

Zmiana jasności obrazu

Projekt JA

Założenia

- Możliwość rozjaśniania obrazu
- Możliwość ściemniania obrazu
- Wykorzystanie instrukcji wektorowych
- Wykorzystanie wielowątkowości

Fragment kodu realizowany z wykorzystaniem bibliotek

- Podział pikseli na trzy tablice
 - R
 - G
 - B
- Rozjaśnianie – zwiększanie wartości o stałą
 - Zakres -255 do 255
- Dzielenie każdej z tablic na 16 bajtowe paczki i wysyłanie każdej z nich z osobna do funkcji .dll

Procedura cpp

```
void LightenC(unsigned char* tab, unsigned char val)
{
    __m128i* l = (__m128i*)tab;           // załadowanie tablicy 16 pikseli do rejestru xmm l
    __m128i* r = &_mm_set1_epi8(val);     // wypełnienie rejestru xmm r 16 wartościami val(współczynnik zmiany jasności)
    _mm_store_si128(l, _mm_adds_epu8(*l, *r)); // zsumowanie dwóch rejestrów xmm (r+l), a następnie zapisanie ich w xmm l
    return;
}

void DimC(unsigned char* tab, unsigned char val)
{
    __m128i* l = (__m128i*)tab;           // załadowanie tablicy 16 pikseli do rejestru xmm l
    __m128i* r = &_mm_set1_epi8(val);     // wypełnienie rejestru xmm r 16 wartościami val(współczynnik zmiany jasności)
    _mm_store_si128(l, _mm_subs_epu8(*l, *r)); // odejmowanie dwóch rejestrów xmm (l-r), a następnie zapisanie ich w xmm l
    return;
}
```

Procedura asm

```
1  .code
2  LightenASM PROC
3      MOVDQU XMM0, [RCX]          ; załadowanie 16 pikseli do rejestru XMM0
4      MOVQ XMM2, RDX              ; załadowanie wartości zmiany jasności do rejestru xmm 2
5      MOV EAX, 0                  ; załadowanie 0 do eax (licznik petli)
6      L1:                        ; petla zapelniajaca kazdy bajt xmm1 wartoscia zmiany jasnosci
7          PADDB XMM1, XMM2        ; dodanie xmm2 na ostatnie 8 bitow xmm1
8          PSLLDQ XMM1, 1          ; przesuniecie rejestru xmm1 bitowe o 1 bajt w lewo
9          ADD EAX, 1              ; inkrementacja licznika petli
10         CMP EAX, 16             ; sprawdzanie warunku koncowego petli (16 iteracji - rejestr xmm1 - 128 bitow = 16 bajtow)
11         JL L1                  ; powrot do poczatk petli
12         PADDB XMM1, XMM2        ; ostatnie dodanie xmm2 do xmm1
13         PADDUSB XMM0, XMM1      ; zsumowanie rejestru zawierajacego piksele z rejestrem zawierajacym wartosci zmiany jasnosci
14         MOVDQU [RCX], XMM0     ; załadowanie nowych wartości pikseli z powrotem do tablicy
15         RET
16 LightenASM ENDP

17
18 DimASM PROC
19     MOVDQU XMM0, [RCX]          ; załadowanie 16 pikseli do rejestru XMM0
20     MOVQ XMM2, RDX              ; załadowanie wartości zmiany jasności do rejestru xmm 2
21     MOV EAX, 0                  ; załadowanie 0 do eax (licznik petli)
22     L1:                        ; petla zapelniajaca kazdy bajt xmm1 wartoscia zmiany jasnosci
23         PADDB XMM1, XMM2        ; dodanie xmm2 na ostatnie 8 bitow xmm1
24         PSLLDQ XMM1, 1          ; przesuniecie rejestru xmm1 bitowe o 1 bajt w lewo
25         ADD EAX, 1              ; inkrementacja licznika petli
26         CMP EAX, 16             ; sprawdzanie warunku koncowego petli (16 iteracji - rejestr xmm1 - 128 bitow = 16 bajtow)
27         JL L1                  ; powrot do poczatk petli
28         PADDB XMM1, XMM2        ; ostatnie dodanie xmm2 do xmm1
29         PSUBUSB XMM0, XMM1      ; odejmowanie rejestru zawierajacego piksele z rejestrem zawierajacym wartosci zmiany jasnosci
30         MOVDQU [RCX], XMM0     ; załadowanie nowych wartości pikseli z powrotem do tablicy
31         RET
32 DimASM ENDP
33 END
34
```