Wojskowa Akademia Techniczna im. Jarosława Dąbrowskiego

Wydział Cybernetyki, kierunek informatyka - inżynieria systemów

Laboratorium z przedmiotu:
Grafika komputerowa

Sprawozdanie z ćwiczenia laboratoryjnego nr 1,2:

Przekształcanie obrazów rastrowych, poprawa jakości obrazu

Prowadzący:
dr inż. Marek Salamon

Wykonał: Radosław Relidzyński, **Grupa:** WCY20IY4S1, nr 17
Daty laboratoriów: 3.11.2022 r., 21.11.2022 r.
Termin oddania ćwiczenia: 6.12.2022 r.

Spis treści

A. Treść zadania – laboratorium 1

1. Używając narzędzia ToolBox / Przybornik zmodyfikować graficzne okno aplikacji wprowadzając:
    a. Modyfikacje opisów funkcji programu zgodnie z zadaniem indywidualnym
    b. Informację o wykonawcy (imię i nazwisko, grupa, data wykonania)
2. Przygotować nowy obraz (kolorowy) o parametrach: $K = L \neq 256$, plik BMP 24 bitowy. Wprowadzić nowy obraz do okna aplikacji.
3. Zmodyfikować algorytmy przekształceń obrazu zgodnie z zadaniem indywidualnym:
    a. Napisać algorytm sterujący generatorem adresu odczytu w celu uzyskania efektu przesuwania poziomego obrazu w kierunku prawej strony ekranu
    b. Napisać algorytm sterujący generatorem adresu odczytu w celu uzyskania efektu zasłaniania pionowego obrazu w kierunku górnej krawędzi ekranu.
    c. Napisać algorytm sterujący generatorem adresu odczytu w celu uzyskania efektu przewijania obrazu wzdłuż przekątnej ekranu w kierunku górnego prawego wierzchołka
4. Używając narzędzia ToolBox wprowadzić pasek ProgressBar i powiązać z kolejną klatką wykonywanego przekształcenia obrazu.

B. Sposób rozwiązania zadań – laboratorium 1

Zastosowane zdjęcie:

Rozwiązanie zadań:

Napisać algorytm sterujący generatorem adresu odczytu w celu uzyskania efektu przesuwania poziomego obrazu w kierunku prawej strony ekranu

Kod programu:

```
public void Efekt1()
{
    //efekt: przesuwanie poziome obrazu w kierunku prawej strony ekranu

    if (p >= K) p = 0;

    for (int j = 1; j <= L; j++)
    {
        for(int i = 1; i <= p; i++)
            ReadTlo(N);
```

```
            for (int i = 1; i <= K - p; i++)
                ReadPixel(i, j);
        }
    }
```

Działanie programu:

Funkcja na każdy krok odczytuje każdy wiersz od początku (nie do końca), ale wypisuje go dopiero od miejsca oddalonego o p od lewej krawędzi.

Napisać algorytm sterujący generatorem adresu odczytu w celu uzyskania efektu zasłaniania pionowego obrazu w kierunku górnej krawędzi ekranu.

Kod programu:

```
    public void Efekt2()
    {
        //efekt: zasłanianie pionowe obrazu w kierunku górnej krawędzi ekranu

        if (p >= L) p = 0;

        for (int j = 1; j <= L - p; j++)
        {
            for (int i = 1; i <= K; i++)
                ReadPixel(i, j);
        }

        for (int j = 1; j <= p; j++)
        {
            for (int i = 1; i <= K; i++)
                ReadTlo(N);
        }
    }
```

Działanie programu:

Funkcja na każdy krok odczytuje każdy wiersz od początku do końca, ale przestaje nie wczytuje w momencie, gdy wiersz należy do p kolejnych ostatnich.

Napisać algorytm sterujący generatorem adresu odczytu w celu uzyskania efektu przewijania obrazu wzdłuż przekątnej ekranu w kierunku górnego prawego wierzchołka

Kod programu:

```
    public void Efekt3()
    {
        //efekt: przewijanie obrazu wzdłuż przekątnej ekranu w kierunku górnego
prawego wierzchołka

        if (p >= L) p = 0;

        for (int j = 1 + p; j <= L; j++)
        {
            for (int i = K - p; i < K; i++)
                ReadTlo(N);
            for (int i = 1; i <= K - p; i++)
                ReadPixel(i, j);
        }
```

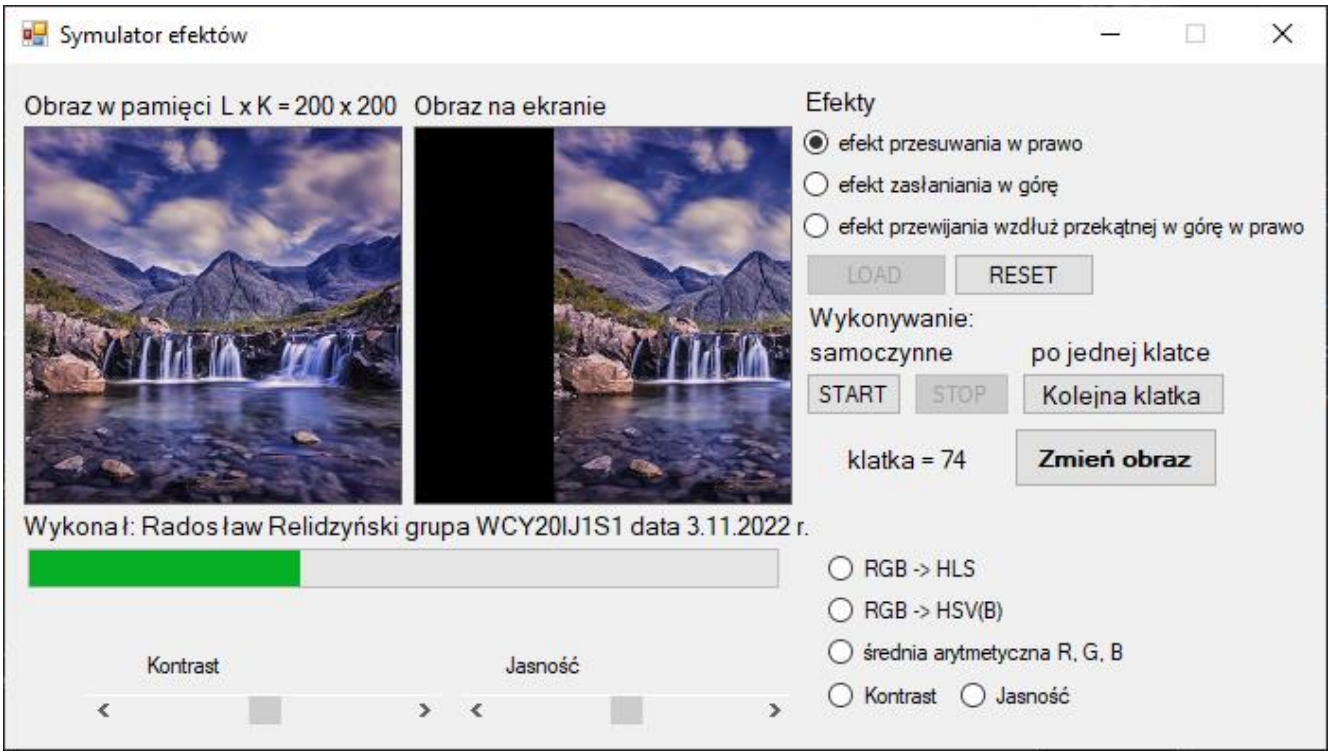
```
            for (int j = 1; j <= p; j++)
            {
                for (int i = K - p; i < K; i++)
                    ReadPixel(i, j);
                for (int i = 1; i <= K - p; i++)
                    ReadTlo(N);
            }

        }
```

Działanie programu:

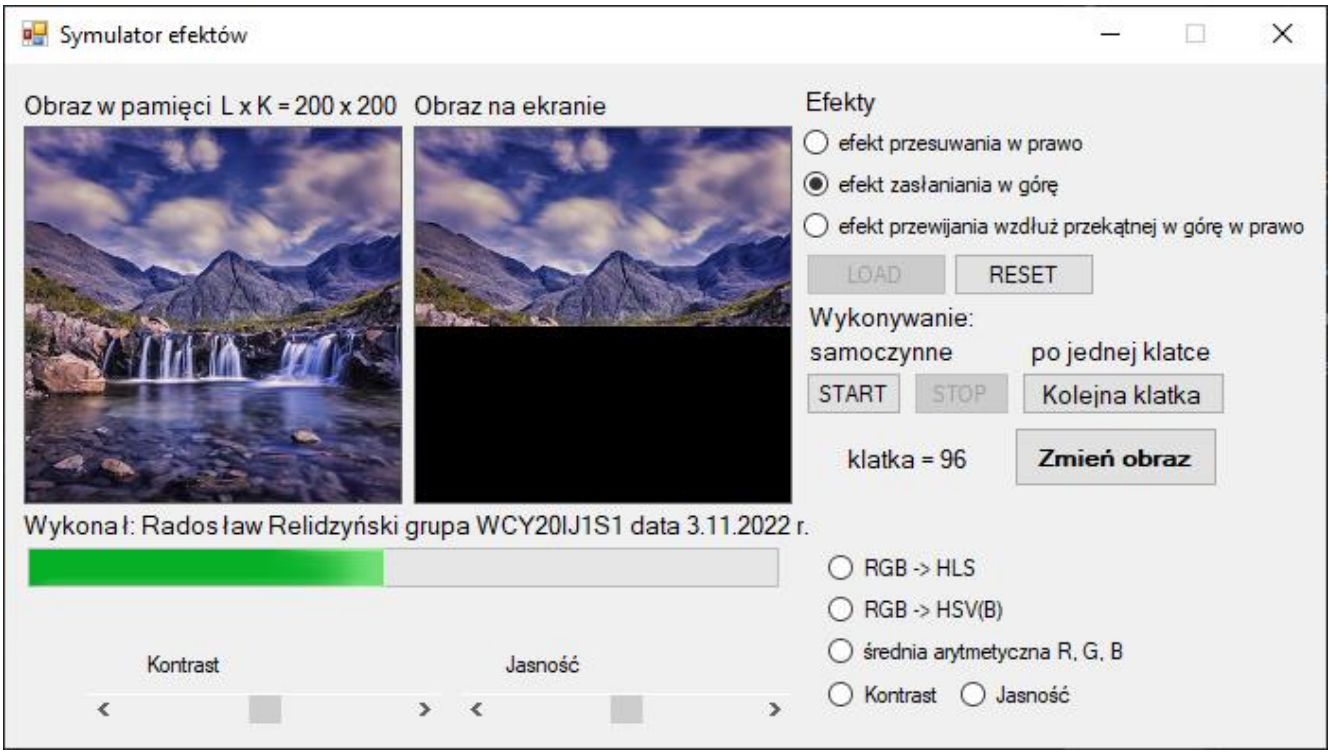
Funkcja pracuje na 4 segmentach programu. Od lewej od góry: tło, obraz (od lewego dolnego rogu), obraz (od prawego górnego rogu), tło. W danym momencie określa co potrzebuje (tło czy piksel). Zmienna p określa granicę w pionie/poziomie między tymi segmentami.

C. Prezentacja działania programu – laboratorium 1

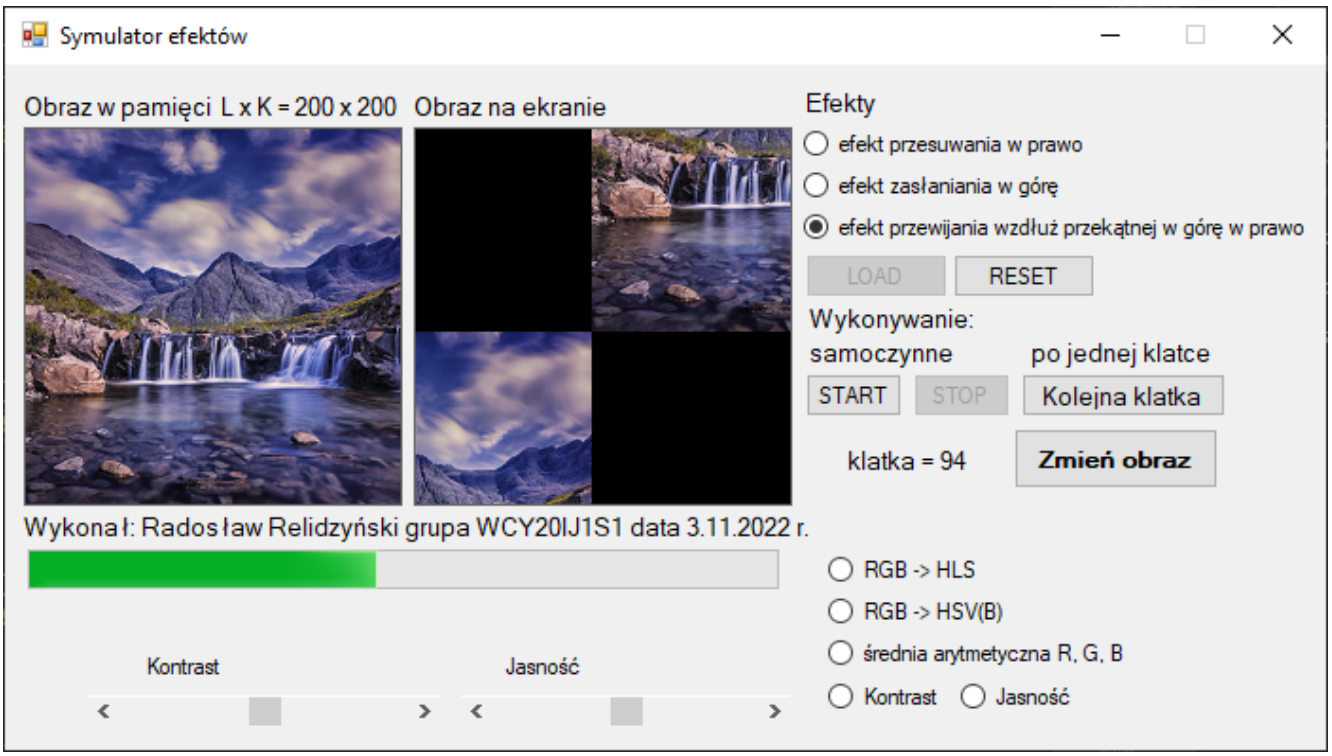
Zadanie 1.



Zadanie 2.



Zadanie 3.



D. Treść zadania – laboratorium 2

1. Zamienić obraz kolorowy (RGB) o 24-bitowej strukturze piksela z ćwiczenia 1 na obraz monochromatyczny reprezentowany przez skalę odcieni szarości (poziomy jasności) wykorzystując:
    a. Równanie przejścia z modelu RGB na HLS,
    b. Równanie przejścia z modelu RGB na HSV(B)

c. Średnią arytmetyczną składowych R, G, B

Wyznaczyć jasność (J) i kontrast (K) uzyskanych obrazów.

2. Wykorzystując liniową korektę tonalną napisać program umożliwiający zmianę jasności w pełnym zakresie:
   a. Obrazu kolorowego RGB (z ćwicz 1.)
   b. Wyznaczyć jasność (J) i kontrast (K) uzyskiwanych wyników

   Do zmiany jasności / kontrastu wykorzystać suwaki ScrollBar (H/V).

E. Sposób rozwiązania zadań – laboratorium 2

Funkcja hls:

```
        public void Zmien_obraz_hls()
         {
             System.Drawing.Color pixel;
             double sum = 0;
             double brightness = 0;
             double contrast = 0;

             for (int j = 1; j <= L; j++)
                 for (int i = 1; i <= K; i++)
                 {
                     pixel = m_obraz_w_pamieci.GetPixel(i - 1, j - 1);

                     //--------------------------------------//
                     //miejsce na kod dokonujący zmianę obrazu
                     int l_hls = (System.Math.Max(System.Math.Max(pixel.R, pixel.G),
pixel.B) + System.Math.Min(System.Math.Min(pixel.R, pixel.G), pixel.B)) / 2;
                     pixel = System.Drawing.Color.FromArgb(l_hls, l_hls, l_hls);
                     sum += l_hls;
                     //--------------------------------------//
                     m_ekran.SetPixel(i - 1, j - 1, pixel);
                 }
             brightness = sum / (L * K);
             sum = 0;

             for (int j = 1; j <= L; j++)
                 for (int i = 1; i <= K; i++)
                 {
                     pixel = m_obraz_w_pamieci.GetPixel(i - 1, j - 1);

                     //--------------------------------------//
                     //miejsce na kod dokonujący zmianę obrazu
                     int l_hls = (System.Math.Max(System.Math.Max(pixel.R, pixel.G),
pixel.B) + System.Math.Min(System.Math.Min(pixel.R, pixel.G), pixel.B)) / 2;
                     double x = l_hls - brightness;
                     sum += x * x;
                     //--------------------------------------//
                 }
             contrast = System.Math.Sqrt(sum / (L * K));
             label3.Text = "jasnosc = " + System.Math.Round(((double)brightness / 255) *
100, 0) + "%";
             label4.Text = "kontrast = " + System.Math.Round(((double)contrast / 127.5) *
100, 0) + "%";
             SetBitMap(ref m_ekran);
         }
```

Zastosowany wzór:

L = (max(R,G,B) + min(R,G,B))/2

```
        public void Zmien_obraz_hsv()
        {
            System.Drawing.Color pixel;
            double sum = 0;
            double brightness = 0;
            double contrast = 0;

            for (int j = 1; j <= L; j++)
                for (int i = 1; i <= K; i++)
                {
                    pixel = m_obraz_w_pamieci.GetPixel(i - 1, j - 1);

                    //--------------------------------------//
                    //miejsce na kod dokonujący zmianę obrazu
                    int l_hsv = System.Math.Max(System.Math.Max(pixel.R, pixel.G),
pixel.B);

                    //pixel = System.Drawing.Color.FromArgb(pixel.R, pixel.G, pixel.B);
                    pixel = System.Drawing.Color.FromArgb(l_hsv, l_hsv, l_hsv);
                    sum += l_hsv;

                    //--------------------------------------//

                    m_ekran.SetPixel(i - 1, j - 1, pixel);
                }
            brightness = sum / (L * K);
            sum = 0;

            for (int j = 1; j <= L; j++)
                for (int i = 1; i <= K; i++)
                {
                    pixel = m_obraz_w_pamieci.GetPixel(i - 1, j - 1);

                    //--------------------------------------//
                    //miejsce na kod dokonujący zmianę obrazu
                    int l_hsv = System.Math.Max(System.Math.Max(pixel.R, pixel.G),
pixel.B);
                    double x = l_hsv - brightness;
                    sum += x * x;

                }
            contrast = System.Math.Sqrt(sum / (L * K));
            label3.Text = "jasnosc = " + System.Math.Round(((double)brightness / 255) *
100, 0) + "%";
            label4.Text = "kontrast = " + System.Math.Round(((double)contrast / 127.5) *
100, 0) + "%";

            SetBitMap(ref m_ekran);
        }
```

Zastosowany wzór:
L = max(R,G,B)

```
        public void Zmien_obraz_avg()
        {
            System.Drawing.Color pixel;
```

```
            double sum = 0;
            double brightness = 0;
            double contrast = 0;

            for (int j = 1; j <= L; j++)
                for (int i = 1; i <= K; i++)
                {
                    pixel = m_obraz_w_pamieci.GetPixel(i - 1, j - 1);

                    //--------------------------------------//
                    //miejsce na kod dokonujący zmianę obrazu
                    int avg = (pixel.R + pixel.G + pixel.B) / 3;

                    pixel = System.Drawing.Color.FromArgb(avg, avg, avg);
                    sum += avg;

                    //--------------------------------------//

                    m_ekran.SetPixel(i - 1, j - 1, pixel);
                }
            brightness = sum / (L * K);
            sum = 0;

            for (int j = 1; j <= L; j++)
                for (int i = 1; i <= K; i++)
                {
                    pixel = m_obraz_w_pamieci.GetPixel(i - 1, j - 1);

                    //--------------------------------------//
                    //miejsce na kod dokonujący zmianę obrazu
                    double avg = (double)(pixel.R + pixel.G + pixel.B) / 3;
                    double x = avg - brightness;
                    //pixel = System.Drawing.Color.FromArgb(avg, avg, avg);
                    sum += x * x;

                }
            contrast = System.Math.Sqrt(sum / (L * K));
            label3.Text = "jasnosc = " + System.Math.Round(((double)brightness / 255) *
100, 0) + "%";
            label4.Text = "kontrast = " + System.Math.Round(((double)contrast / 127.5) *
100, 0) + "%";

            SetBitMap(ref m_ekran);
        }
```

Zastosowany wzór:
L = (R+G+B)/3

```
        public void Zmien_obraz_jasnosc()
        {
            System.Drawing.Color pixel;
            double sum = 0;
            double brightness = 0;
            double contrast = 0;
            double x = hScrollBar2.Value;


            for (int j = 1; j <= L; j++)
                for (int i = 1; i <= K; i++)
                {
                    pixel = m_obraz_w_pamieci.GetPixel(i - 1, j - 1);
                    int r = (int)System.Math.Max(System.Math.Min(x + pixel.R, 255), 0);
```

```
                    int g = (int)System.Math.Max(System.Math.Min(x + pixel.G, 255), 0);
                    int b = (int)System.Math.Max(System.Math.Min(x + pixel.B, 255), 0);
                    double avg = (double)(r + g + b) / 3;
                    pixel = System.Drawing.Color.FromArgb(r, g, b);
                    sum += avg;

                    //------------------------------------//

                    m_ekran.SetPixel(i - 1, j - 1, pixel);
                }
            brightness = sum / (L * K);
            if (brightness == 255)
                contrast = 0;
            else
                contrast = System.Math.Sqrt(sum / (L * K));

            label3.Text = "jasnosc = " + System.Math.Round(((double)brightness / 255) *
100, 0) + "%";
            label4.Text = "kontrast = " + System.Math.Round(((double)contrast / 127.5) *
100, 0) + "%";


            SetBitMap(ref m_ekran);
        }
```

Zastosowany wzór:

$$J = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} f(i,j)$$

```
        public void Zmien_obraz_kontrast()
        {
            System.Drawing.Color pixel;
            double suma = 0;
            double jasnosc = 0;
            double kontrast = 0;
            int setVal = hScrollBar1.Value;


            for (int j = 1; j <= L; j++)
                for (int i = 1; i <= K; i++)
                {
                    pixel = m_obraz_w_pamieci.GetPixel(i - 1, j - 1);

                    double a;
                    if (setVal <= 0) a = 1.0 + (setVal / 256.0);
                    else a = 256.0 / System.Math.Pow(2, System.Math.Log(257 - setVal,
2));

                    int R = (int)System.Math.Min(System.Math.Max(a * (pixel.R - 127.5) +
127.5, 0), 255);
                    int G = (int)System.Math.Min(System.Math.Max(a * (pixel.G - 127.5) +
127.5, 0), 255);
                    int B = (int)System.Math.Min(System.Math.Max(a * (pixel.B - 127.5) +
127.5, 0), 255);
                    suma += (R + G + B) / 3;
                    pixel = System.Drawing.Color.FromArgb(R, G, B);

                    m_ekran.SetPixel(i - 1, j - 1, pixel);
                }
            jasnosc = suma / (L * K);
```

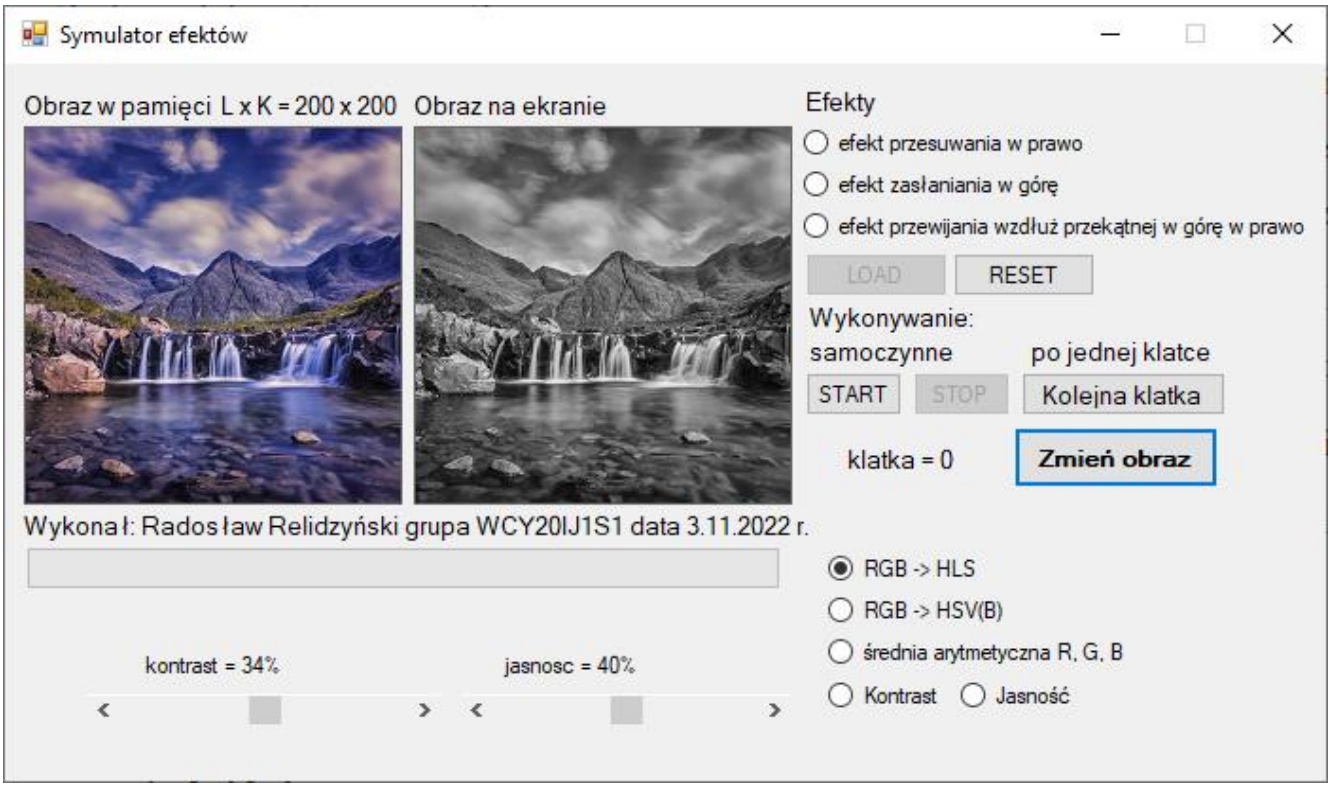
```
            suma = 0;
            for (int j = 1; j <= L; j++)
                for (int i = 1; i <= K; i++)
                {
                    pixel = m_obraz_w_pamieci.GetPixel(i - 1, j - 1);

                    double a;
                    if (setVal <= 0) a = 1.0 + (setVal / 256.0);
                    else a = 256.0 / System.Math.Pow(2, System.Math.Log(257 - setVal,
2));

                    int R = (int)System.Math.Min(System.Math.Max(a * (pixel.R - 127.5) +
127.5, 0), 255);
                    int G = (int)System.Math.Min(System.Math.Max(a * (pixel.G - 127.5) +
127.5, 0), 255);
                    int B = (int)System.Math.Min(System.Math.Max(a * (pixel.B - 127.5) +
127.5, 0), 255);
                    double r = (R + G + B) / 3 - jasnosc;
                    suma += r * r;
                }

            kontrast = System.Math.Sqrt(suma / (L * K));

            label3.Text = "Jasność = " + System.Math.Round(jasnosc / 255 * 100, 0) + "%";
            label4.Text = "Kontrast = " + System.Math.Round(kontrast / 127.5 * 100, 0) +
"%";

            SetBitMap(ref m_ekran);
        }
```

Zastosowany wzór:

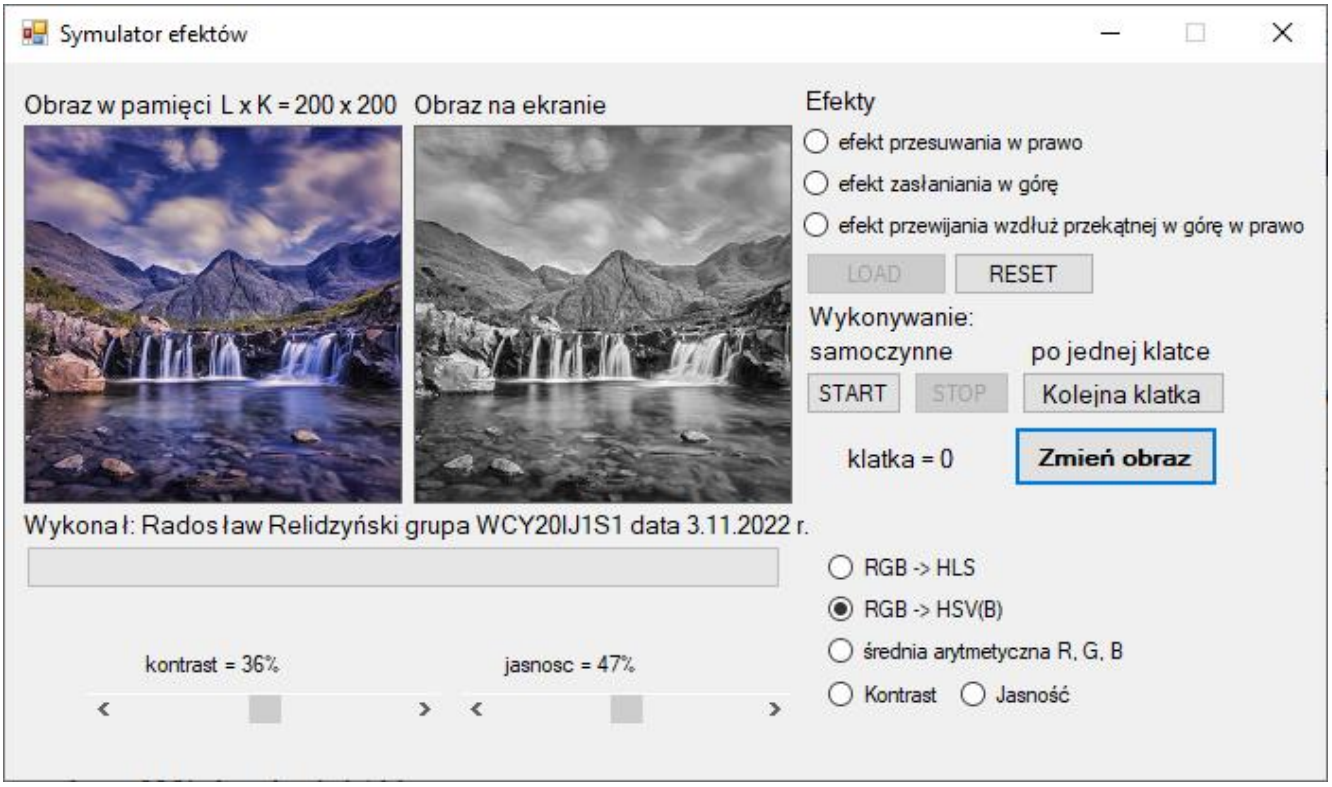
$$C = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[f(i,j) - J]^2}$$

F. Prezentacja działania programu – laboratorium 2

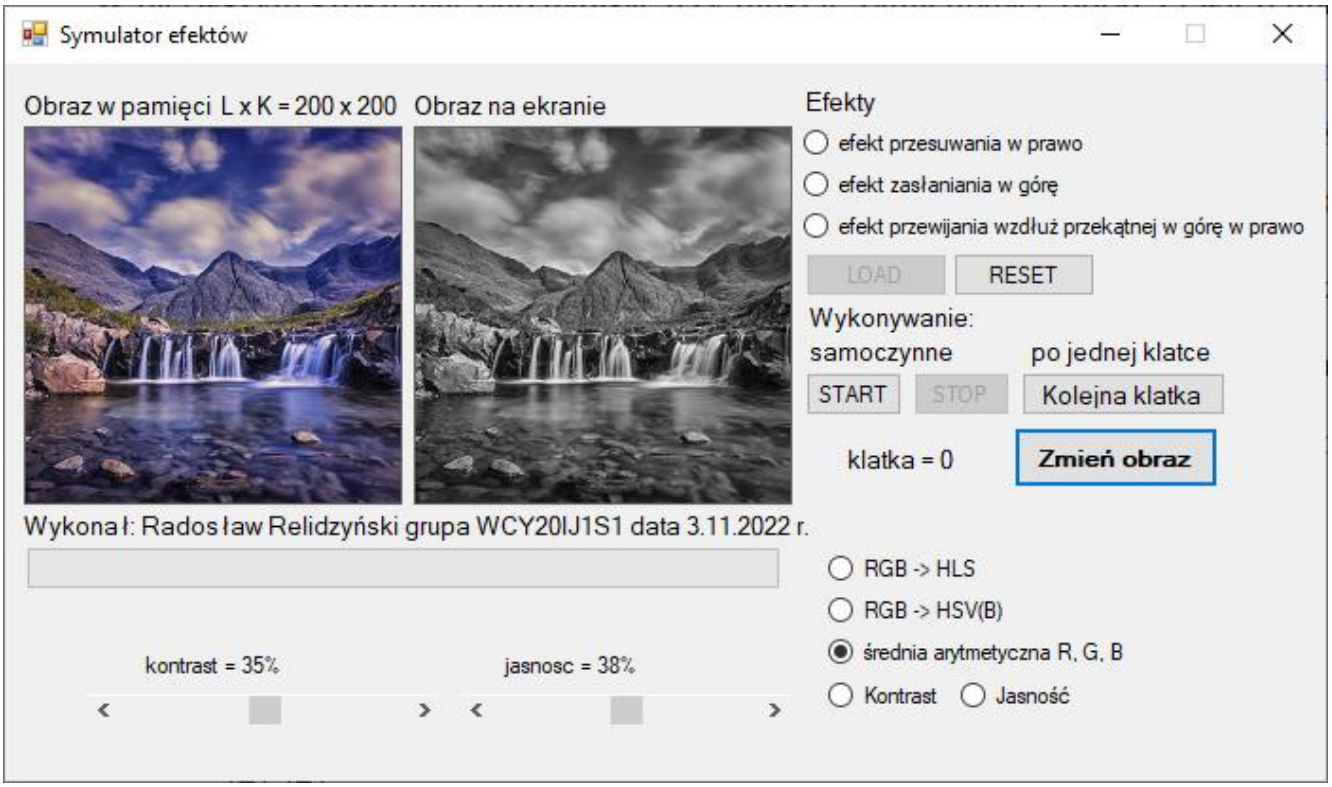
RGB -> HLS



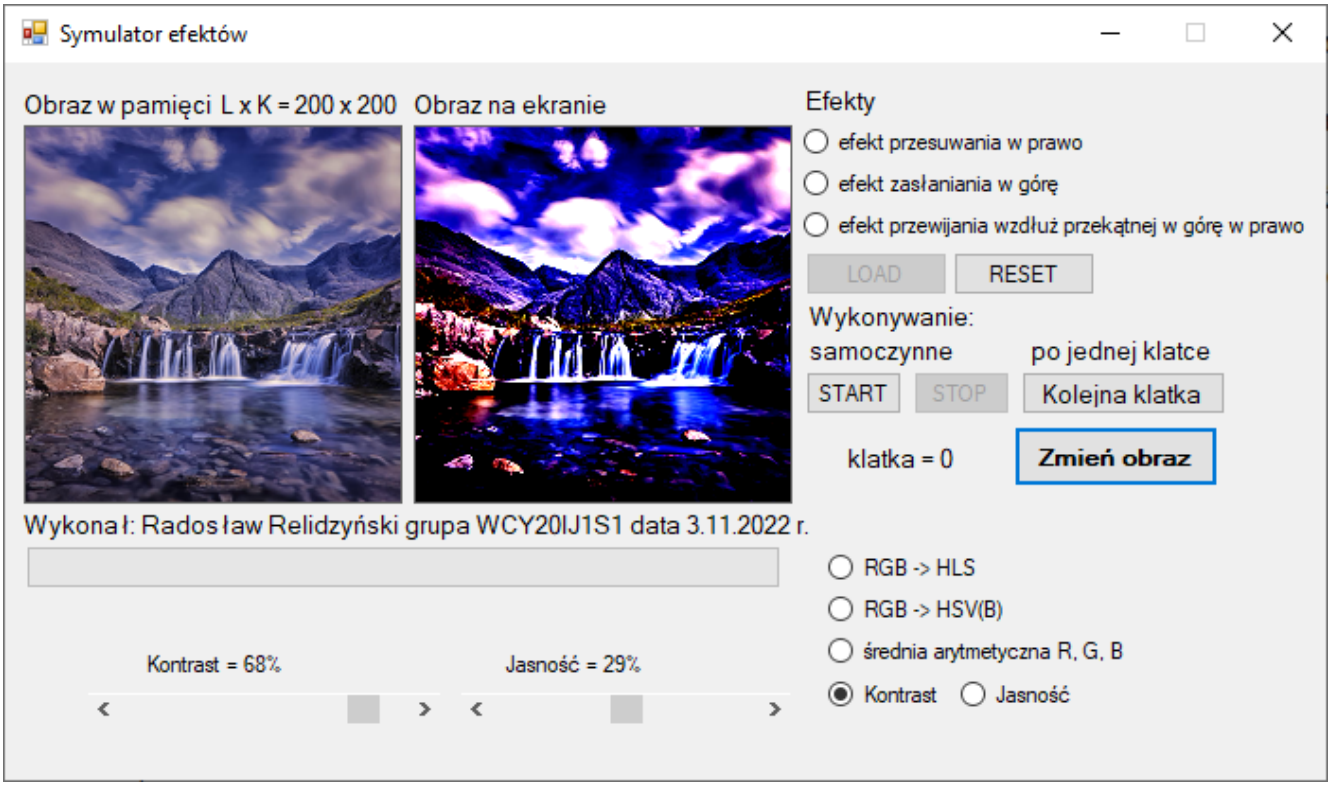
RGB -> HSV(B)



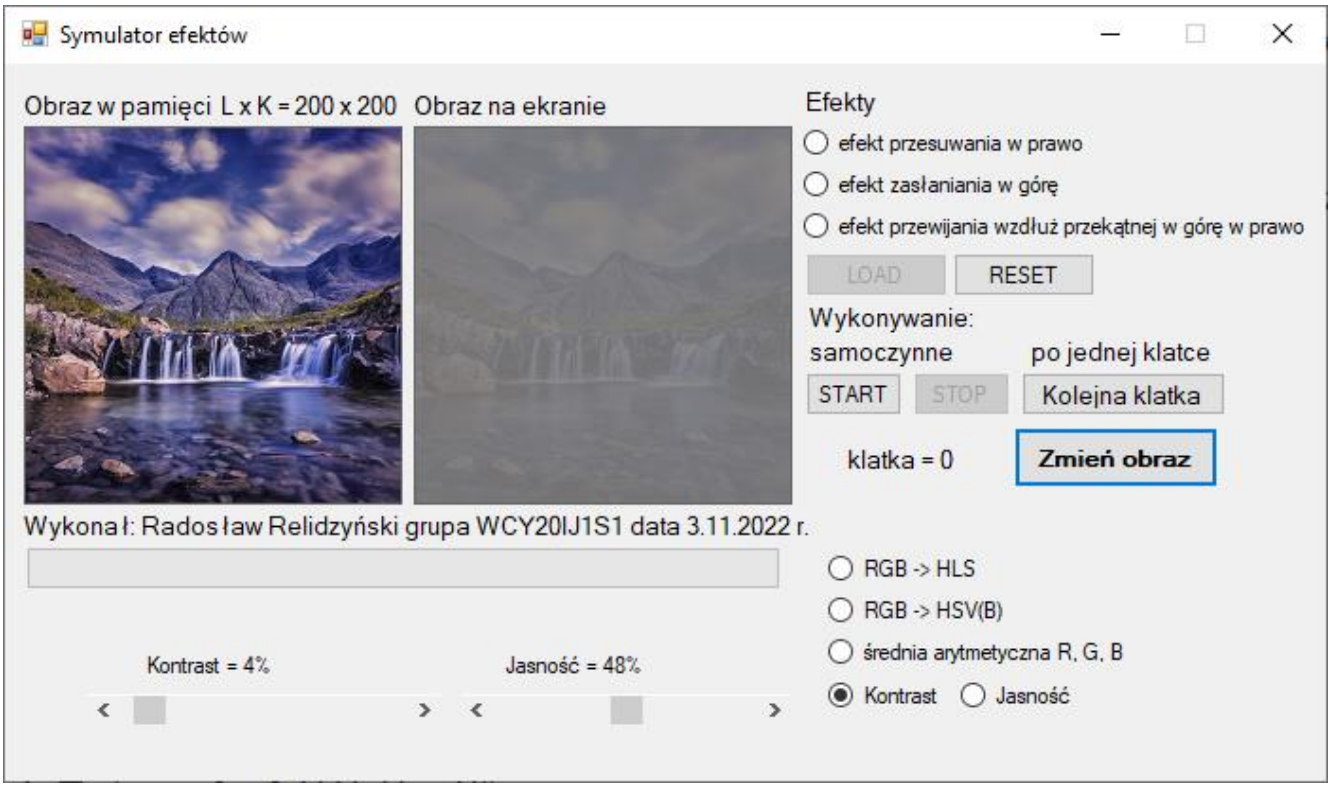
RGB -> średnia arytmetyczna R, G, B



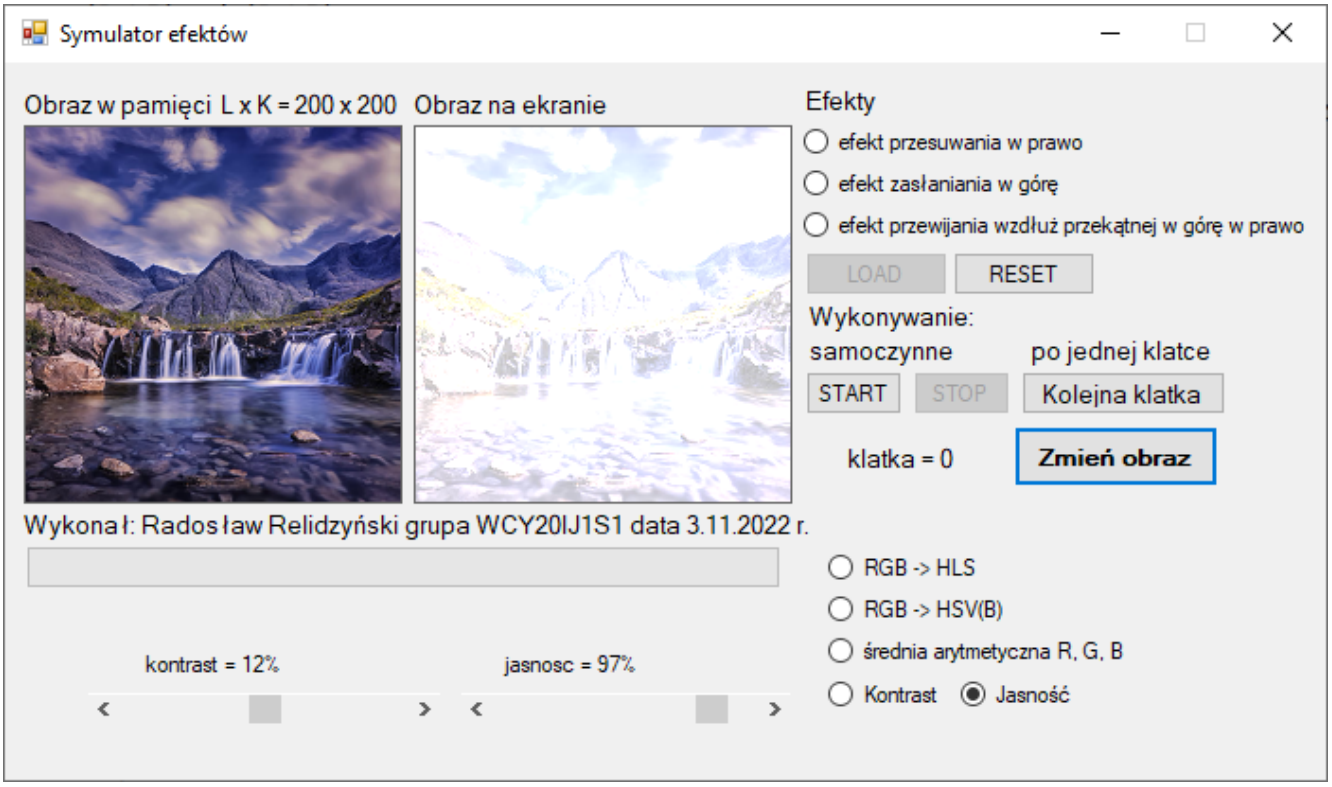
Zmiana kontrastu dużo



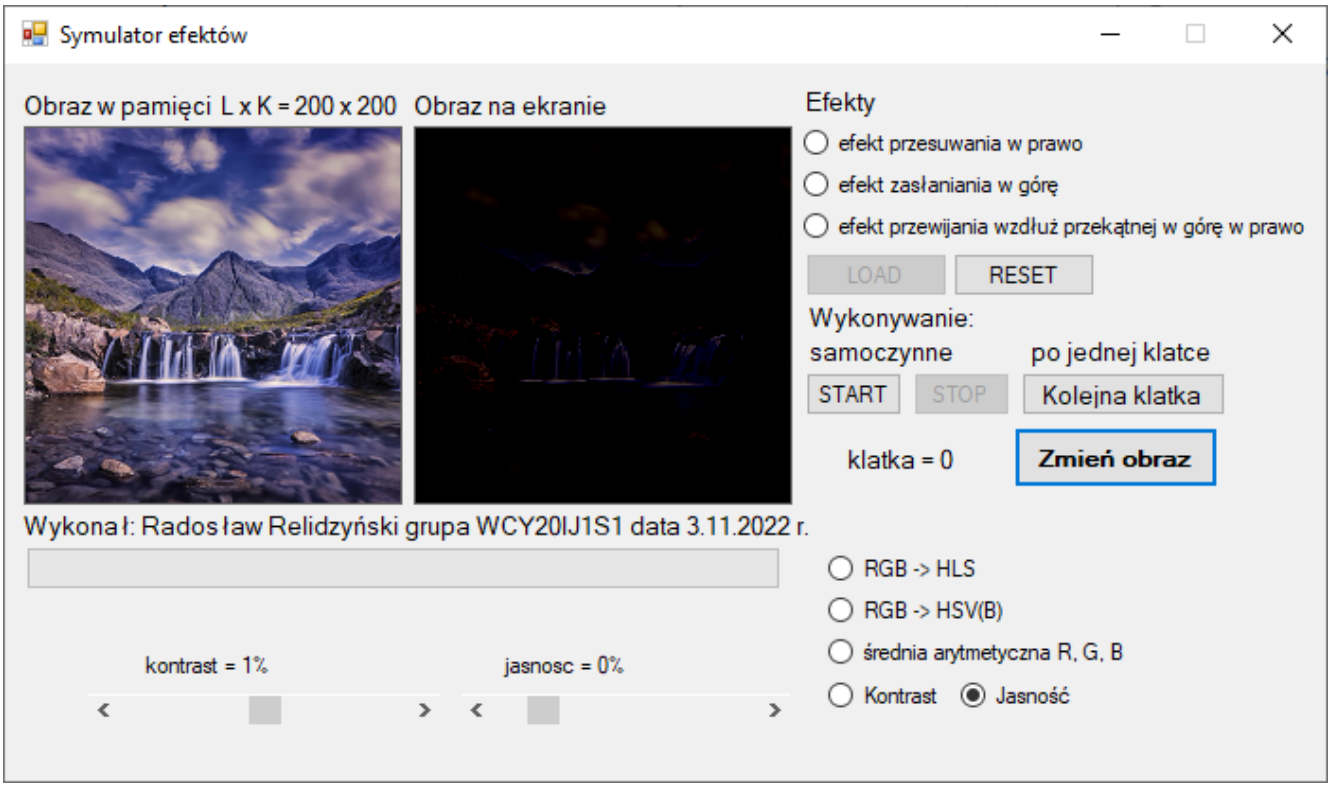
Zmiana kontrastu mało



Zmiana jasności dużo



Zmiana jasności mało



G. Wnioski

Laboratorium 1.

Udało się zrealizować zadanie, osiągnięty efekt jest zgodny z oczekiwanym.

To ćwiczenie pokazało, że tworzenie efektów przejścia, zejścia i tym podobnych odbywa się na manipulacji położeniem konkretnych pikseli. W zależności od oczekiwanego efektu określa się, które pola mają być puste, a które uzupełnione konkretnym pikselem bazowego obrazu. Wczytywanie kolejnej klatki obrazu odbywa się poprzez wczytywanie kolejno każdego piksela wiersz po wierszu od góry do dołu.

Laboratorium 2.

Udało się zrealizować zadanie, osiągnięty efekt jest zgodny z oczekiwanym.

Modele HLS, HSV(B), średnia arytmetyczna:

Choć każdy model konwertuje obraz na monochromatyczny, to każdy z nich stosuje inny wzór, co powoduje, że rezultaty dla każdego modelu różnią się. Największą jasność, a za razem kontrast osiągnął model HSV(B), a najmniejszy kontrast HLS, a najmniejszą jasność średnia arytmetyczna.

Ustawianie janości/kontrastu:

Zmiana jasności odbywa się przez zwiększanie/zmniejszanie składowych o wartość wszystkich składowych RGB. Zmiana kontrastu odbywa się przez zmianę wartości współczynnika a. Obydwa parametry oblicza się w oparciu o tablicę LUT(i) oraz o wartość i_{max} oznaczającą maksymalną dopuszczalną wartość składowej piksela obrazu.