obrazu:

```
private void DisplayCapturedPicture(object sender,
    NewFrameEventArgs eventArgs)
{
    Bitmap bitmap = new Bitmap(eventArgs.Frame, new
        Size(cameraPictureBox.Width, cameraPictureBox.Height));
    BrightnessCorrection brightnessCorrection = new
        BrightnessCorrection(Brightness);
    ContrastCorrection contrastCorrection = new
        ContrastCorrection(Contrast);
    SaturationCorrection saturationCorrection = new
        SaturationCorrection(Saturation * 0.1f);
    HueModifier hueModifier = new HueModifier(Hue);

    bitmap =
        brightnessCorrection.Apply((Bitmap)bitmap.Clone());
    bitmap = contrastCorrection.Apply((Bitmap)bitmap.Clone());
    bitmap =
        saturationCorrection.Apply((Bitmap)bitmap.Clone());
    bitmap = hueModifier.Apply((Bitmap)bitmap.Clone());

    if (connectedCameraIsRecording)
    {
        videoWriter.WriteVideoFrame(bitmap);
    }

    cameraPictureBox.Image = bitmap;
}
```

Na początku zapisujemy bitmapę z aktualnej ramki zdarzenia o rozmiarach okienka do wyświetlania obrazu. Następnie pobieramy jasność, nasycenie, kontrast i odcień obrazu, których wartość ustalana jest na podstawie położenia suwaków. Kolejnym krokiem jest zmiana tych parametrów w naszej bitmapie. Ostatnimi krokami jest sprawdzenie, czy nagrywamy obraz oraz podmienienie obrazu w okienku, na nowy. Jeżeli nagrywamy obraz, bitmapa zostaje zapisana.

- Rozpoczęcie nagrywania

```

private void startRecordingButton_Click(object sender,
EventArgs e)
{
    try
    {
        if (connectedCamera.IsRunning)
        {
            try
            {
                saveVideoFileDialog = new SaveFileDialog();
                saveVideoFileDialog.Filter = "Avi Files (*.avi)|*.avi";
                saveVideoFileDialog.Title = "Wskaz miejsce do zapisu";
                saveVideoFileDialog.ShowDialog();
                videoWriter = new VideoFileWriter();
                videoWriter.Open(saveVideoFileDialog.FileName,
                                cameraPictureBox.Image.Width,
                                cameraPictureBox.Image.Height, 30,
                                VideoCodec.MPEG4);
                connectedCameraIsRecording = true;
            }
            catch (Exception) { }
        }
    }
    catch (Exception)
    {
        MessageBox.Show("Należy najpierw uruchomic wybrana kamere");
    }
}

```

Po sprawdzeniu, czy aktualnie przetwarzamy jakiś obraz z kamery tworzymy plik o rozszerzeniu .avi, do którego będziemy zapisywać nasze nagranie. Następnie tworzymy obiekt zapisujący nagranie do pliku, w którym podajemy nazwę tego pliku, wymiary obrazu, liczbę klatek na sekundę oraz kodek - w naszym przypadku, jest to 30 klatek na sekundę oraz MPEG-4.

- Zrobienie zdjęcia

```

private void takePicture_Click(object sender, EventArgs e)
{
    Bitmap picture = new Bitmap(cameraPictureBox.Image);
    picture.Save("screen.bmp");
    picture.Dispose();
    picture = null;
}

```

Aby zrobić zdjęcie, pobieramy obraz z okienka i zapisujemy go jako bitmapę. następnie przy użyciu funkcji Save() zapisujemy go do pliku "screen.bmp" w folderze roboczym.

## 4 Wnioski

Platforma AForge posiada wiele ciekawych funkcjonalności, umożliwiających zarządzać obrazem wg własnych zapotrzebowań. A ponieważ jest to platforma z otwartym źródłem, ciągle jest ona rozwijana przez każdego programistę, który ma na to ochotę.

Podczas zajęć nie było większych problemów z wykorzystaniem owej platformy. Dzięki jednoczesnej możliwości modyfikacji obrazu i jego podglądu, mogliśmy się przekonać jak wpływają zmiany poszczególnych parametrów.

Link do repozytorium, w którym znajduje się program napisany na zajęciach: <https://github.com/JakubKalina/UP-Kamera-USB>.