

## Filtracja w domenie przestrzeni

### 1 Filtracja liniowa 2D

Filtracja liniowa 2D jest definiowana jako splot obrazu z maską. Podobnie jak w przypadku filtracji 1D, te same warunki odnośnie sumy i wartości elementów są narzucone na maskę przy danej filtracji.

Do filtracji służy polecenie `imfilter(obraz, maska, opcje)`; Dostępne opcje są pogrupowane w 3 kategoriach:

- Opcje związane z brzegiem obrazu:
  - **X** - punkty leżące na zewnątrz obszaru, a pokryte maską, mają przypisaną wartość X. Domyślnie X=0;
  - 'symmetric' - potrzebne obszary są tworzone poprzez odbicie symetryczne
  - 'replicate' - potrzebne obszary są tworzone poprzez kopiowanie brzegowych wartości
  - 'circular' - potrzebne obszary są doklejane cyklicznie
- Opcje związane z rozmiarem wyjściowym:
  - 'same' - obraz wyjściowy ma taki sam rozmiar jak wejściowy
  - 'full' - obraz wyjściowy jest większy niż wejściowy
- algorytm filtracji
  - 'corr' - filtracja jest dokonywana z wykorzystaniem funkcji korelacji.
  - 'conv' - wielowymiarowa filtracja z wykorzystaniem splotu.

#### 1.1 Filtracja dolnoprzepustowa

Przypisują one elementowi centralnemu średnią z obszaru pokrytego maską. Stosuje się do tego maski typu:  $MF = \text{ones}(3)/9$ ,  $MF = \text{ones}(5)/25$ . Działanie tego rodzaju filtru powoduje na ogół rozmycie krawędzi, generalizację obrazu oraz zawężenie zakresu intensywności. Cechą charakterystyczną tej filtracji jest, że suma elementów maski równa się 1. Najczęściej spotykane maski bazują na średniej ważonej, gdzie waga elementu centralnego jest większa od pozostałych, np.  $MF = [1 \ 1 \ 1; 1 \ 2 \ 1; 1 \ 1 \ 1]/10$ .

#### 1.2 Filtracja górnoprzepustowa

Filtracja górnoprzepustowa służy do wyeksponowania detali i zmienności na obrazach, takich jak krawędzie, narożniki, pojedyncze punkty. Wykorzystuje się do tego pierwsze i drugie pochodne intensywności. Najczęściej używanymi operatorami są: gradienty Roberta, Sobela czy Prewitta oraz laplasjany.

##### 1.2.1 Operator Prewitta

Operator ten jest oparty o pierwszą pochodną. Wykorzystuje on następujące maski (po lewej maska horyzontalna, po prawej wertykalna):

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

### 1.2.2 Operator Sobela

Operator ten jest oparty o uśrednioną pochodną dyskretną, gdzie wartość pochodnej liczonej przez środek ma dwukrotnie większą wagę niż pochodne liczone z elementów sąsiadujących. Wykorzystuje on następujące maski (po lewej maska horyzontalna, po prawej wertykalna):

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

### 1.2.3 Operator Kirscha

Operator Kirscha polega na przypisaniu elementowi centralnemu wartości maksymalnej z filtracji ośmioma maskami, obróconymi względem siebie o  $45^\circ$ . Pozwala to lepiej wyodrębnić maksymalną amplitudę z ośmiu kierunków filtracji. Pierwsze trzy maski dane są następującymi tablicami:

$$MF^0 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad MF^{45} = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad MF^{90} = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$$

### 1.2.4 Laplasjany

Lapsjan jest kombinacją drugich pochodnych cząstkowych funkcji  $I(m,n)$ : (wyostrzające, gradientowe):

$$L(x, y) = \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2} \quad (1)$$

Suma elementów maski laplasjanów może być równa 0 (wykrywanie krawędzi) lub 1 (wyostrzenie obrazu). Jest to filtracja izotropowa. Do najczęściej spotykanych masek laplasjanów zaliczamy:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 0.25 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.25 \\ 0.5 & 0.5 & 0 & -0.5 & 0 & 0.5 & 0.5 \\ 0.5 & 0 & -0.75 & -1.5 & -0.75 & 0 & 0.5 \\ 0.5 & -0.5 & -1.5 & -2 & -1.5 & -0.5 & 0.5 \\ 0.5 & 0 & -0.75 & -1.5 & -0.75 & 0 & 0.5 \\ 0.5 & 0.5 & 0 & -0.5 & 0 & 0.5 & 0.5 \\ 0.25 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.25 \end{bmatrix}$$

## 2 Filtracja nieliniowa

Istnieje również cała gama filtrów, których działanie zależy od np. uporządkowania obrazu. Są to np. filtry medianowe, maksymalizujące, minimalizujące czy entropii, które przypisują filtrowanemu punktowi odpowiednio medianę, maksimum, minimum i entropię z obszaru pokrytego maską. Większość z tych filtrów 2D umożliwia filtrację sygnałów

1D. W tym celu należy ustawić wartość szerokości (dla sygnałów pionowych) lub wysokości (dla sygnałów poziomych) maski na 1 lub stworzyć pionowy/poziomy wektor złożony z wartości 0/1. W pakiecie MatLAB zostały zdefiniowane następujące metody filtracji:

- `medfilt1(sygnal, rozmiar)`; - filtracja medianowa 1D
- `medfilt2(obraz, [rozmiar])`; - filtracja medianowa 2D
- `ordfilt2(obraz, nr, zasięg)`; - zwraca wartość znajdującą się na pozycji nr w uporządkowanym wektorze (1 - minimum, wielkość maski - maksimum).
- `rangefilt(obraz, zasięg)` - zwraca różnicę pomiędzy maksimum i minimum intensywności dla pikseli otoczenia zdefiniowanych w tablicy *zasięg*, np. `ones(3)`.
- `entropyfilt(obraz, zasięg)`; - zwraca wartość entropii E w blokach o zadanym rozmiarze (domyślnie 9x9 pikseli), obliczoną wg wzoru:

$$E = - \sum_{n=0}^N \log p(n) \cdot p(n) \quad (2)$$

N - ilość poziomów intensywności

p(n) - prawdopodobieństwo wystąpienia piksela o intensywności n

- `stdfilt(obraz, zasięg)` - dla każdego piksela zwraca wartość odchylenia standardowego z otoczenia.

## 2.1 Funkcja edge

Do wykrywania naroży może również służyć funkcja `edge(obraz, 'metoda', parametry)`. Funkcja ta na wejściu przyjmuje obraz monochromatyczny, a zwraca mapę logiczną, gdzie wartość logiczną "1" ma wykryty narożnik. Funkcja ta posiada 6 metod detekcji krawędzi, zarówno liniowych, jak i nieliniowych (Sobel, Prewitt, Roberts, LoG - Laplasjan Gaussa, zerocross, Canny).

Jedną z ważniejszych metod jest filtracja Canny: `edge(obraz, 'canny', [ $\theta_1, \theta_2$ ], sigma)`. W metodzie tej  $\theta_1$  określa dolny próg odcięcia,  $\theta_2$  określa górny. Jeżeli podany jest tylko jeden próg, to jest on traktowany jako górny, a dolny jest liczony według wzoru:  $\theta_1 = 0.4 \cdot \theta_2$ . Sigma jest to standardowe odchylenie dla filtru gaussowskiego. Domyślnie  $\sigma = 1$ . Filtr Canny działa w kilku krokach:

1. Wygładzanie obrazu za pomocą filtru Gaussa o wariancji  $\sigma^2$
2. Wyznaczenie obrazu krawędziowego prostym operatorem krawędziowym w obszarze 2x2 lub 3x3
3. Cieniowanie krawędzi metodą tłumienia niemaksymalnego - zerowanie elementów krawędziowych niebędących maksimumami lokalnymi
4. Cieniowanie krawędzi metodą tłumienia z progami histerezy  $\theta_1, \theta_2$

Dla wszystkich tych metod, poleceniem `[BW, próg]=edge(...)` można uzyskać informacje o użytym progu detekcji krawędzi. Dla metod gradientowych (Sobel, Prewitt, Roberts) automatyczny próg binaryzacji jest liczony na podstawie magnitudy policzonego gradientu.

## 2.2 VMF - Vector median flow

Jest to filtracja stosowana wyłącznie dla obrazów wielokanałowych (np. RGB, CMYK, satelitarne). Polega ona na zastąpieniu wartości piksela centralnego przez wartość koloru

leżącego najbliżej w przestrzeni kolorów. Jeżeli w masce istnieje piksel o identycznej wartości koloru jak piksel centralny, to wartość piksela nieulega zmianie. Odległość dwóch wartości kolorów w danej przestrzeni najczęściej liczona jest z użyciem normy L1 lub L2.

### 3 Filtracja adaptatywna (Wienera)

Osobną kategorią filtrów nieliniowych są filtry adaptacyjne. Są to filtry, których charakterystyka zmienia się w zależności od analizowanego obszaru. W celu usuwania szumu działają one dwuetapowo:

1. Dla każdego punktu i jego otoczenia obliczamy wartość parametru, który kwalifikuje dany punkt jako należący lub nie do krawędzi. W przypadku filtru Wienera jest to wariancja w obszarze obrazu pokrytym maską.
2. jeżeli dany punkt został zakwalifikowany jako nie należący do krawędzi, zostaje on poddany silnemu uśrednieniu. W przeciwnym wypadku jego wartość pozostaje bez zmian lub poddany zostaje uśrednieniu o niewielkiej mocy.

Do filtrów tego typu zaliczamy filtr Wienera: `wiener2(obraz, [rozmiar])`.

### 4 Dekonwolucja

Operacją odwrotną do splotu (czyli m.in. filtracji) jest dekonwolucja. Istnieje kilka technik dekonwolucji zaimplementowanych w MatLABie:

- Dekonwolucja ślepa: `deconvblind`
- Dekonwolucja przy użyciu algorytmu Lucy-Richardsona: `deconvlucy`
- Dekonwolucja przy użyciu algorytmu Wienera: `deconvwnr`
- Dekonwolucja "unormowana": `deconvreg`

### 5 Szum

Do "zaszumiania" obrazów służy polecenie `imnoise(obraz, typ, parametry)`. Podstawowymi rodzajami szumu są:

- 'gaussian' - szum o rozkładzie gaussa z średnią  $m$  i wariancją  $\sigma^2$ . Domyślnie  $m=0$  i  $\sigma^2=0.01$ ;
- 'localvar' - szum gaussowski o średniej  $m=0$  i wariancji zależnej od otoczenia punktu. Aby użyć tej funkcji należy jako parametr dodać tablicę rozkładu wariancji o rozmiarze równym zaszumionemu obrazowi.
- 'poisson' - szum o rozkładzie Poissona
- 'salt & pepper' - zamienia wartość piksela na minimum lub maximum z dostępnej skali z gęstością zamiany  $d$ . Domyślnie  $d=0.05$ ;
- 'speckle' - multiplikatywny szum zmieniający wartość piksela zgodnie z równaniem:

$$L(m, n) = L(m, n) + n \cdot L(m, n) \quad (3)$$

gdzie  $n$  - liczba z rozkładu normalnego o średniej  $m=0$  i wariancji  $\sigma^2$ . Domyślnie  $\sigma^2=0.04$ ;

## 6 Zadania

1. Stwórz obraz (czarne tło, 200x200), gdzie centralnie jest położony znak "+" zbudowany z prostokątów o wymiarze 100x20 px. Stwórz filtrację wykrywającą wszystkie narożniki.
2. Wczytaj obraz *pout.tif* i dokonaj jego normalizacji. Następnie dodaj do niego szum (przetestuj wszystkie pięć rodzajów szumu). Stwórz filtrację odsumiającą. Zaprojektuj metodę oceny skuteczności procedury odsumiania.
3. Wczytaj obraz *peppers.png*. Każdą z warstw RGB przefiltruj uśredniającą filtracją Gaussa o zmiennym odchyleniu i rozmiarze (R -  $5 \times 5$ ,  $\sigma = 1$ ; G -  $9 \times 9$ ,  $\sigma = 2$ , B -  $13 \times 13$ ,  $\sigma = 2.5$ ). Na jednej figurze wyświetl wynik przed i po filtracji.
4. Wczytaj obraz *circles.png*. Dokonaj jego rozmycia poprzez trzykrotną filtrację maskami z zadania 3. Dokonaj dekonwolucji dowolną metodą stosując tylko jedną z tych masek. Porównaj wyniki dekonwolucji.