

## Ćwiczenie 1

### FILTRACJA OBRAZU

#### Zakres pracy

W ramach ćwiczenia należy do istniejącego programu dodać możliwość wykonywania następujących przekształceń obrazu:

- liniowej filtracji splotowej przy użyciu predefiniowanych masek o rozmiarach 3×3 (uśredniająca, Gaussa, Sobela, laplasjanu, wyostrzająca),
- filtracji medianowej przy użyciu maski 3×3, 5×5 oraz krzyżowej (5-pikselowej),
- wykrywania krawędzi przy użyciu operatora LoG sparametryzowanego wartością odchylenia standardowego, odpowiadającą za rozmiar maski i ustawianą w zakresie 0,6..2,4.

#### Informacje pomocnicze

##### 1. Filtry liniowe.

Działanie filtra liniowego można opisać wzorem:

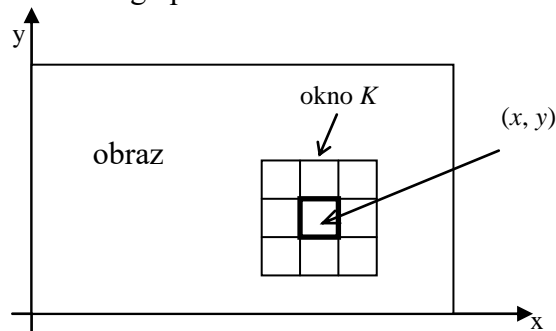
$$J_w(x, y) = \sum_{i, j \in K} J(x-i, y-j)w(i, j)$$

gdzie:  $K$  jest otoczeniem rozważanego piksela (w naszym przypadku 3×3),

$w(i, j)$  – wagi pikseli otaczających piksel  $(x, y)$ ,

$J(x, y)$  – jasność obrazu źródłowego.

Obliczenie jasności punktu w obrazie wynikowym przeprowadza się przez sumowanie elementów  $J(x-i, y-j)$  z odpowiednimi wagami  $w(i, j)$  wokół punktu  $(x, y)$ . Operację tę powtarza się oddzielnie dla każdego piksela obrazu.



Przyjmuje się, że wagi  $w(i, j)$  są liczbami całkowitymi. Jeśli wszystkie współczynniki spełniają warunek  $w(i, j) \geq 0$ , stosuje się następującą normalizację:

$$J_w(x, y) = \frac{\sum_{i, j \in K} J(x-i, y-j)w(i, j)}{\sum_{i, j \in K} w(i, j)}.$$

W przypadku, gdy niektóre ze współczynników  $w(i, j)$  mają wartości ujemne, do obrazu wynikowego wprowadzamy przesunięcie jasności, tak by w wyświetlanej bitmapie  $J_w = 0$  odpowiadało poziomowi 127 (wartości wychodzące poza zakres obcinamy).

Maski filtrów:

- uśredniająca
 

1	1	1
1	1	1
1	1	1
  
- Gaussa
 

1	4	1
4	12	4
1	4	1
  
- Sobela
 

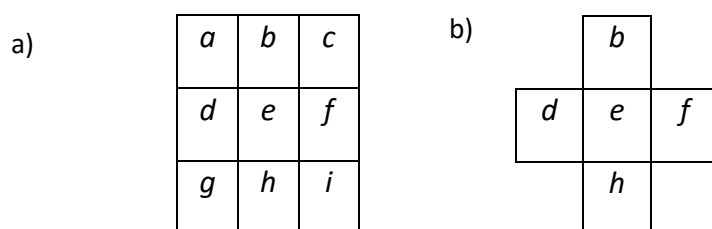
1	2	1	-1	0	1
0	0	0	-2	0	2
-1	-2	-1	-1	0	1
  
- laplasjanu
 

-2	1	-2
1	4	1
-2	1	-2
  
- wyostrzająca
 

0	-1	0
-1	5	-1
0	-1	0

## 2. Filtr medianowy.

Mediana jest wartością środkową w monotonicznie uporządkowanym ciągu jasności pikseli, które na obrazie pokrywa maska filtra. Typowe maski używane do oczyszczania obrazów za pomocą filtrów medianowych pokazano poniżej.



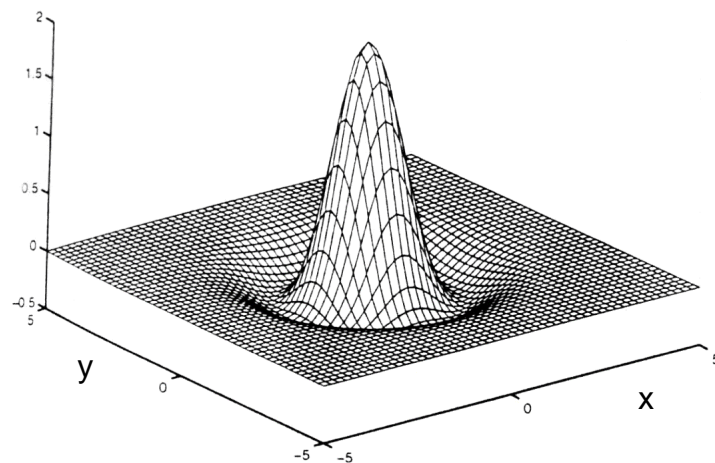
## 3. Filtr LoG.

Funkcja *LoG* (*Laplacian of Gaussian*) jest opisana następującym wzorem:

$$LoG(x, y) = -\left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}\right) \cdot e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

gdzie  $\sigma$  jest parametrem zmieniającym się w granicach 0,6..2,4.

Wykres funkcji LoG to tzw. sombrero (*Mexican hat*):



Maska filtra LoG powinna aproksymować kształt powyższej funkcji. Należy ją tworzyć dynamicznie, w zależności od zadanej przez użytkownika wartości parametru  $\sigma$ . Szerokość maski  $\Delta$  wyznacza się ze wzoru

$$\Delta = 1 + 2 \lceil 2.5\sigma \rceil.$$

Przyjmujemy, że w tym przypadku maska będzie zawierać wagi o wartościach zmiennoprzecinkowych, przeskalowanych tak, by ich suma wynosiła (w przybliżeniu) zero.

### Wskazówki implementacyjne

- Podczas implementowania filtrów medianowych warto skorzystać z funkcji `std::sort()`, np.:

```
#include <algorithm>
vector<int> vNeighbors;
sort(vNeighbors.begin(), vNeighbors.end());
```

- Aby uzyskać prawidłowy efekt mapy krawędzi po zastosowaniu filtra LoG, należy najpierw dodać do wyniku wartość 127 (jak w przypadku filtrów górnoprzepustowych), a następnie zaznaczyć w kolorze czarnym (na białym tle) tylko te punkty, dla których jasność jest nadal mniejsza od 0.