

Operacje bezkontekstowe

Akceleracja Algorytmów Wizyjnych

Jan Rosa 12.03.2025

Kopia obrazu wejściowego (bez zmian)

Kod kopiuje obraz wejściowy do obrazu wyjściowego bez żadnych zmian.

```
__kernel void sobel_filter(__read_only image2d_t inputImage, __write_only
image2d_t outputImage)
{
    int2 coord = (int2)(get_global_id(0), get_global_id(1));

    float4 pixel = (float4)(0);
    //float4 Gy = Gx;
    //if( coord.x >= 1 && coord.x < (get_global_size(0)-1) && coord.y >= 1 &&
    coord.y < get_global_size(1) - 1)
    //{

    pixel = convert_float4(read_imageui(inputImage, imageSampler, (int2)
(coord.x, coord.y)));

    write_imageui(outputImage, coord, convert_uint4(pixel));
    //}
}
```

Opis:

get_global_id(0) i get_global_id(1) pobierają współrzędne piksela w obrazie.
read_imageui odczytuje wartość piksela ze współrzędnych (coord.x, coord.y).
convert_float4 konwertuje wartość na typ float4.
write_imageui zapisuje odczytaną wartość do obrazu wyjściowego.



Ekstrakcja składowych barwnych RGB

```
pixel = convert_float4(read_imageui(inputImage, imageSampler, (int2)(coord.x,  
coord.y)));  
pixel[3] = 0;  
pixel[1] = 0;  
pixel[2] = 0;
```

Kod ustawia wartości kanałów R, G na 0, co oznacza usunięcie informacji o tych kolorach.



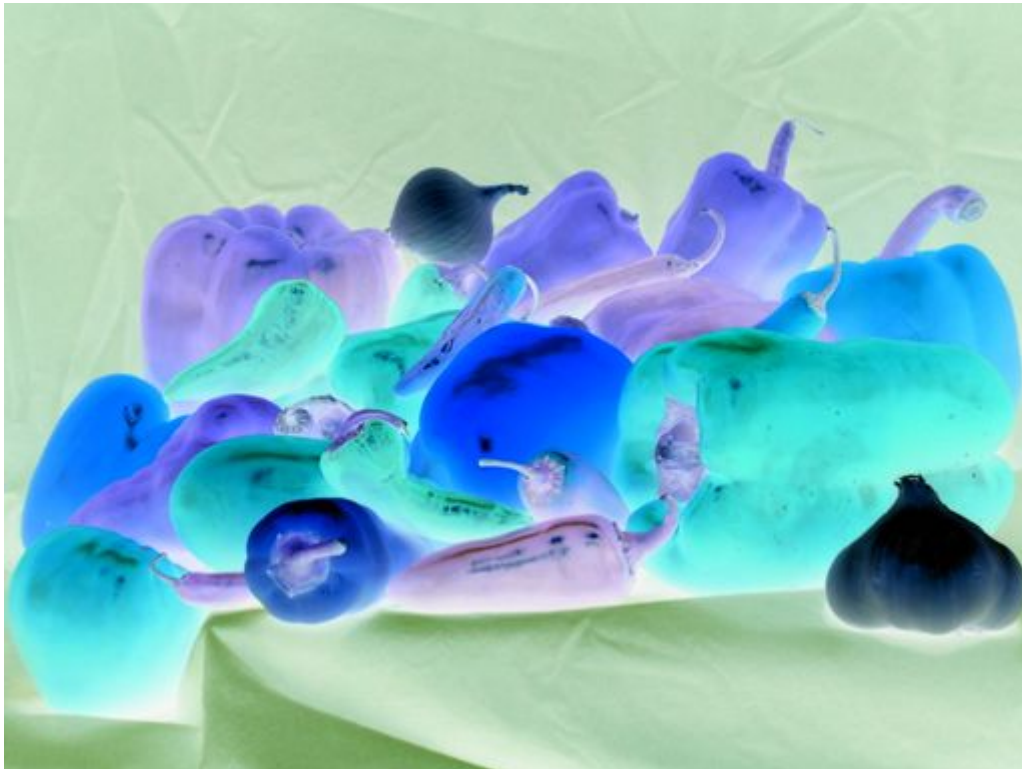


Negacja obrazu kolorowego,

```
pixel = (uint4)(255) - read_imageui(inputImage, imageSampler, (int2)(coord.x,  
coord.y));
```

Opis:

- Odczytujemy wartość piksela.
- Odejmujemy tę wartość od 255, co odwraca kolory (np. czarny staje się biały, niebieski – żółty itp.).



Binaryzacja obrazu kolorowego z jednym progiem,

```
pixel = 255*step(128.0, pixel);
```

Opis:

- `step(128.0, pixel)` zwraca 1.0, jeśli `pixel >= 128`, lub 0.0, jeśli `pixel < 128`.
- Wynik jest mnożony przez 255, co daje efekt binaryzacji.



Konwersja przestrzeni barw z kolorowej do monochromatycznej

```
float4 conv_val = (float4){0.2989, 0.5870, 0.1140, 0.0};  
pixel = dot(pixel, conv_val);
```

Opis:

Współczynniki {0.2989, 0.5870, 0.1140} reprezentują wagę składowych R, G i B w jasności piksela.

dot(pixel, conv_val) oblicza sumę ważoną składowych, co daje wartość w odcieniach szarości.



Wzmocnienie obrazu monochromatycznego z nasyceniem

```
pixel = dot(pixel, conv_val);  
pixel = clamp(2*pixel, 0, 255);
```

Opis:

- `dot(pixel, conv_val)` przekształca obraz do skali szarości.
- `2 * pixel` zwiększa jasność dwukrotnie.
- `clamp(..., 0, 255)` ogranicza wartości do zakresu `[0, 255]`, aby uniknąć przepełnienia.



write_imageui

```
void write_imageui (aQual image2d_t image, int2 coord, uint4 color)
```

Funkcja wbudowana używana przez kernel do zapisu części obrazu przez kernele.

Opis:

- `image2d_t image` – obraz, do którego zapisujemy dane.
- `int2 coord` – współrzędne piksela.
- `uint4 color` – zapisany kolor piksela.