

LABORATORIUM CYFROWEGO PRZETWARZANIA SYGNAŁÓW

Ćwiczenie 4

Cyfrowe przetwarzanie obrazów

1. Cel ćwiczenia

- Poznanie metod reprezentacji w dziedzinie częstotliwości obrazów traktowanych jako sygnały dwuwymiarowe.
- Przedstawienie podstawowych metod filtracji obrazów i ich właściwości.

2. Podstawy teoretyczne

Przy cyfrowym przetwarzaniu sygnałów jednowymiarowych (np. akustycznych) rolę zmiennej niezależnej (indeksu) pełnił unormowany czas. W przypadku przetwarzania obrazów (np. fotografii) mamy do czynienia z sygnałami dwuwymiarowymi, w których zmienne niezależne mają sens położenia na płaszczyźnie. Jest to szczególny przypadek sygnałów wielowymiarowych, w których zmienne niezależne mogą mieć różny sens, a najczęściej są to współrzędne przestrzenne. W ćwiczeniu nie będziemy zajmować się zmianami obrazów w czasie. Przebiegi czasowe, np. sygnałów wizyjnych (luminancji i chrominancji), stanowią sygnały jednowymiarowe i do ich przetwarzania stosuje się metody omówione w poprzednich ćwiczeniach.

Jeżeli dwuwymiarowy sygnał analogowy zapiszemy jako $x_a(t_1, t_2)$, gdzie t_1, t_2 są współrzędnymi przestrzennymi, to w rezultacie próbkowania w dwóch wymiarach z częstotliwościami próbkowania f_{s1}, f_{s2} otrzymuje się sygnał dyskretny

$$x(m, n), \quad \text{gdzie } m=t_1/T_{s1}, T_{s1}=1/f_{s1} \text{ oraz } n=t_2/T_{s2}, T_{s2}=1/f_{s2} \quad (4.1)$$

Okresy próbkowania T_{s1}, T_{s2} mają w tym przypadku wymiar *długości*, a indeksy m, n są *odległościami unormowanymi względem okresów próbkowania*.

Próbkę $x(m, n)$ obrazu zdyskretyzowanego należy interpretować nie jako punkt geometryczny, ale *element obrazu* o wymiarach ($T_{sx}T_{sy}$ – tzw. *piksel* (ang. *pixel* – *picture element*). Najczęściej piksele mają kształt kwadratów o boku T (gdy okresy próbkowania w obu wymiarach są jednakowe: $T_{sx}=T_{sy}=T$). Wartość próbki ma sens *jasności* (lub *natężenia barwy*) i jest zwykle zapisywana na 8 bitach (256 poziomów). Przy takim podejściu częstotliwość próbkowania jest tożsama z *rozdzielczością* obrazu, np. przy $f_s=300 \times 600$ dpi (*dots per inch* – punktów na cal) okresy próbkowania są równe: $T_{sx}=1/300$ cala, $T_{sy}=1/600$ cala.

Aby obraz analogowy można było odtworzyć z próbek, częstotliwości próbkowania f_{s1}, f_{s2} muszą być – zgodnie z twierdzeniem o próbkowaniu i odtwarzaniu sygnału – co najmniej 2 razy większe od najwyższych częstotliwości występujących w widmie sygnału analogowego (tzn. piksele muszą być odpowiednio małe).

2.1. Widmo obrazu dyskretnego

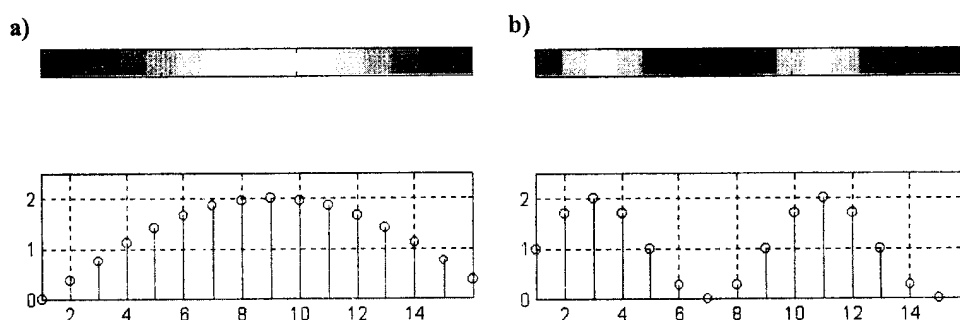
Widmo Fourierowskie dwuwymiarowego sygnału dyskretnego $x(m, n)$ jest określone wzorem:

$$X(e^{j\Omega_x}, e^{j\Omega_y}) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(m, n) \cdot e^{-jm\Omega_x} \cdot e^{-jn\Omega_y} \quad (4.2)$$

gdzie $\Omega_x = \omega_x T_{sx} = \omega_x / f_{sx}$, $\Omega_y = \omega_y T_{sy} = \omega_y / f_{sy}$ oznaczają pulsacje unormowane względem częstotliwości próbkowania. **Częstotliwość przestrzenną** $f = \omega / 2\pi$ określa się jako *liczbę zmian wartości jasności na jednostkowym odcinku dowolnego fragmentu obrazu*.

Wynika z tego, że w widmie obrazu występują wysokie częstotliwości, gdy na małych odległościach następują duże zmiany jasności (duże różnice wartości pikseli). Fragmenty obrazu, w których zmiany jasności przebiegają łagodnie zawierają składowe widma o niskich częstotliwościach (Rys. 1). Duże częstotliwości przestrzenne wiążą się zatem z występowaniem w obrazie małych i kontrastowych detali, natomiast małe częstotliwości wynikają z istnienia dużych i w miarę jednolitych obszarów.

Rozumowanie dotyczące częstotliwości nie zależy od orientacji analizowanego fragmentu względem krawędzi obrazu. *Widmo przestrzenne jest niezmiennicze względem ustawienia (obrotu) obrazu*, co m.in. ułatwia rozpoznawanie obrazów w dziedzinie częstotliwości. Jednakowe częstotliwości przestrzenne obrazów dwuwymiarowych tworzą na wykresie współśrodkowe okręgi o środkach w punkcie odpowiadającym częstotliwości zerowej.



Rys. 1. Linia obrazu stanowiąca sygnał o małej (a) i dużej częstotliwości (b) oraz przebiegi zmian wartości pikseli (jasności) wzdłuż linii. Analogiczną analizę można przeprowadzić dla linii ustawionej pod dowolnym kątem

Dwuwymiarowe dyskretne $M \times N$ -punktowe przekształcenie Fouriera obrazu $x(m, n)$ definiuje się jako wynik próbkowania widma ciągłego (4.2) w punktach $\Omega_{xk} = 2\pi k/M$, $k=0, 1, \dots, M-1$ i $\Omega_{yl} = 2\pi l/N$, $l=0, 1, \dots, N-1$:

$$X(k, l) = DFT[x(m, n)] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) \cdot e^{-jm\Omega_{xk}} \cdot e^{-jn\Omega_{yl}} \quad (4.3)$$

Odwrotne dwuwymiarowe dyskretne przekształcenie Fouriera umożliwia jednoznaczne odtworzenie pikseli obrazu na podstawie współczynników widma dyskretnego (4.3):

$$x(m, n) = IDFT[X(k, l)] = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} X(k, l) \cdot e^{jm\Omega_{xk}} \cdot e^{jn\Omega_{yl}} \quad (4.4)$$

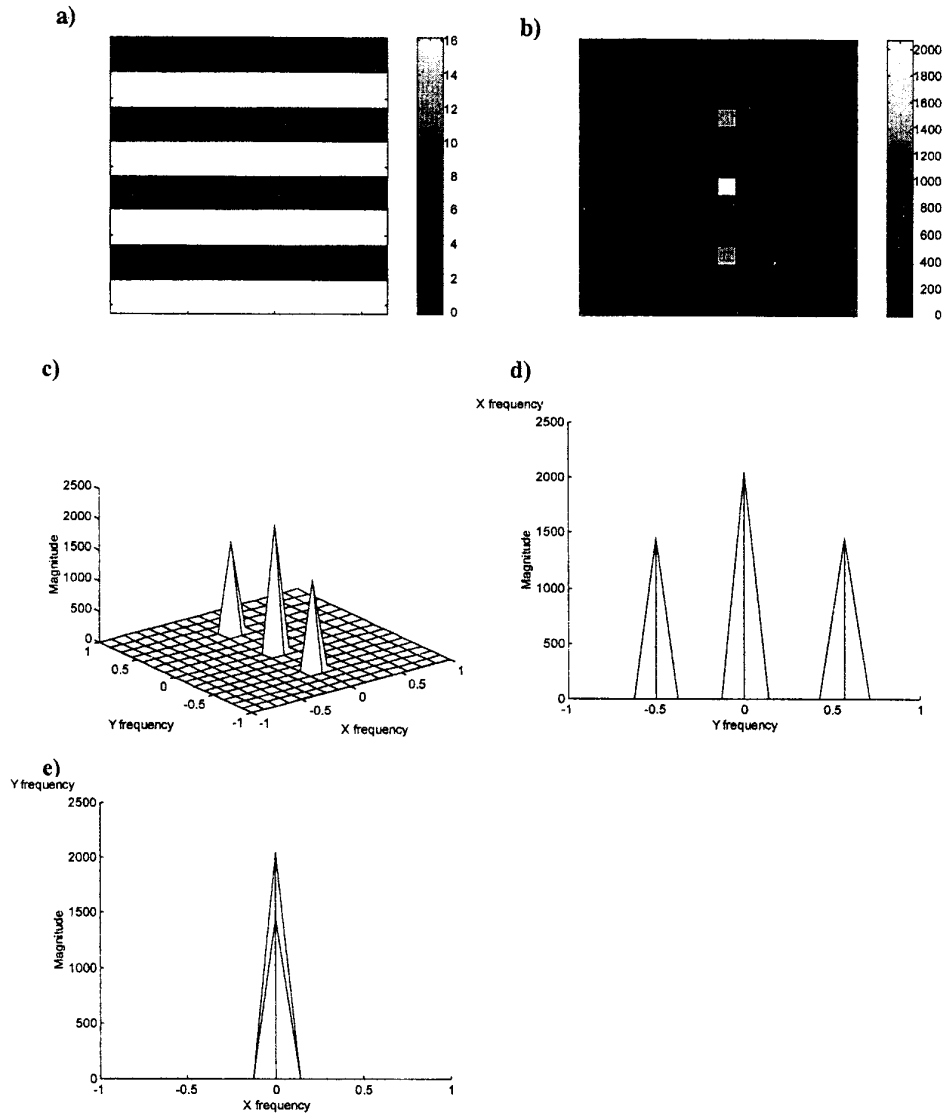
Moduł $|X(k, l)|$ stanowi **dyskretne widmo amplitudowe obrazu**.

2.2. Filtracja liniowa obrazu

Wynik filtracji sygnału dwuwymiarowego $x(m, n)$ zapisuje się jako spłot tego sygnału z dwuwymiarową charakterystyką impulsową $h(m, n)$ filtra:

$$y(m, n) = x(m, n) * h(m, n) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x(k, l) \cdot h(m-k, n-l) \quad (4.5)$$

Charakterystykę impulsową filtra definiuje się przy tym jako jego odpowiedź na pobudzenie dwuwymiarową funkcją delta Kroneckera



Rys. 2. Obraz poziomych czarno-białych pasów (a) i jego widmo (b,c). Na wykresie widma oglądanego prostopadłe do osi OY (d) widać dwa boczne prążki odpowiadające częstotliwości pasów. Na wykresie widma oglądanego prostopadłe do osi OX (e) widać tylko składową stałą, bo jasność obrazu w poziomie nie zmienia się.

$$\delta(m, n) = \begin{cases} 1 & \text{dla } m = n = 0 \\ 0 & \text{dla pozostałych par } (m, n) \end{cases} \quad (4.6)$$

przy zerowych warunkach początkowych.

W przypadku obrazów możliwa jest realizacja filtracji *nieprzyczynowej*, tzn. za pomocą filtrów, których $h(m, n)$ ma wartości niezerowe dla ujemnych m i/lub n . Wartość piksela po filtracji zależy wtedy od wartości pikseli sąsiednich, zarówno o indeksach mniejszych jak i większych od przetwarzanego.

Ze względu na efektywność analizy i projektowania filtracja liniowa obrazów ogranicza się w zasadzie do filtrów o skończonej odpowiedzi impulsowej SOI (FIR) o zerowej charakterystyce fazowej $\varphi(\Omega) \equiv 0$. Granice sumowania we wzorze (4.5) są w takim przypadku skończone. Skończoną charakterystykę impulsową $h(m, n)$ w postaci macierzy będziemy nazywać *maską* (albo *jądrem*) filtra.

Jeżeli $x(m, n)$ jest obrazem o wymiarach $M \times N$, a $h(i, j)$ jest maską filtra o wymiarach $I \times J$ (M, I – liczby wierszy, N, J – liczby kolumn), to wartość splotu obrazu z maską w punkcie (m, n) wyznacza się z zależności:

$$y(m, n) = \sum_{i=-a}^a \sum_{j=-b}^b x(m+i, n+j) \cdot h(i, j) \quad (4.7)$$

gdzie a =część całkowita($I/2$), b =część całkowita($J/2$), przy czym współrzędne (0,0) ma lewy górny punkt obrazu i *środkowy* punkt maski filtra.

Przykład 1: Dla podanych wartości pikseli obrazu x i maski h (centralny punkt $h(0,0)=8$) wartość splotu w punkcie $y(3,2)$ będzie wynosić:

$x=$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>3</td><td>7</td><td>6</td><td>12</td><td>...</td></tr> <tr><td>8</td><td>7</td><td>10</td><td>6</td><td>...</td></tr> <tr><td>14</td><td style="background-color: #cccccc;">13</td><td>8</td><td>5</td><td>...</td></tr> <tr><td>1</td><td>12</td><td>5</td><td>2</td><td>...</td></tr> <tr><td>2</td><td>8</td><td>2</td><td>6</td><td>...</td></tr> <tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr> </table>	3	7	6	12	...	8	7	10	6	...	14	13	8	5	...	1	12	5	2	...	2	8	2	6
3	7	6	12	...																											
8	7	10	6	...																											
14	13	8	5	...																											
1	12	5	2	...																											
2	8	2	6	...																											
...																											

$h=$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td style="background-color: #cccccc;">8</td><td>-1</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td></tr> </table>	-1	-1	-1	-1	8	-1	-1	-1	-1
-1	-1	-1								
-1	8	-1								
-1	-1	-1								

$$y(3,2)=8 \cdot (-1)+7 \cdot (-1)+10 \cdot (-1)+14 \cdot (-1)+13 \cdot 8+8 \cdot (-1)+1 \cdot (-1)+12 \cdot (-1)+5 \cdot (-1)=39$$

Ponieważ w punktach przy krawędziach obrazu maska filtra wystaje poza obraz, przyjmuje się jedną z następujących możliwości wyznaczenia splotu:

- wartości pikseli poza obrazem są równe zero. Obraz będący wynikiem splotu ma te same wymiary co oryginał, ale na brzegach powstają zafałszowania,
- splot oblicza się tylko w tych punktach, w których maska nie wystaje poza obraz. Obraz po filtracji jest mniejszy od oryginalnego,
- uzupełnia się brakujące wartości, np. metodą lustrzanego odbicia względem krawędzi obrazu. Obraz po filtracji ma wymiary oryginału, a zafałszowania są mniejsze niż w pierwszym przypadku.

2.3. Filtracja obrazu w dziedzinie częstotliwości

Transmitancję filtra liniowego definiuje się jako *dwuwymiarową dyskretną transformatę Z* jego odpowiedzi impulsowej:

$$H(u, v) = Z[h(m, n)] = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h(m, n) \cdot u^{-m} \cdot v^{-n} \quad (4.8)$$

Korzystając z zapisu operatorowego zależność (4.5) można przedstawić w formie transmitancyjnej:

$$Y(u, v) = H(u, v) \cdot X(u, v) \quad (4.9)$$

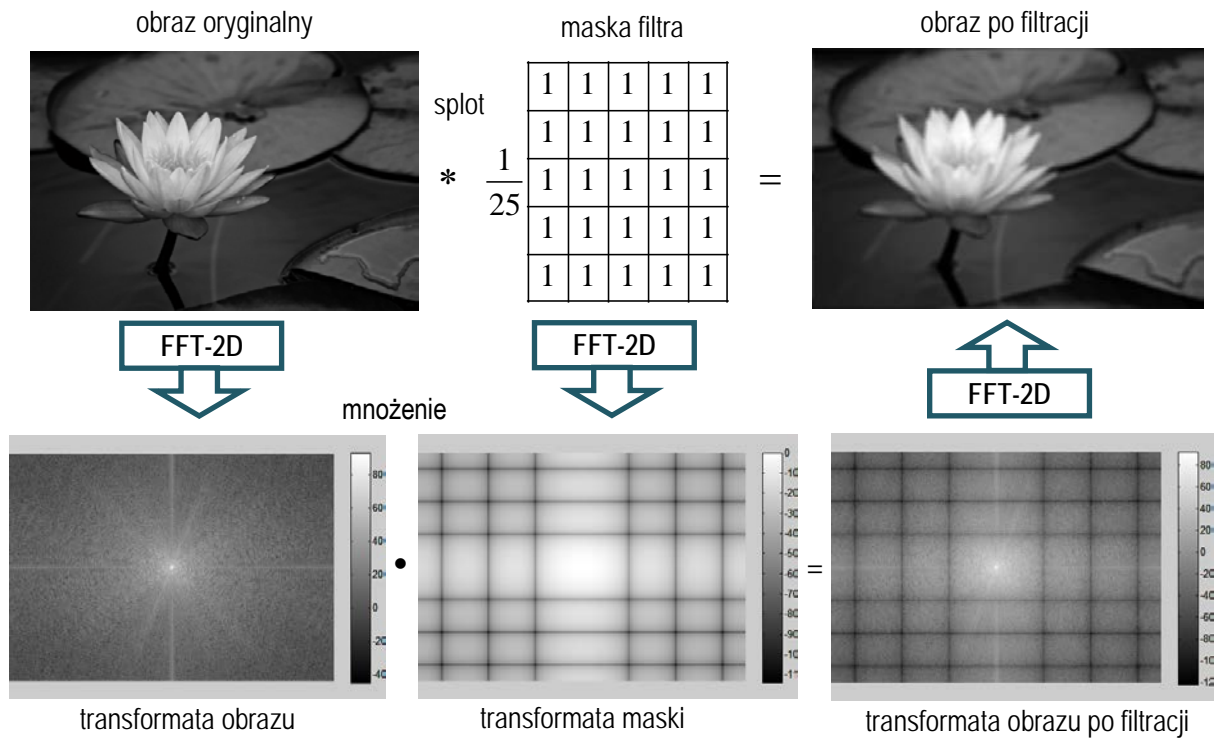
gdzie: $X(u, v)=Z[x(m, n)]$ i $Y(u, v)=Z[y(m, n)]$ (warunki początkowe zerowe).

Jeżeli $x(m,n)$ jest obrazem o wymiarach $M \times N$, a $h(i,j)$ jest maską filtra o wymiarach $I \times J$, to dyskretna transformata obrazu po filtracji jest opisana zależnością:

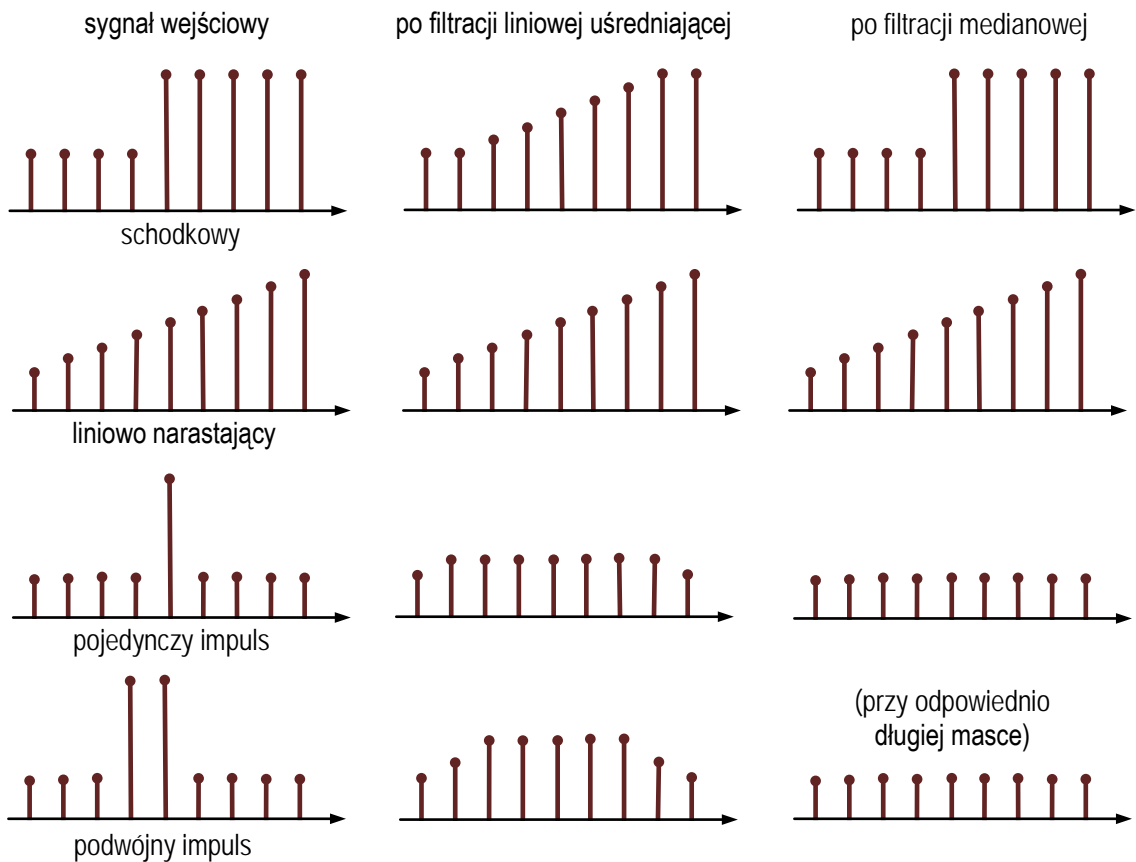
$$Y(k, l) = H(k, l) \cdot X(k, l) \quad (4.10)$$

gdzie $H(k,l)=DFT[h(i,j)]$, $X(k,l)=DFT[x(i,j)]$ są $K \times L$ -punktowymi dyskretnymi transformatami Fouriera, tzn. $k=0,1,\dots,K-1$, $l=0,1,\dots,L-1$. Filtracji w dziedzinie obrazu realizowanej poprzez operację splotu odpowiada mnożenie transformat obrazu i filtra w dziedzinie częstotliwości (Rys. 3). Ponieważ obraz po filtracji uzyskuje się przez wyznaczenie odwrotnej transformaty Fouriera $y(m,n)=IDFT[Y(k,l)]$, więc wymiar transformaty $DFT[x(i,j)]$ powinien być równy wymiarowi obrazu.

Projektowanie charakterystyki filtra w dziedzinie częstotliwości pozwala na modyfikowanie widma przetwarzanego obrazu, np. usuwanie określonych częstotliwości stanowiących okresowe zakłócenia (powtarzające się wzory na obrazie, np. pasy).



Rys. 3. Filtracja w dziedzinie obrazu (splot) i częstotliwości (mnożenie transformat)



Rys. 4. Ilustracja działania filtra medianowego w przypadku jednowymiarowym. Filtr zachowuje krawędź sygnału schodkowego i usuwa zakłócenia impulsowe krótsze niż połowa rzędu (wielkości okna) filtra

2.4. Nieliniowy filtr medianowy

Filtr medianowy jest jednym z najczęściej stosowanych filtrów nieliniowych. *Medianą* dla wycinka sygnału o długości $N=2L+1$ próbek jest $L+1$ (środkowa) próbka w ciągu powstałym po uporządkowaniu próbek w porządku rosnącym ich wartości (*tyle samo próbek jest mniejszych lub równych i większych lub równych od mediany*; dla parzystej liczby próbek można wziąć średnią z dwóch środkowych). Np. dla próbek o wartościach 22, 8, 2, 5, 1 mediana ma wartość 5. Filtracja polega więc na uszeregowaniu wartości próbek znajdujących się w oknie filtra (lub masce w przypadku filtra dwuwymiarowego) i przypisaniu sygnałowi wyjściowemu wartości środkowej. *Rozmiar maski jest jedynym parametrem filtra medianowego.*

Filtr medianowy można określić jako nieliniowy filtr dolnoprzepustowy, ale w odróżnieniu od liniowych filtrów dolnoprzepustowych ma dobre właściwości jeśli chodzi o zachowywanie krawędzi (stąd określenie *edge-preserving filter*). Jego zaletą jest też całkowite usuwanie silnych zakłóceń impulsowych, podczas gdy filtry liniowe takie zakłócenia tylko uśredniają (Rys. 4).

Przykład 2: Jeżeli dwuwymiarowy filtr medianowy ma maskę $h(i,j)$ o wymiarach 3×3 , to dla punktu $x(3,2)$ obrazu z przykładu 1 w obręb maski wchodzi piksele, których uszeregowanie daje wartości: 1, 5, 7, 8, 8, 10, 12, 13, 14. W wyniku filtracji medianowej otrzymujemy wartość środkową ciągu, tzn. $y(3,2)=8$.

2.5. Korelacja dwuwymiarowa

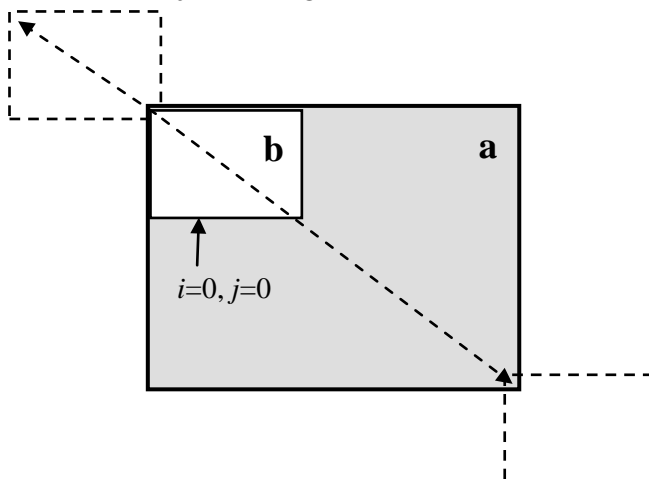
Dwuwymiarową korelację wzajemną obrazów (sygnałów dwuwymiarowych) $x(m,n)$ i $y(m,n)$ określonych w postaci macierzy \mathbf{a} o wymiarach $M_a \times N_a$ i wzorca \mathbf{b} o mniejszych wymiarach $M_b \times N_b$ (indeksowanie od zera) określa się jako:

$$R_{ab}(i,j) = \frac{1}{M_b N_b} \sum_{m=0}^{M_a-1} \sum_{n=0}^{N_a-1} x(m,n) \cdot y(m+i, n+j)$$

$$-(M_b - 1) \leq i \leq M_a - 1$$

$$-(N_b - 1) \leq j \leq N_a - 1$$
(4.11)

Macierz korelacji \mathbf{R}_{ab} jest macierzą o wymiarach $(M_a+M_b-1) \times (N_a+N_b-1)$. Obliczanie korelacji polega na przesuwaniu jednego obrazu („wzorca”) na tle drugiego. Współczynniki macierzy korelacji osiągają największe wartości dla przesunięć (i,j) , dla których wzorec jest podobny do odpowiedniego fragmentu obrazu. Autokorelacja \mathbf{R}_{aa} osiąga maksimum dla $R_{aa}(0,0)$, kiedy macierze \mathbf{a} pokrywają się.



Rys. 5. Sposób obliczania korelacji wzajemnej dwóch macierzy. Dla każdego przesunięcia $R_{ab}(i,j)$ oblicza się jako sumę iloczynów nakładających się wyrazów macierzy \mathbf{a} i \mathbf{b} .

2.6. Transformata kosinusowa DCT

Transformatę kosinusową (*Discrete Cosine Transform* – DCT) dyskretnego sygnału jednowymiarowego $x(n)$ definiuje się jako:

$$DCT[x(n)] = X_c(k) = p(k) \sum_{n=0}^{N-1} x(n) \cdot \cos \frac{(2n+1)k\pi}{2N}, \quad k = 0, 1, \dots, N-1 \quad (4.12)$$

gdzie: $p(k) = \begin{cases} \sqrt{1/N} & \text{dla } k = 0 \\ \sqrt{2/N} & \text{dla } k \neq 0 \end{cases}$.

Dla sygnałów rzeczywistych transformata DCT ma również wartości rzeczywiste. W porównaniu z transformatą Fouriera DFT transformata kosinusowa ma lepsze właściwości skupiania energii sygnału (suma kwadratów niewielu początkowych współczynników DCT reprezentuje większość energii sygnału). W DCT nie ma symetrii współczynników jak w DFT, *wszystkie są od siebie niezależne*.

Odwrotna transformata kosinusowa:

$$x(n) = IDCT[X_c(k)] = p(n) \sum_{k=0}^{N-1} X_c(k) \cdot \cos \frac{(2k+1)n\pi}{2N}, \quad n = 0, 1, \dots, N-1 \quad (4.13)$$

Współczynniki N -punktowej transformaty kosinusowej $DCT_N[x_o(n)]$ są równe pierwszym N współczynnikom $2N$ -punktowej transformaty Fouriera $DFT_{2N}[x_2(n)]$ wektora

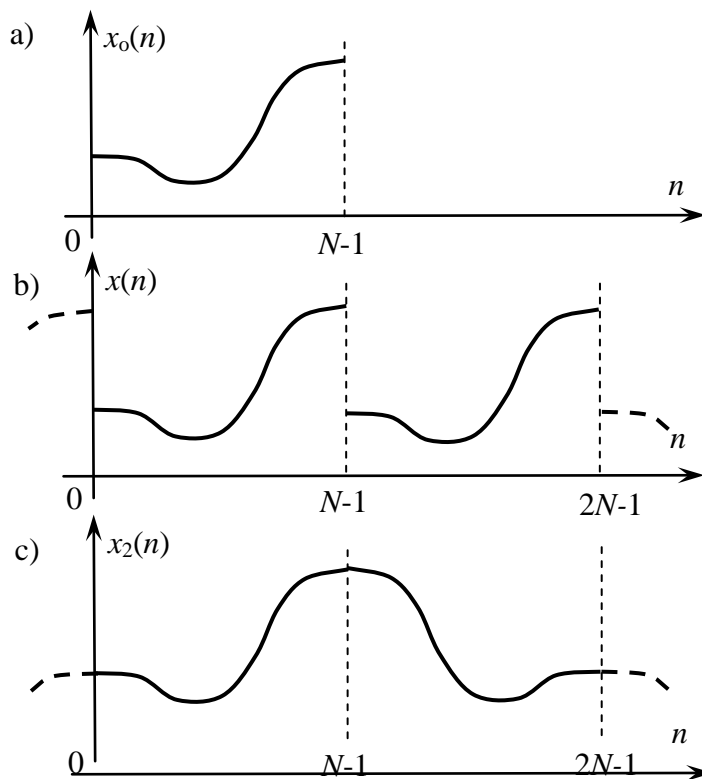
$$x_2(n) = \begin{cases} x(n) & \text{dla } 0 \leq n \leq N-1 \\ x(2N-n-1) & \text{dla } N \leq n \leq 2N-1 \end{cases} \quad (4.14)$$

stanowiącego lustrzanie symetryczną wersję 2 wycinków powielonego okresowo $x_o(n)$ (rys. 6), tj.

$$X_c(k) = X_F(k), \quad k = 0, 1, \dots, N-1, \quad (4.15)$$

gdzie: $X_c(k) = DCT_N[x(n)]$, $X_F(k) = DFT_{2N}[x_2(n)]$. (Transformata Fouriera symetrycznego wycinka $x_2(n)$ ma współczynniki rzeczywiste).

Zatem transformata kosinusowa wycinka sygnału z nieciągłościami na końcach daje taki sam wynik jak transformata Fouriera sygnału bez nieciągłości. $DFT_N[x_o(n)]$ wycinka z nieciągłościami zawiera rozmycie widma wprowadzające fałszywe składowe o wyższych częstotliwościach. Tej wady nie ma transformata DCT.



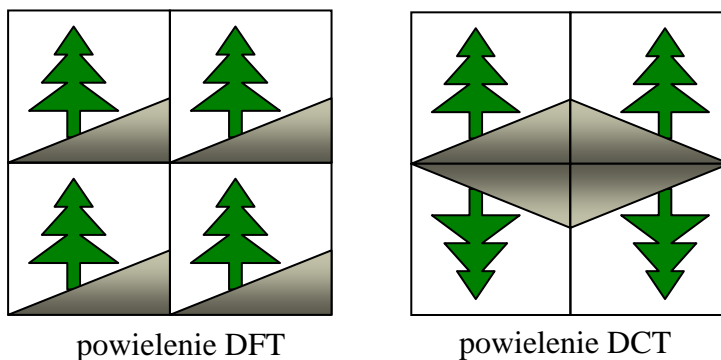
Rys. 6. Okresowe powielenie odcinka sygnału (a) przez transformatę Fouriera (b – z nieciągłościami) i transformatę kosinusową (c – bez nieciągłości)

Dwuwymiarową transformatę kosinusową (DCT2) definiuje się jako:

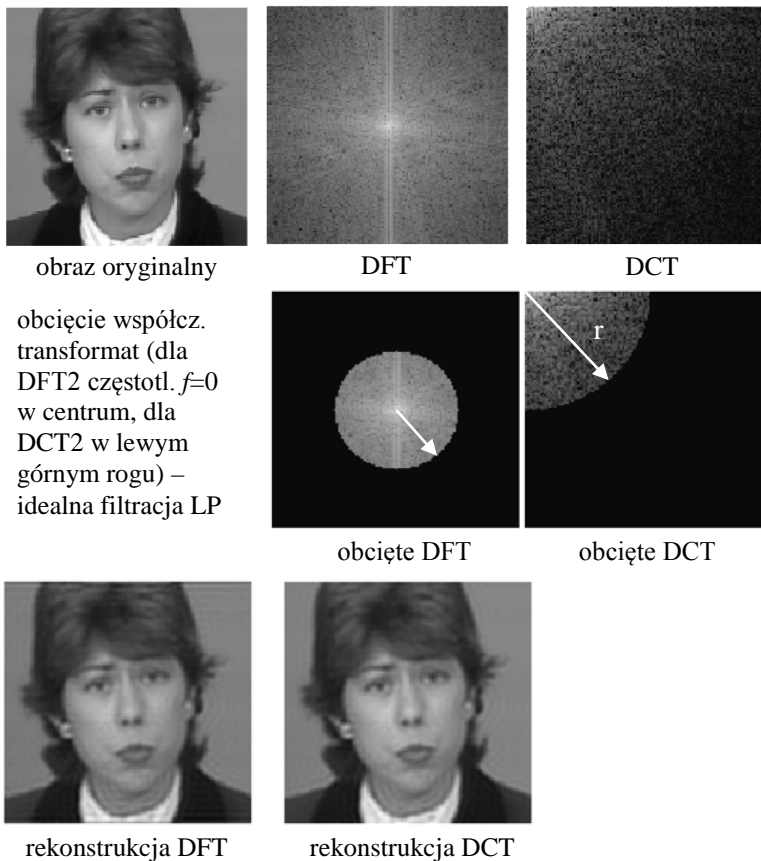
$$\begin{aligned}
 DCT[x(m,n)] &= X_c(k,l) = \\
 &= p(k)q(l) \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n) \cos \frac{(2m+1)k\pi}{2M} \cos \frac{(2n+1)l\pi}{2N}, \quad (4.16) \\
 &0 \leq k \leq M-1, \quad 0 \leq l \leq N-1
 \end{aligned}$$

gdzie współczynniki $p(k) = \begin{cases} \sqrt{1/M} & \text{dla } k = 0 \\ \sqrt{2/M} & \text{dla } k \neq 0 \end{cases}$, $q(l) = \begin{cases} \sqrt{1/N} & \text{dla } l = 0 \\ \sqrt{2/N} & \text{dla } l \neq 0 \end{cases}$.

DCT2 jest szeroko stosowana np. przy kompresji obrazów. Ze względu na to, że transformata kosinusowa traktuje blok obrazu (obraz jest zwykle dzielony na bloki 8x8 pikseli) jako wycinek bez nieciągłości na brzegach (rys. 7), wprowadza ona mniej tzw. „artefaktów blokowych”. Zniekształcenia przy krawędziach bloków dla DFT2 są bardziej widoczne, szczególnie po kompresji (np. obcięciu części współczynników).



Rys. 7. Okresowe powielenie dwuwymiarowego bloku obrazu przez DFT2 i DCT2



Rys. 7. Transformacja obrazu i rekonstrukcja po obcięciu części współczynników poza okręgiem r

3. Obliczenia komputerowe - zadania do wykonania wg instrukcji

Uwaga:

- Bloki instrukcji Matlab'a do realizacji poszczególnych zadań wygodnie jest kopiować do edytora Matlab'a i tam dokonywać edycji parametrów, dopisywania nowych linii itp., a następnie uruchamiać jako skrypty komendą **Run** ▷.
 - Przy przechodzeniu do następnego punktu ćwiczenia zwracać uwagę, żeby nie dochodziło do ewentualnego nakładania się na siebie w tym samym oknie wykresów różnych wielkości. Normalnie rysowanie nowego wykresu kasuje poprzednią zawartość otwartego okna, o ile ta zawartość nie została przeznaczona do zachowania za pomocą instrukcji `hold on` (domyślne `hold off` kasuje poprzednią zawartość pola wykresu w otwartym oknie). Wstawianie `hold on` jest potrzebne, kiedy instrukcja zaleca nakładanie na siebie dla porównania kilku wykresów tej samej wielkości otrzymanych przez powtarzanie jakiegoś bloku instrukcji dla różnych wartości pewnych parametru(-ów). Po zakończeniu danego zadania takie okna wykresów należy zamykać.
 - Do projektowania filtrów wykorzystywane są funkcje biblioteki Matlab'a **Image Processing Toolbox**. Listing funkcji zapisanej w formie skryptu można otrzymać w oknie komend za pomocą instrukcji `type nazwa_funkcji`.
- W oknie **Current Folder** przejść do folderu **Moje dokumenty/MATLAB/DSP**. Jest to folder roboczy dla ćwiczeń z *Cyfrowego przetwarzania sygnałów*.

3.1. Wyznaczanie widma obrazu cyfrowego jako sygnału dwuwymiarowego

- Do obliczania i wykreślenia wykresów dwuwymiarowego przekształcenia DFT obrazów (bitmap w skali szarości) zapisanych w macierzy x będzie wykorzystywana funkcja `dft2`.

Sprawdzić, czy plik funkcji `dft2.m` znajduje się w folderze roboczym. Jeżeli nie, poniższy listing funkcji należy skopiować do edytora Matlab'a i zapisać w folderze roboczym w pliku `dft2.m`.

```
function X = dft2(x,opt)
% Dwuwymiarowe widmo DFT obrazu i wykres 3D amplitudy widma.
% Częstotliwość unormowana -1<=fn<=1, fn=f/fNyq
% Parametry wejśc.: x - obraz (MxN pikseli), opt (opcja) -> skala log (dB)
% Wyjście: X = dft2(x) (składowa DC w centrum), dft2(x) -> rys. wykresy
X = fftshift(fft2(x)); % FFT2(x), zamiana ćwiartek -> składowa DC w centrum
if nargin==0 % bez argumentu wyjściowego -> wykresy DFT2
    [f1,f2] = freqspace(size(x)); % wektory częstotl. unorm. -1<=fn<1
    % Rozszerzenie wyniku X i wekt. [f1,f2] dla sklejenia brzegów: -1<=fn<=1
    [ny,nx]=size(x);
    if mod(ny,2)==0, ny1=ny+1; f2=[f2 1]; else ny1=ny; end
    if mod(nx,2)==0, nx1=nx+1; f1=[f1 1]; else nx1=nx; end
    X1 = zeros(ny1,nx1);
    X1(1:ny,1:nx)=X;
    if mod(ny,2)==0, X1(ny1,1:nx)=X(1,:); end
    if mod(nx,2)==0, X1(1:ny,nx1)=X(:,1); end
    if mod(ny,2)==0 & mod(nx,2)==0, X1(ny1,nx1)=X(1,1); end
    X = X1;
    [f1,f2] = meshgrid(f1,f2); % siatka unorm. częstotl. do wykresu 3D
    figure % Widmo amplitudowe 3D
    if nargin<2
        mesh(f1,f2,abs(X)),
        zlabel(' |X(f_{nx}, f_{ny})| ');
        title('DFT2(x) Magnitude, f_n= f / f_{Nyq}')
    else
        mesh(f1,f2,20*log10(abs(X))),
        zlabel(' 20 log |X(f_{nx}, f_{ny})| (dB) ');
        title('DFT2(x) Magnitude (dB), f_n= f / f_{Nyq}')
```

```

end
colormap(cool), colorbar, view(135,45); % azimuth, elevation (stopnie)
xlabel('f_{nx}'), ylabel('f_{ny}')
figure % Mapa barwna 2D widma amplitud.
if nargin<2
    imagesc([-1 1],[-1 1],abs(X)),
    xlabel('f_{nx}'), ylabel('f_{ny}'),
    title('DFT2(x) Magnitude, f_n= f / f_{Nyq}')
else
    imagesc([-1 1],[-1 1],real(20*log10(X))),
    xlabel('f_{nx}'), ylabel('f_{ny}'),
    title('DFT2(x) Magnitude (dB), f_n= f / f_{Nyq}')
end
colorbar, colormap(jet)
end
end

```

- W przypadku informacji o braku powyższej funkcji należy ją utworzyć w edytorze Matlaba i zapisać w folderze roboczym w formie pliku `dft2.m`.
- Wywołanie funkcji z argumentem wyjściowym: `x=dft2(x)` zapisuje widmo DFT2 obrazu `x` (obliczone za pomocą funkcji `fft2` Matlaba) w macierzy `X` ze składową DC w centrum (bez wykreślania wykresów).
Wywołanie bez lewostronnego argumentu wyjściowego `dft2(x)` wykreśla w dwóch nowych oknach wykresy 3D i mapę barw 2D amplitudy widma w skali liniowej.
Wywołanie z dowolnym drugim argumentem `dft2(x,opt)` wykreśla wykresy amplitudy widma w decybelach (skala logarytmiczna).
Na wykresie 2D wartość amplitudy widma jest reprezentowana za pomocą kolorów określonej mapy barw `colormap`. Związana z nią skala wartości jest pokazana na słupku `colorbar`.
- A. Wygenerować obraz ze wzorem w formie pionowych pasów z szarymi przejściami od bieli do czerni o zadanej częstotliwości przestrzennej f_n (gdzie $f_n=1$ odpowiada częstotliwości Nyquista: 2 piksele na okres).

```

N=[60, 60]; % wymiary macierzy obrazu
x=zeros(N);
fn=0.1; % unormowana częstotliwość wzoru
for j=1:N(2),
    x(:,j)=0.5+0.5*sin(pi*fn*j); % wzór: pionowe pasy (fale)
end;
imagesc(x), colormap(gray), colorbar % wzór w skali szarości
dft2(x); % wykresy amplitudy widma

```

- Powtórzyć obliczenia dla $f_n=0.5$.
- Zwrócić uwagę na zmianę położenia niezerowych prążków widma w zależności od f_n . Prążek w centrum widma reprezentuje składową stałą DC sygnału obrazu (piksele mają wartości od 0 do 1).
- Obejrzyć widma na wykresie 3D patrząc prostopadłe do osi `OX` i `OY` z poziomu zerowego.
Uaktywnić przycisk **Rotate 3D** z belki narzędzi okna wykresu i obracać go myszką do uzyskania współrzędnych widoku (*azimuth-elevation*):
`Az=90, El=0` % przebieg widma wzdłuż osi `OY` (oś `OX` prostopadła do ekranu)
`Az=0, El=0` % przebieg widma wzdłuż osi `OX` (oś `OY` prostopadła do ekranu)
- Przeprowadzić obliczenia dla pasów pionowych czarno-białych (przebieg prostokątny).

```

...
fn=0.1; % unormowana częstotliwość wzoru
for j=1:N(2),
    x(:,j)=0.5+0.5*sign(sin(pi*fn*j)); % wzór: pionowe czarno-białe pasy
end;
...

```

➤ Co reprezentują niższe niezerowe prążki widma ?

B. Wygenerować obraz ze wzorem w formie ukośnych pasów z szarymi przejściami od bieli do czerni i wyjaśnić położenie niezerowych prążków widma na płaszczyźnie częstotliwości.

```
...
fn=0.2;           % unormowana częstotliwość wzoru
for i=1:N(1),
    for j=1:N(2),
        x(i,j)=0.5+0.5*sin(pi*fn*(i-j)); % pasy ukośne
    end;
end;
...
```

C. Wygenerować obraz ze wzorem w formie kraty z szarymi przejściami od bieli do czerni i wyjaśnić położenie niezerowych prążków widma na płaszczyźnie częstotliwości.

```
...
fn=0.2;           % unormowana częstotliwość wzoru
for i=1:N(1),
    x(i,:)=0.5+0.5*sin(pi*fn*i);       % wzór: poziome pasy (fale)
end;
x = (x + x')/2; % suma poziomych i pionowych (transpozycja x) pasów
...
```

➤ Powtórzyć obliczenia dla wzoru kraty z ostrymi przejściami i porównać wynik z uzyskanym poprzednio.

```
...
for i=1:N(1),
    x(i,:)=0.5+0.5*sign(sin(pi*fn*i));
end;
x = (x + x')/2;
...
```

D. Wygenerować obraz ze wzorem w formie współśrodkowych kolistych fal z szarymi przejściami od bieli do czerni i skomentować wzór widma.

```
...
fn=0.2;           % unormowana częstotliwość wzoru
for i=1:N(1),
    for j=1:N(2),
        x(i,j)=0.5+0.5*sin(pi*fn*sqrt((round(N(1)/2)-i)^2+...
            (round(N(2)/2)-j)^2)); % współśrodkowe okręgi
    end;
end;
...
```

➤ Zarejestrować wykresy widma w skali decybelowej. Czy widoczność szczegółów jest lepsza?

```
dft2(x,1);       % wykresy amplitudy widma w dB
```

➤ Powtórzyć obliczenia dla okręgów czarno-białych i obejrzyć wykresy widma w skali liniowej i logarytmicznej i skomentować je.

```
...
for j=1:N(2),
    x(i,j)=0.5+0.5*sign(sin(pi*fn*sqrt((round(N(1)/2)-i)^2+...
        (round(N(2)/2)-j)^2))); % współśrodkowe czarno-białe okręgi
end;
...
```

3.2. Filtracja liniowa 2D w dziedzinie obrazu

A. Utworzyć macierz maski $h(m,n)$ nieprzyczynowego liniowego filtra uśredniającego LP, wyznaczyć charakterystykę częstotliwościową jego wzmocnienia i przeprowadzić filtrację wczytanego obrazu.

Uwaga: Podana nazwa jest nazwą przykładową, każda grupa ćwiczeniowa powinna wybrać własny obraz(y) z folderu roboczego. W razie potrzeby należy zmieniać wymiary okien graficznych z wynikami w celu zachowania proporcji obrazów.

```
x = imread('flower.jpg'); imshow(x),
disp('Wymiary obrazu w pikselach:'), size(x)
x = rgb2gray(x);
N = [5 5]; % wymiary maski filtra
h = ones(N)/sum(sum(ones(N))) % zdefiniowanie maski LP z sumą współcz. =1
xf = imfilter(double(x),h); % filtracja obrazu oryginalnego z wart. double
xf = uint8(xf); % powrót do formatu uint8
H = freqz2(h); % ch-ka częstotliwościowa 2-wym. maski h filtra
figure
subplot(121), imagesc(x), colormap(gray),
title('obraz oryginalny')
subplot(122), imagesc(xf), colormap(gray),
title('obraz po filtracji LP')
figure
subplot(121), imagesc([-1 1],[-1 1],abs(H)), axis('square')
colormap(cool), colorbar
title('Wzmocnienie |H(f_x,f_y)| filtra'), xlabel('f_x'), ylabel('f_y')
subplot(122), freqz2(h), zlabel('|H(f_x,f_y)|')
```

- W dokumentacji **Help** Matlaba wyszukać informacje na temat funkcji `imfilter` oraz `filter2` realizujących filtrację dwuwymiarową. Jaką metodą (za pomocą wyznaczania korelacji czy splotu (konwolucji) obrazu i maski) realizowana była filtracja obrazu? W oknie **Workspace** sprawdzić zakres wartości zmiennych obrazów `x` i `xf`.
- Powtórzyc filtrację uśredniającą z filtrem o większej masce: `N=[9 9]`.
- Porównać i przedyskutować wyniki filtracji oraz charakterystyki amplitudowe filtrów. Jakie jest wzmocnienie filtrów LP dla małej i dużej częstotliwości przestrzennej?

B. Przeprowadzić operację zakłócania obrazu przez dodanie szumu *białego o rozkładzie Gaussa* oraz szumu *impulsowego 'salt and pepper'* (nadawanie losowo wybranym pikselom wartości maksymalnej lub minimalnej):

```
...
xg = imnoise(x,'gaussian',0,0.01); % szum Gaussa o średniej m=0 i var=0.01
xsp = imnoise(x,'salt & pepper',0.03); % 3% szumu impulsowego
```

i dokonać filtracji zakłóconych obrazów za pomocą filtrów uśredniających LP

```
N = [5 5];
h = ones(N)/sum(sum(ones(N))) % zdefiniowanie maski f. uśredniającego LP
xfg = imfilter(xg,h); % filtracja obrazu z szumem Gaussa
xfsp = imfilter(xsp,h); % filtracja obrazu z szumem 'salt&pepper'
subplot(221), imshow(xg), title('obraz szumem Gaussa')
subplot(222), imshow(xfg), title('po filtracji f. uśredniającym')
subplot(223), imshow(xsp), title('obraz z szumem 'salt & pepper')
subplot(224), imshow(xfsp), title('po filtracji f. uśredniającym')
```

- Powtórzyc filtrację uśredniającą z filtrem o większej masce: `N=[9 9]`.

C. Przeprowadzić filtrację liniową filtrem LP Gaussa $h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$ wykorzystując do wyznaczenia maski funkcję `fspecial` tworzenia predefiniowanych filtrów 2D (Zobacz **Help** funkcji `fspecial`).

```
N = [9 9]; % wymiary maski filtra
sig = 2; % odchylenie standardowe sigma funkcji Gaussa
h = fspecial('gaussian',N,sig) % zdefiniowanie maski filtra Gaussa
xf = imfilter(double(x),h); % filtracja obrazu oryginalnego z wart. double
```

```

xf = uint8(xf);      % powrót do formatu uint8
H = freqz2(h);      % ch-ka częstotliwościowa 2-wym. maski h filtra
figure
subplot(121), imagesc(x), colormap(gray),
title('obraz oryginalny')
subplot(122), imagesc(xf), colormap(gray),
title('obraz po filtracji f. Gaussa')
figure
subplot(121), imagesc([-1 1],[-1 1],abs(H)), axis('square')
colormap(cool), colorbar
title('Wzmocnienie |H(f_x,f_y)| filtra'), xlabel('f_x'), ylabel('f_y')
subplot(122), freqz2(h), zlabel('|H(f_x,f_y)|')

```

- Porównać maskę i charakterystykę amplitudową filtra Gaussa z maską filtra uśredniającego. W oknie komend sprawdzić sumę współczynników maski filtra Gaussa.

```
sum(sum(h))
```

- Dokonać filtracji obrazów z zakłóceniami za pomocą zaprojektowanego filtra Gaussa LP

```

xgf = imfilter(xg,h); % filtracja obrazu z szumem Gaussa
xspf = imfilter(xsp,h); % filtracja obrazu z szumem 'salt&pepper'
subplot(221), imshow(xg), title('obraz szumem Gaussa')
subplot(222), imshow(xgf), title('po filtracji f. Gaussa')
subplot(223), imshow(xsp), title('obraz z szumem 'salt & pepper')
subplot(224), imshow(xspf), title('po filtracji f. Gaussa')

```

- D.** Przeprowadzić operację liniowej filtracji obrazu za pomocą filtra symulującego ruch obiektu względem obiektu i powodującego rozmycie (*blur*) obrazu.

```

leng = 15;          % długość ruchu w pikselach (dł. przekątnej maski)
theta = 10;        % kąt ruchu w stopniach (ruch ukośny)
h = fspecial('motion',leng,theta) % wyznaczenie maski
xf = imfilter(double(x),h);
...
subplot(122), imagesc(xf), colormap(gray),
title('obraz po filtracji f. rozmywającym')
...

```

- Zwrócić uwagę na kształt charakterystyki amplitudowej. Wyjaśnić wymiary maski filtra (jak mają się one do długości *leng*?) oraz wzór utworzony przez niezerowe współczynniki maski filtra (jak ma się on do kąta *theta*?).

- E.** Zdefiniować maskę i przeprowadzić liniową filtrację HP obrazu.

```

h = [-1 -1 -1; -1 8 -1; -1 -1 -1] % maska 3x3 filtra HP
xf = imfilter(x,h);
...
subplot(122), imagesc(xf), colormap(gray),
title('obraz po filtracji HP')
...

```

- Zarejestrować wyniki i powtórzyć filtrację HP z filtrem o większej masce $N=[5 \ 5]$ definiowanej za pomocą algorytmu ogólnego.

```

N = [5 5];
h = -ones(N);
h(ceil(N(1)/2),ceil(N(2)/2))=N(1)*N(2)-1; % wartość środkowego współcz.
h % wyświetlenie maski
xf = imfilter(x,h);
...

```

- Porównać grubość i wyrazistość krawędzi obiektów obrazu wykrytych za pomocą użytych masek HP. Wyjaśnić to w kontekście kształtu i poziomu wzmocnienia charakterystyk amplitudowych obu filtrów w zakresie wysokich częstotliwości (bliskich jedności).

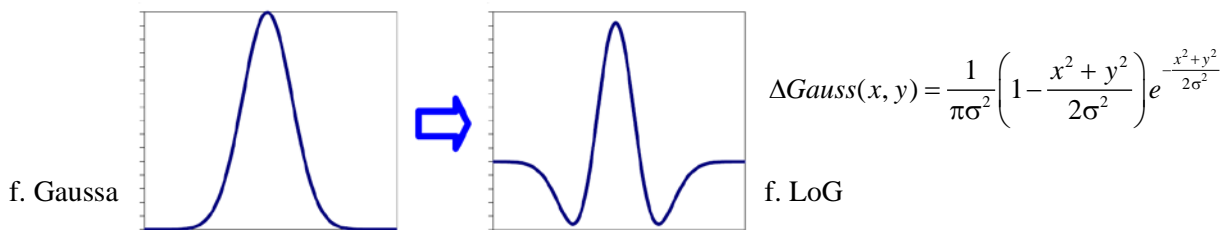
F. Wyznaczyć aproksymację laplasjanu obrazu uwydatniającego krawędzie (o dowolnej orientacji):

Maska filtra laplasjanu funkcji f_{special} ma postać:

$$h_{\Delta} = \frac{4}{\alpha+1} \begin{bmatrix} \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \\ \frac{1-\alpha}{4} & -1 & \frac{1-\alpha}{4} \\ \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \end{bmatrix}, \quad 0 \leq \alpha \leq 1$$

```
alpha = 1/4;
h = fspecial('laplacian',alpha) % wyznaczenie maski laplasjanu
xf = imfilter(double(x),h);
...
subplot(122), imagesc(xf), colormap(gray),
title('laplasjan obrazu')
...
```

- Porównać maskę laplasjanu z poprzednio stosowaną maską HP 3×3 i ich charakterystyki wzmocnienia. Powtórzyć obliczenia dla $\alpha=0.7$ i porównać wyniki.
- Wyznaczyć laplasjan obrazu za pomocą filtra LoG (*Laplacian of Gaussian*), tj. laplasjanu funkcji Gaussa.



```
N = [5,5]; sigma = 1.0; % parametry: rozmiar maski i odchylenie standard.
h = fspecial('log',N,sigma) % maska LoG
xf = imfilter(double(x),h);
...
subplot(122), imagesc(xf), colormap(gray),
title('obraz po filtracji LoG')
...
```

- Porównać maskę i charakterystykę wzmocnienia filtra LoG z odpowiednikami dla laplasjanu. W jakim zakresie częstotliwości filtr taki ma największe wzmocnienie? Czy można nazwać go filtrem pasmowoprzepustowym BP?

3.3. Wykrywanie krawędzi obiektów, wyostrenie obrazu, wyrównywanie histogramu

A. Wyznaczyć aproksymacje składowych gradientu obrazu oraz obraz czarno-biały będący efektem progowania logicznego za pomocą funkcji detekcji krawędzi `edge` (Zobacz **Help** funkcji `edge`).

```
x = imread('circuit.tif'); % obraz PCB
[xbws,thresh,gh,gv]=edge(double(x),'sobel'); % maski kierunkowe 'sobel'
disp('Próg progowania logicznego obrazu:')
thresh
figure
subplot(1,2,1), imshow(x), title('obraz PCB')
subplot(1,2,2), imshow(~xbws), title('obraz BW po progowaniu')
figure
subplot(1,2,1), imagesc(gv), colormap(gray), colorbar,
title('składowa grad_y')
subplot(1,2,2), imagesc(gh), colormap(gray), colorbar,
title('składowa grad_x')
```

- Jakie maski Sobela wyznaczają pionową i poziomą składową gradientu? Zwrócić uwagę na zakresy wartości macierzy gradientów i wyznaczoną automatycznie wartość progu rozdziału na wartości logiczne czarny-biały.
- Powtórzyć operację wykrywania krawędzi za pomocą masek Prewitta (`edge(..., 'prewitt')`) i porównać skuteczność detekcji.

B. Przeprowadzić operację wyostrzania krawędzi obrazu za pomocą odejmowania od obrazu jego laplasjanu dla zmiennych typu `uint8`:

```
x = imread('flower.jpg');
x = rgb2gray(x);
h = fspecial('laplacian') % maska laplasjanu 3x3
xlap = imfilter(x,h); % filtracja maską laplasjanu
xdif = imsubtract(x,xlap); % odjęcie laplasjanu od oryginału
subplot(1,3,1), imshow(x), title('obraz oryginalny')
subplot(1,3,2), imagesc(xlap), axis image, axis off
title('laplasjan obrazu'),
subplot(1,3,3), imagesc(xdif), axis image, axis off
title('oryginał - laplasjan')
```

C. Powtórzyć obliczenia dla macierzy przekonwertowanych do typu `double` i z dodatkowym współczynnikiem wagowym `c` odejmowanego laplasjanu:

```
xlap = imfilter(double(x),h); % filtracja maską laplasjanu
c = 0.8; % waga laplasjanu
xdif = imsubtract(double(x),c*xlap); % odjęcie laplasjanu
maxdif = max(max(xdif)) % max różnicy double
mindif = min(min(xdif)) % min różnicy double
subplot(1,3,1), imagesc(xlap), axis image, axis off
title('laplasjan (double)'),
subplot(1,3,2), imagesc(xdif), axis image, axis off
title('różnica (double)'),
subplot(1,3,3), imshow(uint8(xdif)),
title('różnica (uint8)')
```

- Porównać zakresy wartości macierzy różnicy `xdif` (wartości maksymalna i minimalna są podane w oknie komend) oraz `uint8(xdif)`. Dlaczego różnica typu `double` ma mniejszy kontrast?
- Ocenić zmiany ostrości obrazów otrzymanych w pkt. B i C.

D. Przeprowadzić poprawę jakości obrazu o niskim kontraście przez wyrównanie histogramu obrazu (*histogram equalization*).

```
x = imread('flower.jpg');
x = rgb2gray(x);
a = 0.5; b = 50;
xc = uint8(a*x+b); % zmniejszenie kontrastu i rozjaśnienie
xe = histeq(xc); % wyrównanie histogramu
subplot(121), imshow(xc), title('obraz o niskim kontraście')
subplot(122), imshow(xe), title('po wyrównaniu histogramu')
figure % histogramy
subplot(121), imhist(xc,128), title('histogram niskiego kontrastu')
subplot(122), imhist(xe,128), title('histogram wyrównany')
```

- Porównać histogramy przed i po wyrównaniu. Zwrócić uwagę na odstępy pomiędzy słupkami (*bins*) histogramu po wyrównaniu. Na powiększonych obrazach sprawdzić, czy nie występuje związany z tym brak płynnej zmiany intensywności (tzw. posteryzacja).
- Zapoznać się z opisami funkcji poprawy jakości obrazu: `imadjust` i `histeq` w dokumentacji **Help**.

3.4. Analiza i filtracja obrazu w dziedzinie częstotliwości

- A. Wyznaczyć widma amplitudowe DFT2 obrazu oryginalnego oraz obrazu po wyodrębnieniu przez odjęcie Laplasjanu jak w pkt. 3.2.C.

```
x = imread('flower.jpg');
x = rgb2gray(x);
h = fspecial('laplacian') % maska laplasjanu 3x3
xlap = imfilter(x,h); % filtracja maską laplasjanu
xdif = imsubtract(x,xlap); % odjęcie laplasjanu od oryginału
X = fftshift(fft2(double(x))); % FFT2 z zamianą ćwiartek, DC w centrum
Xdif = fftshift(fft2(double(xdif))); % widmo różnicy
[f1,f2] = freqspace(size(x),'meshgrid'); % siatka częstotl. -1<=f/fNyq<1
subplot(1,2,1), mesh(f1,f2,20*log10(abs(X))), % skala magnitudy w dB
xlabel('f_x'), ylabel('f_y'), zlabel('DFT2 Magnitude (dB)')
colormap(hot), colorbar, axis([-1 1 -1 1 40 100]), title('DFT2 obrazu')
subplot(1,2,2), mesh(f1,f2,20*log10(abs(Xdif))),
colorbar, axis([-1 1 -1 1 40 100]),
xlabel('f_x'), ylabel('f_y'), title('DFT2 różnicy x - x_{Lap}')
```

- Które widmo i dlaczego ma większą zawartość wysokich częstotliwości?

- B. Przeprowadzić filtrację obrazu w dziedzinie częstotliwości, tj. przez mnożenie transformat DFT2 obrazu i maski filtra $DFT2[x] \cdot DFT2[h]$, a następnie przeprowadzenie transformacji odwrotnej $IDFT2$ do dziedziny obrazu. Zwrócić uwagę, że wymiary macierzy transformat muszą być takie same, ponieważ mnoży się odpowiadające sobie współczynniki, które reprezentują te same częstotliwości przestrzenne.

```
x = imread('flower.jpg');
x = rgb2gray(x);
h = fspecial('average',[5 5]) % maska filtra uśredniającego
xf1 = imfilter(x,h,'conv'); % splot w dziedzinie czasu dla porównania
X = fft2(double(x)); % transformata DFT2 obrazu
[nx,ny]=size(X); % ważne: wymiary macierzy X i H muszą być takie same!
H = fft2(h,nx,ny); % wyznaczenie DFT2 filtra o rozmiarze obrazu
Xf = X .* H; % mnożenie transformat Fouriera obrazu i maski filtra
xf2 = uint8(real(iff2(Xf))); % transformacja powrotna do dziedziny obrazu
subplot(1,2,1), imshow(xf1), title('filtracja przez konwolucję')
subplot(1,2,2), imshow(xf2), title('filtracja przez mnoż. transformat')
figure
subplot(1,3,1), imagesc([-1 1],[-1 1],20*log10(abs(X))), colorbar,
colormap(cool)
title('|X(f_x,f_y)| obrazu (dB)'), xlabel('f_x'), ylabel('f_y')
subplot(1,3,2), imagesc([-1 1],[-1 1],20*log10(abs(H))), colorbar,
colormap(cool)
title('|H(f_x,f_y)| filtra (dB)'), xlabel('f_x'), ylabel('f_y')
subplot(1,3,3), imagesc([-1 1],[-1 1],20*log10(abs(Xf))), colorbar,
colormap(cool)
title('|X*H| po filtracji (dB)'), xlabel('f_x'), ylabel('f_y')
```

- Powtórzyć obliczenia dla filtra

```
h = fspecial('log',[5 5]) % maska filtra Laplacian of Gaussian
```

- Czy wyniki filtracji w dziedzinie czasu i częstotliwości są w obu przypadkach takie same?

3.5. Filtracja nieliniowa obrazu

- A. Przeprowadzić filtrację obrazu zakłóconego szumem Gaussa i szumem impulsowym ‘salt and pepper’ jak w pkt. 3.2.B za pomocą nieliniowego filtra medianowego.

```
x = imread('flower.jpg');
x = rgb2gray(x);
```



```

xg = imnoise(x,'gaussian',0,0.01); % szum Gaussa o średniej m=0 i var=0.01
xsp = imnoise(x,'salt & pepper',0.03); % 3% szumu impulsowego
N = [3 3]; % wymiary maski filtra
xf = medfilt2(x,N); % filtracja medianowa obrazu oryginalnego
xgf = medfilt2(xg,N); % filtracja obrazu z szumem Gaussa
xspf = medfilt2(xsp,N); % filtracja obrazu z szumem 'salt & pepper'
subplot(121), imagesc(x), colormap(gray), title('obraz oryginalny')
subplot(122), imagesc(xf), title('po filtracji medianowej')
figure
subplot(221), imshow(xg), title('obraz szumem Gaussa')
subplot(222), imshow(xgf), title('po filtracji medianowej')
subplot(223), imshow(xsp), title('obraz z szumem 'salt & pepper')
subplot(224), imshow(xspf), title('po filtracji medianowej')

```

- Powtórzyć filtrację z filtrem medianowym o większej masce, np. $N = [7 \ 7]$. Porównać jakość filtracji medianowej obu typów zakłóceń z jakością filtracji liniowej LP z pkt. 3.2 (filtr uśredniający, filtr Gaussa).

B. Przeprowadzić filtrację nieliniową *k-order* nadającą bieżącemu pikselowi *k*-tą kolejną wartość z ciągu uporządkowanych rosnąco wartości objętych domeną (maską) filtra stanowiącą tzw. *element strukturalny* SE. (Filtracja medianowa to przypadek szczególny: wybór środkowej wartości z ciągu uporządkowanych elementów domeny).

```

...
N=[3 3]; domain = ones(N); % określenie domeny - kwadratowego elementu SE
k = 2; % wybór k-tego elementu w ciągu rosnącym 3x3 wartości z domeny
xf = ordfilt2(x,k,domain); % filtracja obrazu oryginalnego
xspf = ordfilt2(xsp,k,domain); % filtracja obrazu z szumem 'salt & pepper'
xgf = ordfilt2(xg,k,domain); % filtracja obrazu z szumem Gaussa
subplot(121), imagesc(x), colormap(gray), title('obraz oryginalny')
subplot(122), imagesc(xf), title('po filtracji k-order, k=')
figure
subplot(221), imshow(xg), title('obraz szumem Gaussa')
subplot(222), imshow(xgf), title('po filtracji k-order, k=')
subplot(223), imshow(xsp), title('obraz z szumem 'salt & pepper')
subplot(224), imshow(xspf), title('po filtracji k-order, k=')

```

- Powtórzyć obliczenia dla $k=5$ i $k=8$, porównać i skomentować wyniki. Filtr *k-order* dla $k=1$ nazywany jest filtrem minimalnym, a dla $k=N(1) * N(2)$ – filtrem maksymalnym.

3.6. Usuwanie zakłóceń okresowych

W niniejszym punkcie zrealizowany zostanie prosty przykład projektowania filtra liniowego 2D o dowolnej charakterystyce częstotliwościowej wzmocnienia tzw. metodą próbkowania (widma pożądanego) w dziedzinie częstotliwości.

Zadanie polega na usunięciu z obrazu zakłóceń okresowych (pionowe pasy). Położenie zakłócenia na płaszczyźnie częstotliwości unormowanych (f_{nx}, f_{ny}) określa się (na oko) na podstawie widma DFT2 obrazu z zakłóceniem. Projektowany filtr, o pożądanej charakterystyce częstotliwościowej $H_0(f_{nx}, f_{ny})$, powinien mieć pasma zaporowe w obszarach występowania zakłóceń na płaszczyźnie częstotliwości, aby zakłócenia zostały wytłumione. Charakterystyka $H(f_{nx}, f_{ny})$, filtra o skończonej masce $h(m,n)$ jest aproksymacją SOI charakterystyki pożądanej, przy czym dokładność aproksymacji zależy od wymiarów maski filtra.

```

[x,map] = imread('josie.bmp');
x = ind2gray(x,map); % obraz oryginalny
N = size(x);
xz = double(x);
fn=0.25; % unormowana częstotliwość wzoru
for j=1:N(2),
    xz(:,j)=xz(:,j)+50*sin(pi*fn*j); % obraz zakłócony, wzór: pionowe pasy
end;
figure(1)

```

```

subplot(121), imshow(x), colormap(gray), title('obraz oryginalny')
subplot(122), imshow(uint8(xz)), title('obraz z zakłóceniem')

% Wyzn. widma obrazu z zakłóceniem dla ustalenia częstotl. zakłócenia
XZ = fftshift(fft2(xz)); % widmo obrazu z zakłóceniem
figure(2)
[fx,fy] = freqspace(size(x),'meshgrid');
subplot(121), mesh(fx,fy,20*log10(abs(XZ))), % skala magnitudy w dB
xlabel('f_x'), ylabel('f_y'),zlabel('DFT2 Magnitude (dB)')
colormap(hsv), colorbar, title('widmo obrazu z zakłóceniem')
axis([-1 1 -1 1 20 120]), view(-20,30),

% Generowanie pożądanej ch-ki częstotl. H0 filtra pasmowozaporowego
N = [31 31]; % rozmiar maski projektowanego filtra
H0 = ones(N); % początk. wzmocnienie =1 w całym zakresie
[fx,fy]=freqspace(size(H0),'meshgrid');
r=0.1; % promień koła zaporowego wokół częstotl. zakłócenia
bs = find((abs(abs(fx)-fn)<r & abs(fy)<r)); % domena pasm zaporowych BS
H0(bs)= 0; % nadanie wzmocnienia =0 w pożądanych obszarach zaporowych
subplot(122), mesh(fx,fy,H0),
xlabel('f_x'), ylabel('f_y'),zlabel('|H_0| Magnitude')
colormap(hsv), colorbar, title('zadana ch-ka wzmocnienia filtra')
axis([-1 1 -1 1 0 1.2]), view(-20,30),

% Projektowanie filtra o charakterystyce H (i masce h w dziedzinie obrazu)
% aproksymującej ch-kę zadaną H0 metodą próbkowania w dziedzinie częstotl.
h = real(fsamp2(H0)); % h ma ch-kę częstotl. H przechodzącą przez punkty H0
figure(3)
subplot(121), imagesc(h,[0,0.05*max(max(h))]), colormap(hsv),
colorbar, axis image, title('maska h(m,n) filtra')
subplot(122), freqz2(h), view(-20,30), axis([-1 1 -1 1 0 1.2])
colorbar, title('rzeczywista ch-ka wzmocnienia |H| filtra')

% Filtracja obrazu z zakłóceniem zaprojektowanym filtrem
xf = uint8(filter2(h,xz)); % filtracja obrazu
figure(4)
imshow(xf), colormap(gray), title('obraz po filtracji')

```

- Przeanalizować uzyskane wyniki. Zapoznać się z opisem **Help** funkcji projektowania maski filtra `fsamp2`. Powtórzyć obliczenia dla filtra o mniejszej masce, np. $N=[15 \ 15]$ i porównać wyniki. Jak zmieniła się jakość aproksymacji zadanej charakterystyki H_0 i skuteczność filtracji?

3.7. Badanie korelacji obrazu z wzorcem

Zadanie polega na określeniu na obrazie z zakłóceniami (normalny szum biały) położenia wzorca, którym jest prostokątny wycinek obrazu oryginalnego, przez wyznaczenie i analizę korelacji wzajemnej 2D obrazu zakłóconego i wzorca (funkcja `xcorr2`). Współrzędne maksimum korelacji wzajemnej odpowiadają współrzędnym lewego górnego rogu wzorca znajdującego się w położeniu, kiedy pokrywa się on z analogicznym fragmentem obrazu.

```

[x,map] = imread('obraz.bmp');
xw = imcrop(x,map); % zaznaczyć myszką prostokątny obszar większego koła i
                    % wyciąć za pomocą Crop Image (prawy klawisz myszy)
x = ind2gray(x,map);
xw = ind2gray(xw,map);
xz = imnoise(x,'gaussian',0,0.01); % szum Gaussa: średnia m, wariancja
[ma,na]=size(xz); [mb,nb]=size(xw); % wymiary obrazu i wzorca
rx=- (mb-1):(ma-1); ry=- (nb-1):(na-1); % skalowanie osi korelacji Rxy
Rxy=xcorr2(double(xz),double(xw))/(mb*nb);% korel. wzajemna obrazu i wzorca
figure(1)
subplot(121), imagesc(xz), colormap(gray), axis image,
title('obraz zakłócony')
subplot(122), imagesc(xw), colormap(gray), axis image,

```

```

title('wzór - wycinek')
figure(2)
subplot(121), imagesc(Rxy), colormap(cool), colorbar, axis image,
[rx,ry] = meshgrid(rx,ry);
subplot(122), mesh(rx',ry',Rxy), colormap(cool),
title('Korelacja wzajemna 2D'), view(80,30),
zlabel('R_{xy}(m,n)'), xlabel('m'), ylabel('n')

```

- Na wykresie 3D sprawdzić, czy współrzędne maksimum korelacji zgadzają się ze wzorem (4.11). Skomentować otrzymane wyniki.

3.8. Porównanie transformaty Fouriera i kosinusowej

Celem realizowanego przykładu jest porównanie efektów zastosowania transformacji Fouriera DFT2 i transformacji kosinusowej DCT2 do kodowania i rekonstrukcji obrazu. Wykonywane będą następujące operacje:

- transformacja obrazu za pomocą transformat DFT2 i DCT2,
- wyzerowanie części współczynników obu transformat reprezentujących wyższe częstotliwości przestrzenne (spełniających warunek $f/f_{Nyq} > r$ dla DFT2 lub $f/f_{Nyq} > 2r$ dla DCT2 ze względu na pełny zakres częstotliwości od 0 do $f_s = 2f_{Nyq}$, $0 < r < 1$), co odpowiada idealnej filtracji LP wokół częstotliwości zerowej i modeluje kompresję obrazu w dziedzinie transformat,
- odtworzenie obrazu na podstawie pozostałych niezerowych współczynników przez obliczenie transformat odwrotnych.

```

x = imread('flower.jpg');
x = rgb2gray(x);
r = 0.8; % promień koła zachowania współczynników
X = fftshift(fft2(double(x))); % wyznaczenie DFT2(x)
dft2(double(x),1); % wykresy magnitudy DFT2, skala dB
Xc = dct2(double(x)); % wyznaczenie DCT2(x)
dct2g(double(x),1); % wykresy magnitudy DCT2, skala dB

% Zerowanie współcz. poza kołem o promieniu r w DFT2
[fx,fy]=freqspace(size(X),'meshgrid'); % f=0 w centrum, -1<=fn<1
c=ones(size(X));
z = fx.^2 + fy.^2 - r^2;
c(find(z>0)) = 1e-6; % zerowanie współcz. poza kołem o promieniu r
Xr = X .* c;
figure
imagesc([-1 1],[-1 1],20*log10(abs(Xr))), colorbar, colormap(jet)
xlabel('f_x'), ylabel('f_y'), title('DFT2 Magnitude (dB)')

% Zerowanie współcz. poza kołem o promieniu 2r w DCT2
[M,N]=size(Xc);
fx=(0:2:2*M-1)/M; fy=(0:2:2*N-1)/N; % grid MxN częstotl. w zakresie 0<=fn<2
[fx,fy]=meshgrid(fx,fy); fx=fx'; fy=fy'; % f=0 w lewym górnym rogu
c=ones(size(Xc));
z = fx.^2 + fy.^2 - (2*r)^2; % 2*r dla DCT ze względu na zakres 0<=fn<2
c(find(z>0)) = 1e-6;
Xcr = Xc .* c;
figure
imagesc([0 2],[0 2],20*log10(abs(Xcr))), colorbar, colormap(jet)
xlabel('f_x'), ylabel('f_y'), title('DCT2 Magnitude (dB)')

% Odtworzenie obrazu przez transformacje odwrotne IDFT2 i IDCT2
xr = uint8(real(ifft2(fftshift(Xr)))); % odtworzenie IFFT2
xcr = uint8(real(idct2(Xcr))); % odtworzenie IDCT2
figure
subplot(131), imshow(x), colormap(gray), title('obraz oryginalny')
subplot(132), imshow(xr), colormap(gray), title('odtworzenie IDFT2')
subplot(133), imshow(xcr), colormap(gray), title('odtworzenie IDCT2')

```

- **Uwaga:** Do prezentowania wykresów 2D/3D magnitudy widma DFT2 wykorzystywane jest funkcja `dft2` (przedstawiona w pkt. 3.1), a do prezentowania wykresów widma DCT2 podobna funkcja `dct2g`.

Sprawdzić, czy w folderze roboczym znajduje się plik funkcji `dct2g.m`. Jeżeli nie, poniższy listing należy skopiować do edytora Matlab'a i zapisać funkcję w folderze roboczym w pliku `dct2g.m`.

```
function Xc = dct2g(x,opt)
% 2-wymiarowe widmo DCT obrazu i wykres 2D/3D w funkcji 0<=fn<=2, fn=f/fNyq
% Parametry wejściowe: x - obraz (MxN pikseli),opt (opcja)-> skala log (dB)
% Wyjście: Xc = DCT2g[x], dct2g(x) -> rys. wykresy

Xc = dct2(x);

if nargin==0,
    [M,N]=size(Xc);
    f1=(0:2:2*M-1)/M; f2=(0:2:2*N-1)/N; % grid częstotliwości w zakresie
0<=fn<2
    [f1,f2]=meshgrid(f1,f2); f1=f1'; f2=f2';
    figure % Widmo amplitudowe DCT2 3D obrazu
    if nargin<2,
        mesh(f1,f2,Xc),
        colormap(jet), colorbar;
        zlabel('|X_c(f_{nx}, f_{ny})|');
        title('DCT2 Magnitude, f_n= f / f_{Nyq}')
    else
        mesh(f1,f2,real(20*log10(Xc))),
        colormap(jet), colorbar;
        zlabel('20 log |X_c(f_{nx}, f_{ny})| (dB)');
        title('DCT2 Magnitude (dB), f_n= f / f_{Nyq}')
    end
    colormap(jet), colorbar, view(140,45); % azimuth, elevation (stopnie)
    xlabel('f_{nx}'), ylabel('f_{ny}')
    figure
    if nargin<2 % Mapa barwna 2D widma amplitud. DCT2
        imagesc([0 1],[0 1],Xc),
        title('DCT2 Magnitude, f_n= f / f_{Nyq}')
    else
        imagesc([0 1],[0 1],real(20*log10(Xc))),
        title('DCT2 Magnitude (dB), f_n= f / f_{Nyq}')
    end
    colorbar, colormap(jet), xlabel('f_{nx}'), ylabel('f_{ny}')
```

- Przeprowadzić obliczenia (transformacje i odtworzenia) dla $r=0.8$, $r=0.4$, $r=0.1$. Porównać i skomentować rezultaty.

4. Zadania do realizacji samodzielnej

Zadanie 4.1. Napisać własną funkcję dwuwymiarowej filtracji obrazu zapisanego w macierzy x przez liniowy filtr dyskretny o prostokątnej masce o wymiarach $M \times N$ zapisanej w macierzy h . Obraz po filtracji zapisać w macierzy y . Ograniczyć się do przypadku, kiedy maska nie wystaje poza obraz, czyli macierz y ma wymiary mniejsze niż x , co odpowiada funkcji Matlab'a `filter2` z opcją `'valid'`.

5. Opracowanie sprawozdania

W sprawozdaniu należy zawrzeć zarejestrowane wyniki eksperymentów obliczeniowych przeprowadzonych według instrukcji oraz zadanych przez prowadzącego zadań do realizacji

samodzielnej z odpowiednimi opisami oraz wyjaśnieniami opartymi na wiedzy z wykładu i literatury, ze szczególnym uwzględnieniem problemów wskazanych symbolem ➤.

Literatura

1. Wróbel Z., Koprowski R.: *Praktyka przetwarzania obrazów z zadaniami w programie Matlab*, Akadem. Oficyna Wyd. EXIT, 2012.
2. Malina W., Smiatacz M.: *Cyfrowe przetwarzanie obrazów*, Akadem. Oficyna Wyd. EXIT, 2008.
3. Solomon Ch., Breckon T.: *Fundamentals of Digital Image Processing. Practical Approach with Examples in Matlab*, John Wiley & Sons, 2011.
4. Mitra S.K.: *Digital Signal Processing. A Computer Based Approach*, 4 ed., McGraw-Hill, 2011.
5. Leis J.W.: *Digital Signal Processing Using Matlab for Students and Researchers*, John Wiley & Sons, 2011.
6. Zieliński T.: *Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań*, WKiŁ, 2005.
7. Smith S.W.: *Cyfrowe przetwarzanie sygnałów. Praktyczny poradnik dla inżynierów i naukowców*, Wyd. BTC, 2007.

Opracował: Dr inż. Janusz Baran

Częstochowa, 1999-2013