Kontrast Obrazu

Wojciech Waleszczyk

Przebieg prezentacji

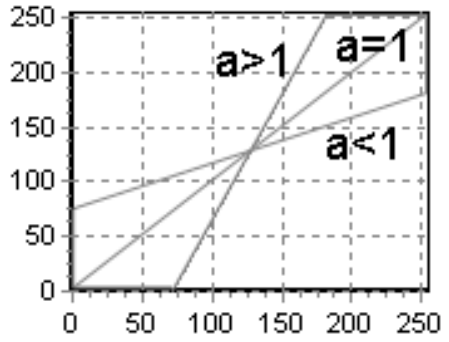
- Założenia projektowe
- Opis algorytmu
- Funkcjonalność plików DLL
- Opis działania DLL w assemblerze i c++
- Przykładowe działanie
- Wnioski

Założenia projektowe

- Program będzie porównywał zmianę kontrastu obrazu w języku wysokiego poziomu(C++) i niskiego(Assembler)
- Program będzie posiadał GUI wykonane w C#
- Program będzie wykonywany na wątkach(1-64)
- 64B architektura programu

Algorytm

$$LUT(i) = \begin{cases} 0 & \text{jeżeli } a(i - \frac{i_{max}}{2}) + \frac{i_{max}}{2} < 0 \\ a(i - \frac{i_{max}}{2}) + \frac{i_{max}}{2} & \text{jeżeli } 0 \le a(i - \frac{i_{max}}{2}) + \frac{i_{max}}{2} \le i_{max} \\ i_{max} & \text{jeżeli } a(i - \frac{i_{max}}{2}) + \frac{i_{max}}{2} > i_{max} \end{cases}$$

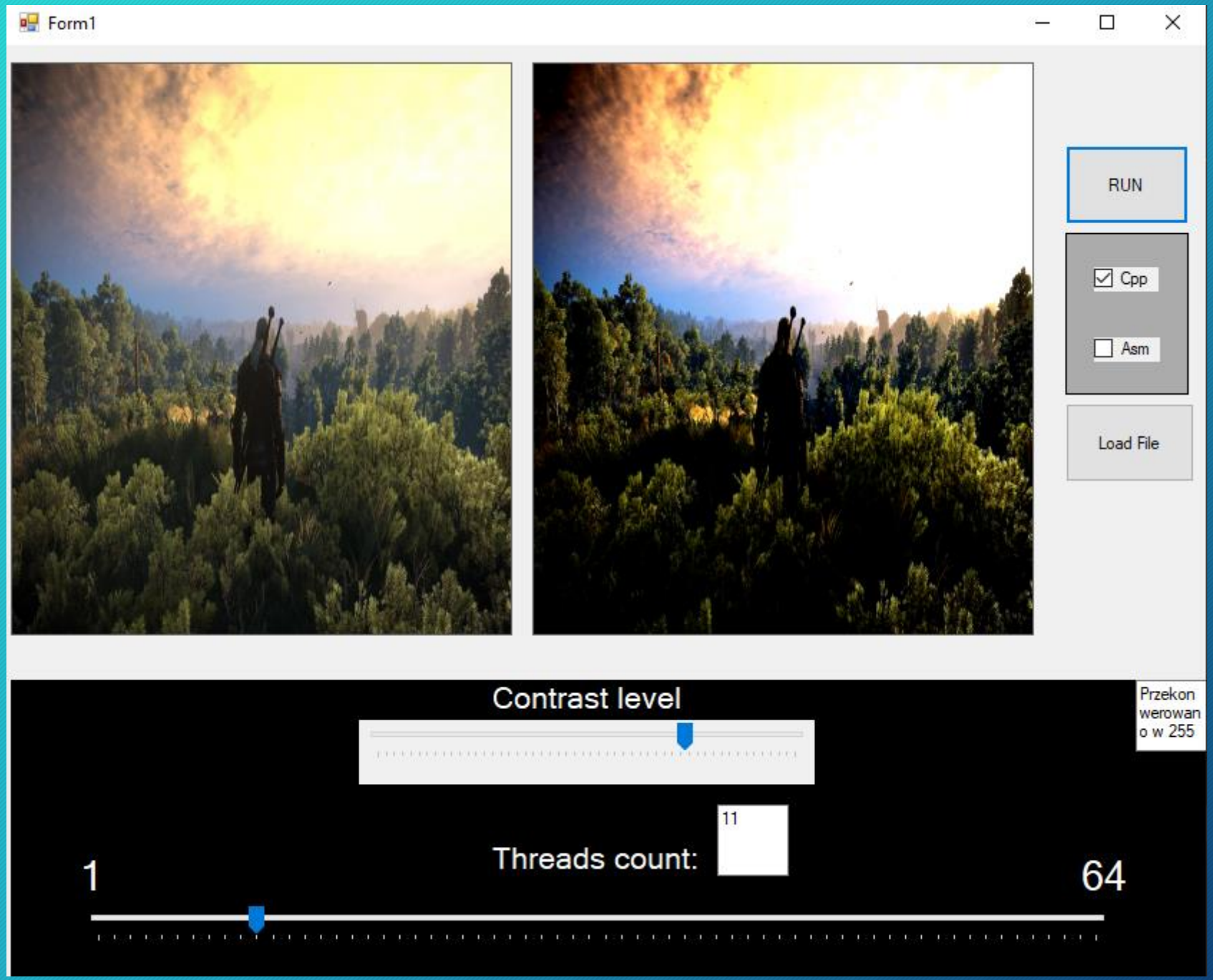


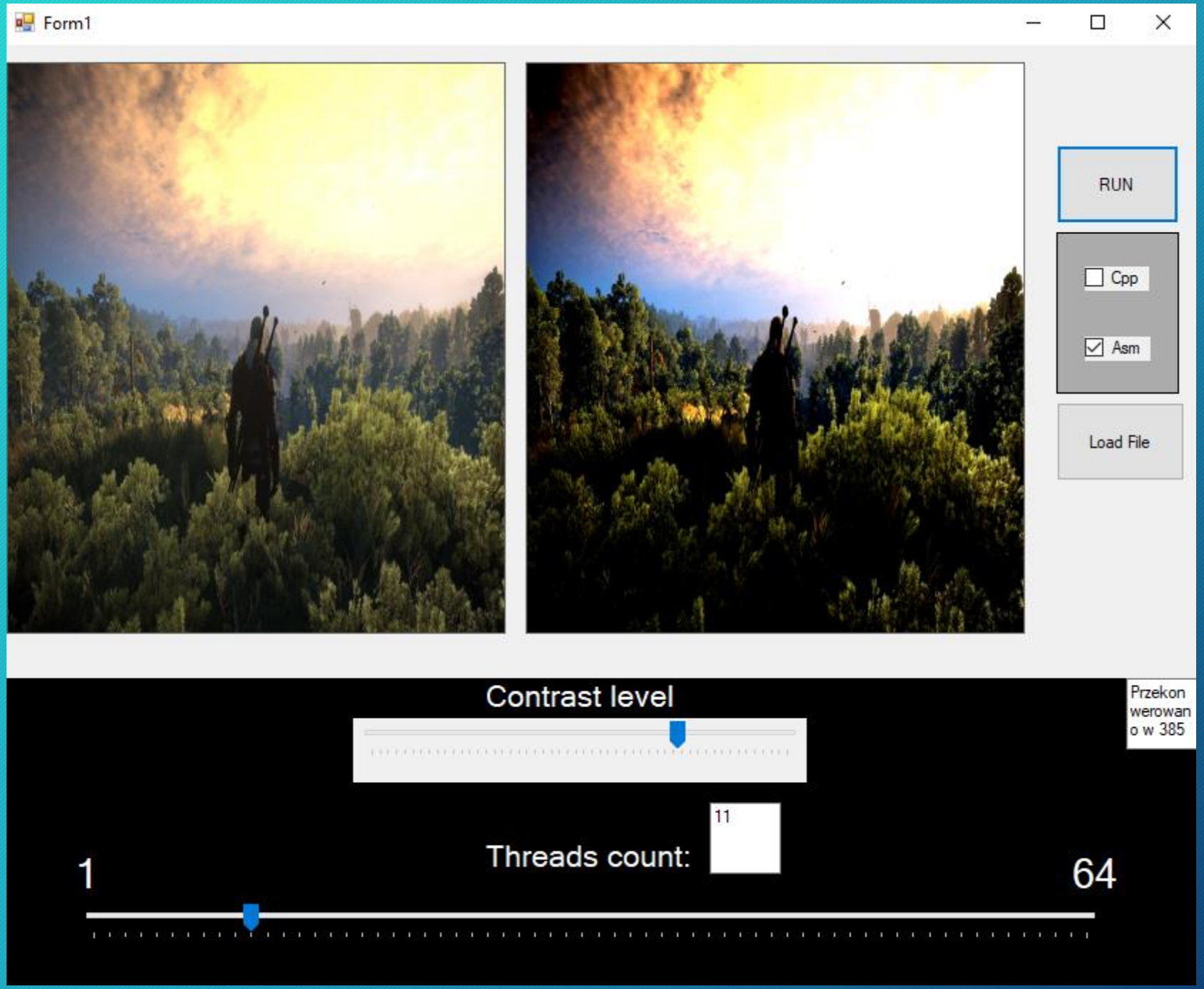
Pliki DLL

- W plikach DLL zawarłem funkcjonalność przypisania tablicy LUT od wartości pikseli obrazu do tablicy pikseli obrazu. W tablicy LUT dla każdego piksela przypisana jest odpowiednia jego wartość ze zmienionym kontrastem.

```
operateOnPixelsAsm PROC

    movdqu xmm1, [rbx + 0*SIZEOF BYTE]      ; load vector 1

    movdqu [rcx],xmm1                       ; load absolute value to result array

    ret
operateOnPixelsAsm ENDP
```

```
void __declspec(dllexport)  BitmapConvert(unsigned char* pixels, unsigned char* LUT)
{
    __m128i tab;
    tab = _mm_loadu_si128((__m128i*)LUT);
    _mm_storeu_si128((__m128i*)pixels, tab);
}
```


Form1
RUN
Cpp
Asm
Load File
Contrast level
Przekon
werowan
o w 255
11
Threads count:
1
64


Form1
RUN
Cpp
Asm
Load File
Contrast level
Przekon
werowan
o w 385
11
Threads count:
1
64

Wnioski

- Przy pracy nad tym projektem udoskonaliłem swoje umiejętności w programowaniu w języku niskiego poziomu jakim jest assembler jak również nauczyłem się pracy z C#
- Poznałem również czasową różnice między tym samym programem w c++ oraz assemblerze.