

# Przetwarzanie obrazów 4

- Handout

SL 2024/2025

Wydział Matematyki, Fizyki i Informatyki



Dr.-Ing. A. Rowińska-Schwarzsteller  
[agnieszka.rowinska-schwarzsteller@ug.edu.pl](mailto:agnieszka.rowinska-schwarzsteller@ug.edu.pl)

18. März 2025

# Spis treści

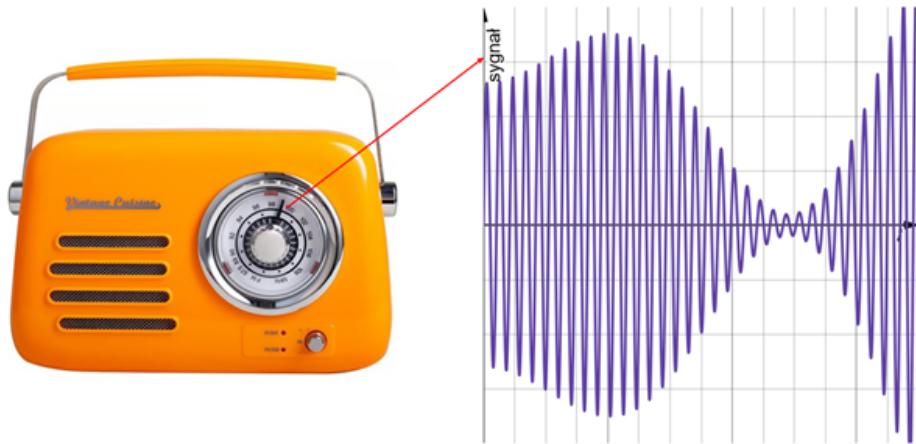
<b>8 Filtrowanie obrazów: filtry liniowe</b>	<b>2</b>
8.1 Wprowadzenie . . . . .	5
8.1.1 Charakterystyka systemu przetwarzania sygnałów cyfrowych (1– $D$ ) . . . . .	5
8.1.2 System liniowy, niezmienniczy w czasie (LTI, <i>linear time-invariant system</i> ) . . . . .	6
8.1.3 Splot a korelacja . . . . .	14
8.2 Operacje lokalne na obrazach . . . . .	16
8.2.1 Punkty położone na krawędzi obrazu . . . . .	16
8.2.2 Sąsiedztwo . . . . .	16
8.3 Formalne własności filtrów liniowych . . . . .	17
8.3.1 Splot . . . . .	19
8.3.2 Własności splotu . . . . .	24
8.3.3 Filtracja w dziedzinie widma (spektrum Fouriera) . . . . .	26
8.4 Charakterystyki filtrów liniowych . . . . .	27
8.4.1 Filtry dolnoprzepustowe . . . . .	27
8.4.2 Filtry górnoprzepustowe . . . . .	44
8.5 Dekonwolucja . . . . .	70
8.5.1 Odpowiedź punktowa systemu ( <i>point spread function, PSF</i> ) . . .	72
8.5.2 Algorytmy dekonwolucji . . . . .	75

# 8

## Filtrowanie obrazów: filtry liniowe

Filtrowanie sygnałów

→ grafika 8.1

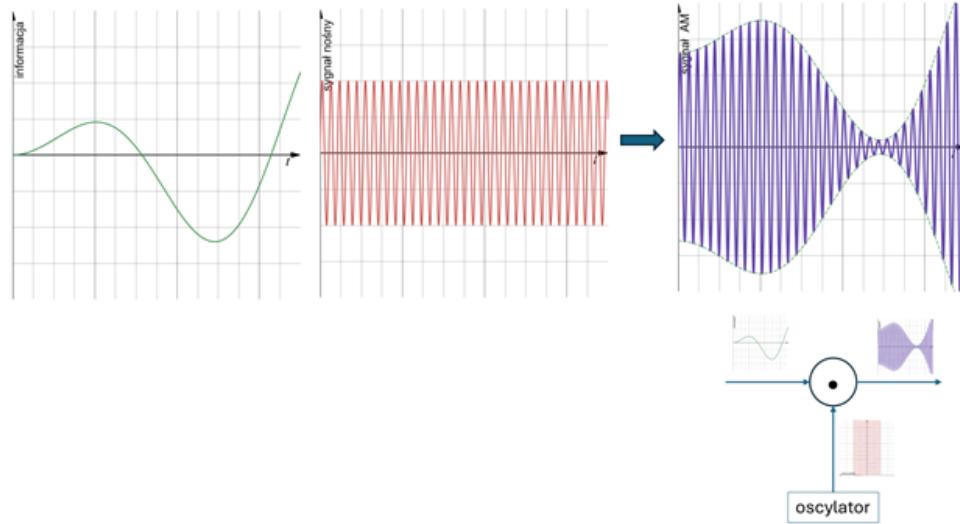


Grafika 8.1: Przykład: filtrowanie sygnałów (filtr średniodzakresowy)

**Przykład: sygnał AM (modulacja amplitudy, *Amplitude modulation*)**

- radio, telewizja analogowa (kolorowa), łączność w kontroli ruchu lotniczego, systemy radiowej nawigacji morskiej i lotniczej

→ grafika 8.2



Grafika 8.2: Modulacja amplitudy sygnału

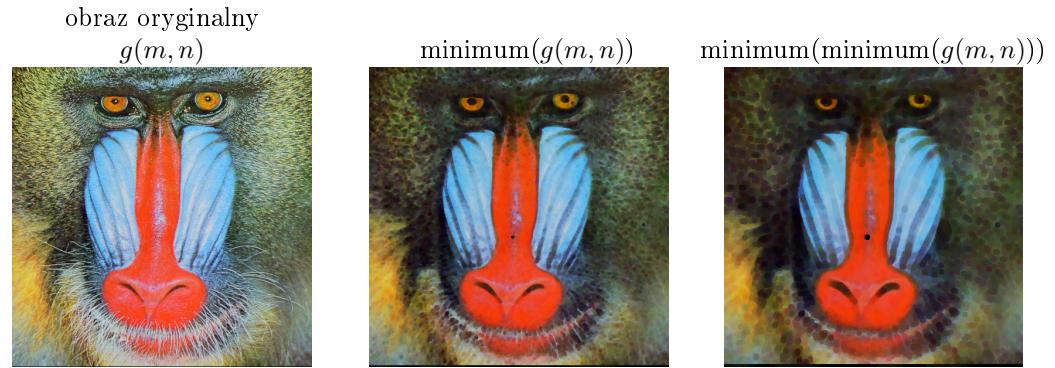
### Filtrowanie obrazów

Celem filtrowania jest poprawa obrazu, w szczególności redukcja obszarów nieistotnych / podkreślenie obszarów zawierających informację.

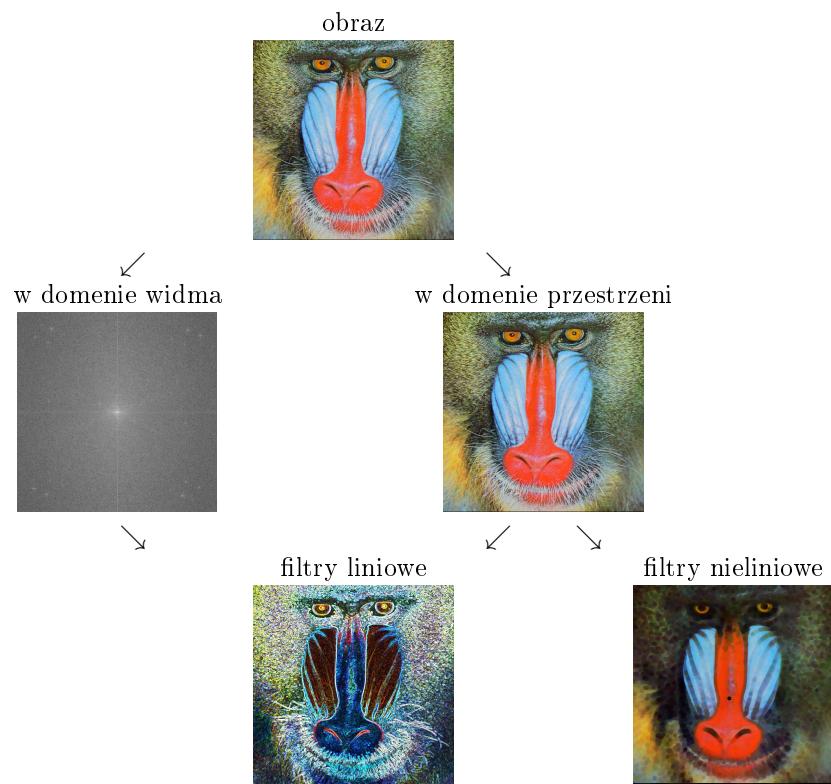
- Operacje filtrujące modyfikują pojedyncze obrazy.
- Aby obliczyć wartość jasności (szarości) w obrazie wyjściowym, operacje filtrowania obejmują sąsiedztwo piksela obrazu wyjściowego.
- Dla obrazu reprezentowanego w modelu RGB filtrację przeprowadza się osobno dla każdej składowej obrazu (oddzielne przekształcenia dla składowej R, G oraz B).
- Rozróżnia się filtry liniowe i nieliniowe.
- Przykład filtra nieliniowego: "filtr minimalny" (filtr kompresujący, filtr erozyjny)  
→ zastąpienie wartości piksela najciemniejszym pikselem w sąsiedztwie, grafika 8.3.

### Metody filtrowania obrazu

→ grafika 8.4



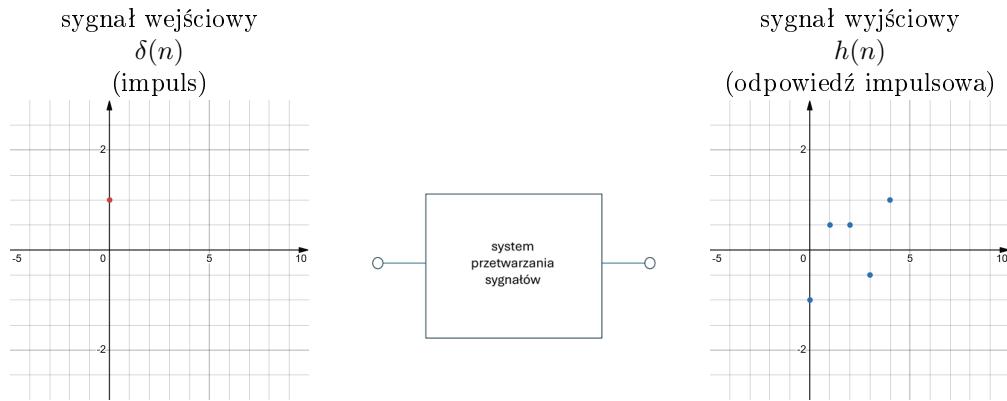
Grafika 8.3: Zastosowanie filtra minimalnego



Grafika 8.4: Metody filtrowania obrazu

## 8.1 Wprowadzenie

### 8.1.1 Charakterystyka systemu przetwarzania sygnałów cyfrowych ( $1 - D$ )

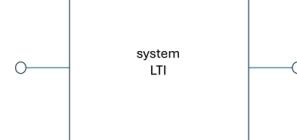
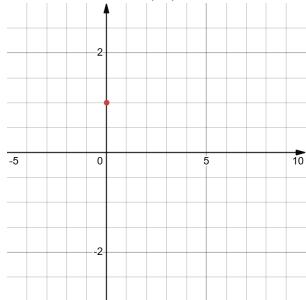


Jaka jest odpowiedź systemu na dowolny sygnał wejściowy  $g$ ?

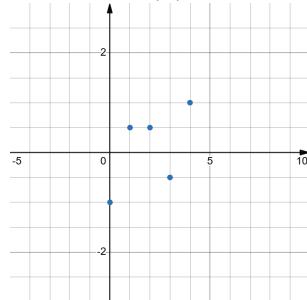


### 8.1.2 System liniowy, niezmienniczy w czasie (LTI, *linear time-invariant system*)

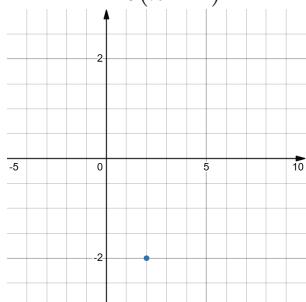
sygnał wejściowy  
 $\delta(n)$



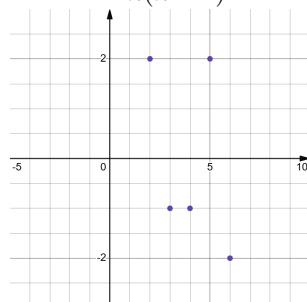
sygnał wyjściowy  
 $h(n)$



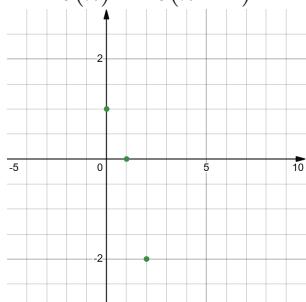
$\downarrow$   
 $-2\delta(n - 2)$



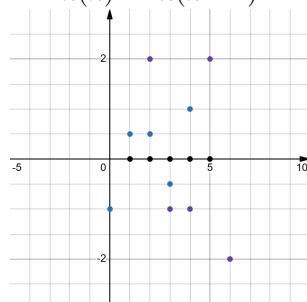
$\downarrow$   
 $-2h(n - 2)$



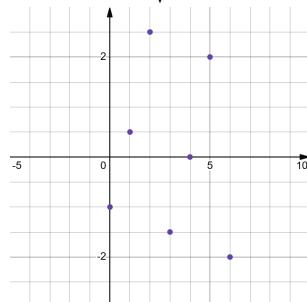
$\delta(n) - 2\delta(n - 2)$



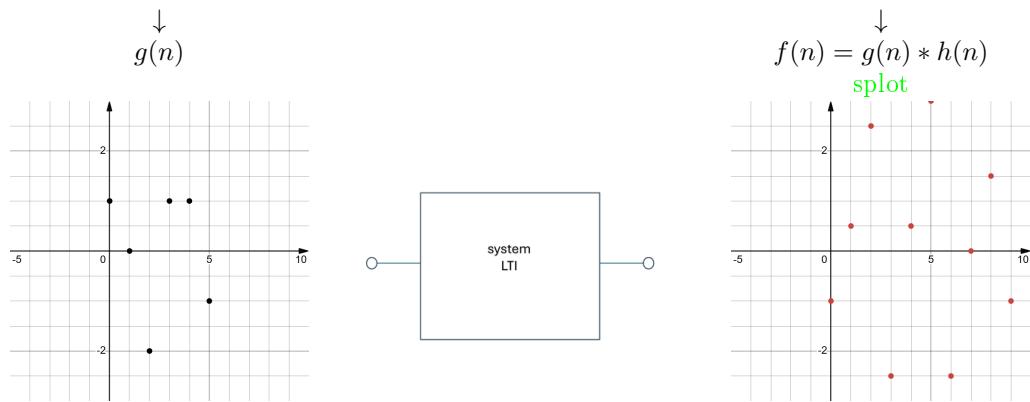
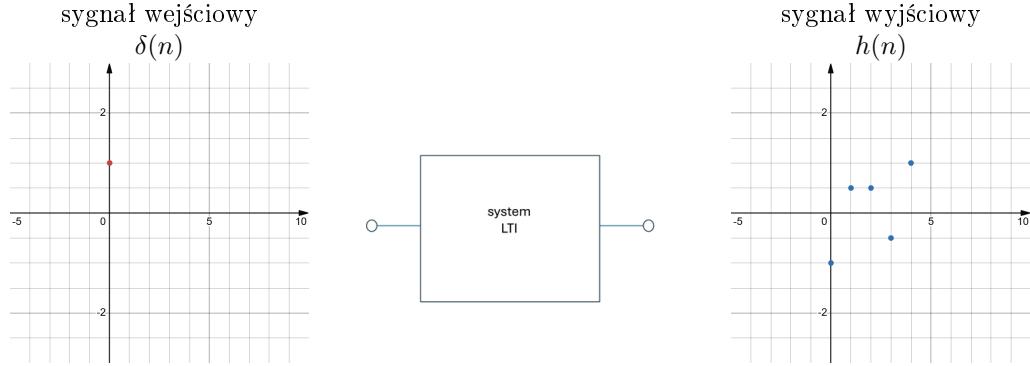
$\downarrow$   
 $h(n) - 2h(n - 2)$



$\downarrow$



Dowolny sygnał wejściowy  $g$ :

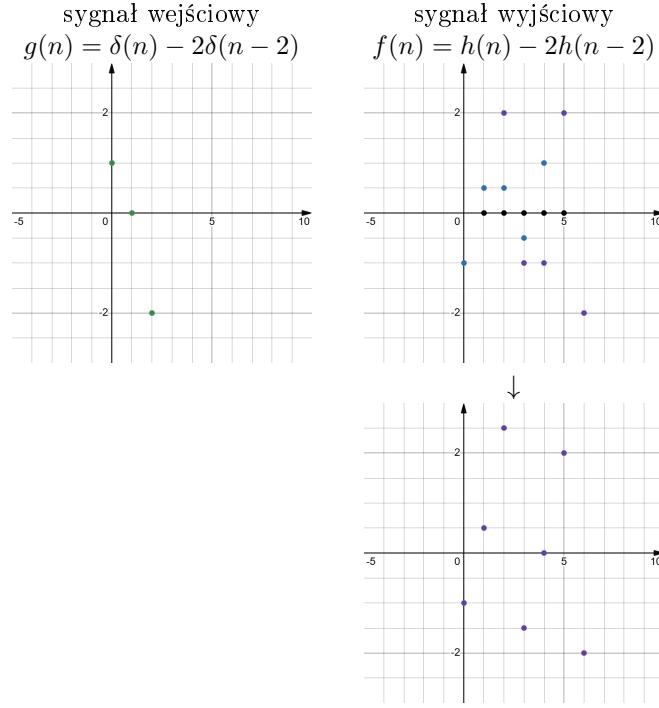


FILTROWANIE SYGNAŁU  
 $g$  FILTREM  $h$

Filtры линейные не являются изобретением цифровой обработки изображений, но они берут свое начало из математики. Выполняется путем фильтрации операция называется линейной сверткой (англ. *convolution*) и соединяет две функции одинаковой размерности, непрерывные или дискретные.

### Wyznaczanie splotu

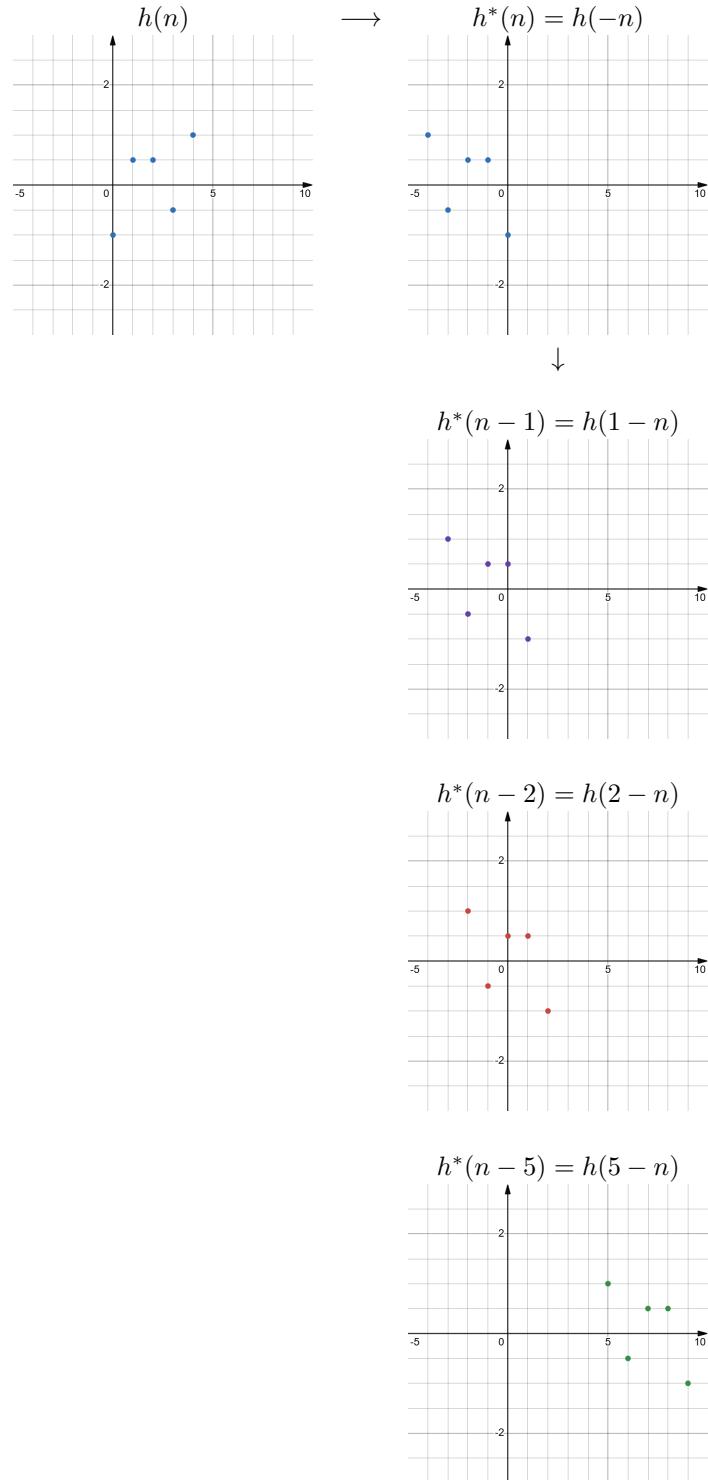
Przykład:



$$\begin{aligned}
 f(0) &= g(0) \cdot h(0) &= \sum_{i=0}^2 g(i) \cdot h(0-i) \\
 f(1) &= g(1) \cdot h(0) + g(0) \cdot h(1) &= \sum_{i=0}^2 g(i) \cdot h(1-i) \\
 f(2) &= g(2) \cdot h(0) + g(1) \cdot h(1) + g(0) \cdot h(2) &= \sum_{i=0}^2 g(i) \cdot h(2-i) \\
 f(3) &= 0 \cdot h(0) + g(2) \cdot h(1) + g(1) \cdot h(2) + g(0) \cdot h(3) &= \sum_{i=0}^2 g(i) \cdot h(3-i) \\
 &\vdots
 \end{aligned}$$

Niech

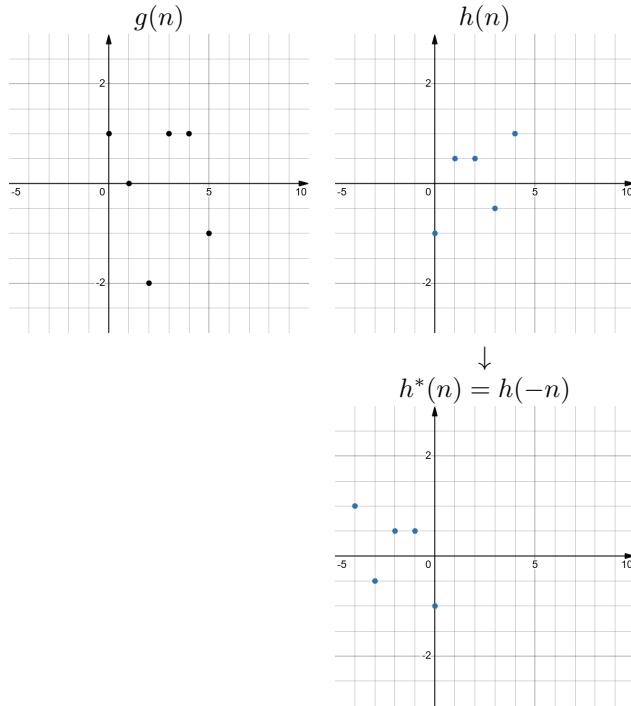
$$h^*(n) = h(-n)$$



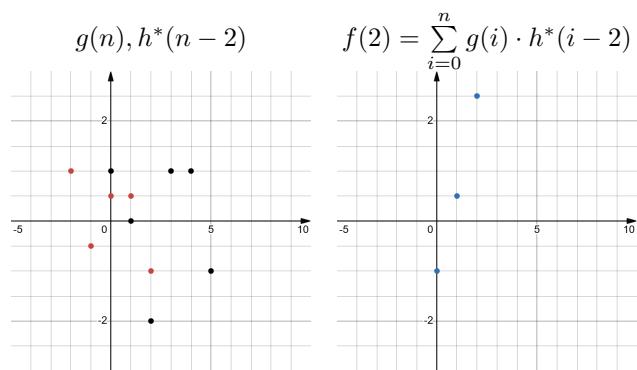
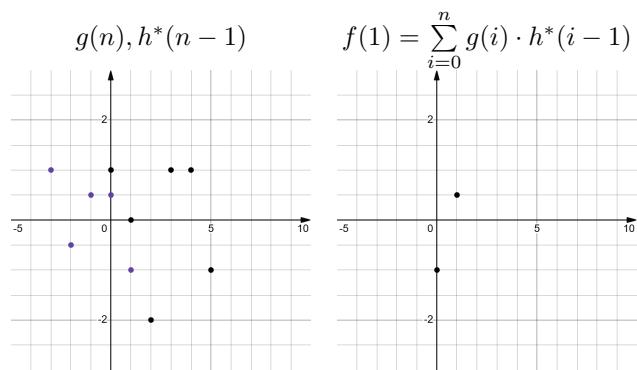
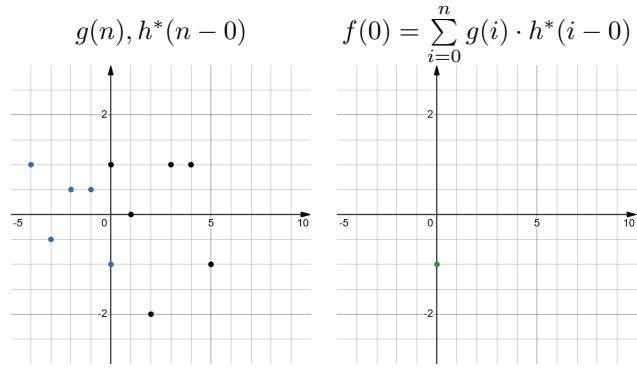
$$\begin{aligned}
 f(0) &= \sum_{i=0}^2 g(i) \cdot h(0-i) = \sum_{i=0}^2 g(i) \cdot h^*(i-0) \\
 f(1) &= \sum_{i=0}^2 g(i) \cdot h(1-i) = \sum_{i=0}^2 g(i) \cdot h^*(i-1) \\
 f(2) &= \sum_{i=0}^2 g(i) \cdot h(2-i) = \sum_{i=0}^2 g(i) \cdot h^*(i-2) \\
 f(3) &= \sum_{i=0}^2 g(i) \cdot h(3-i) = \sum_{i=0}^2 g(i) \cdot h^*(i-3) \\
 &\vdots
 \end{aligned}$$

**Splot  $f = g * h$ :**

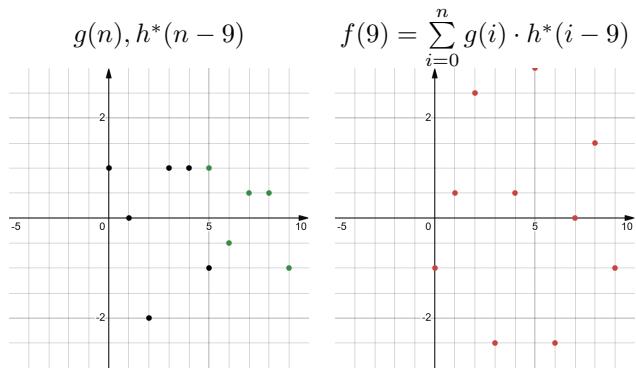
- wyznaczenie  $h^*(n) = h(-n)$



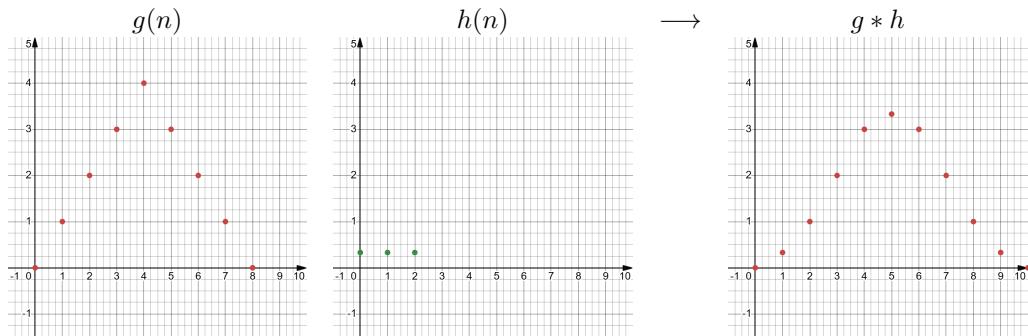
- ważenie sygnału  $g$  sygnałem  $h^*$  (korelacja  $g$  z  $h^*$ )



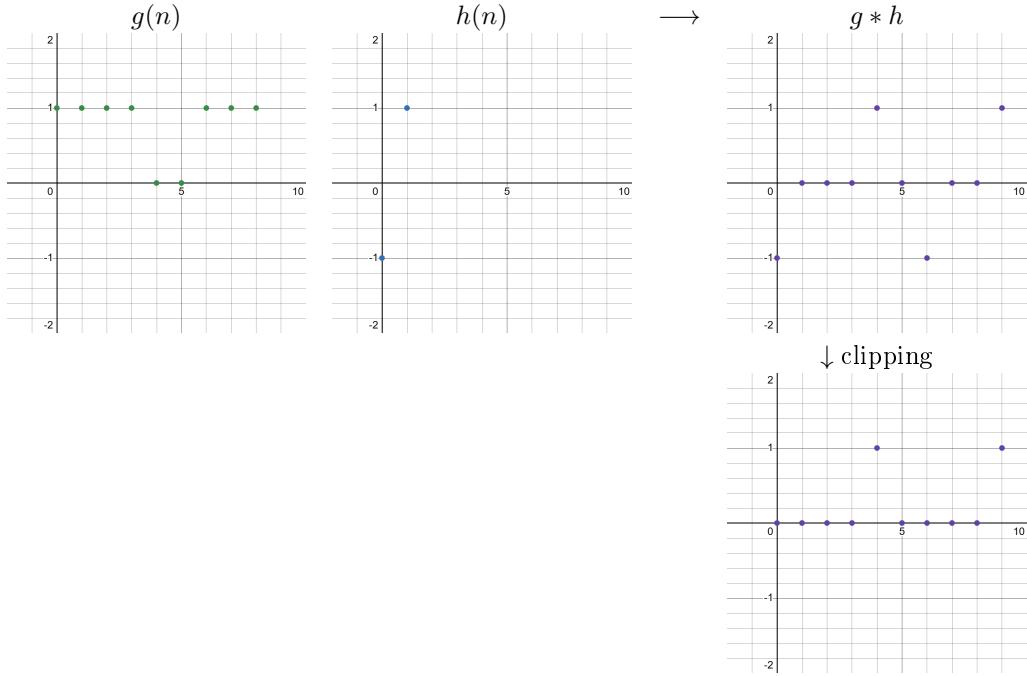
⋮

**Przykłady 1 – D**

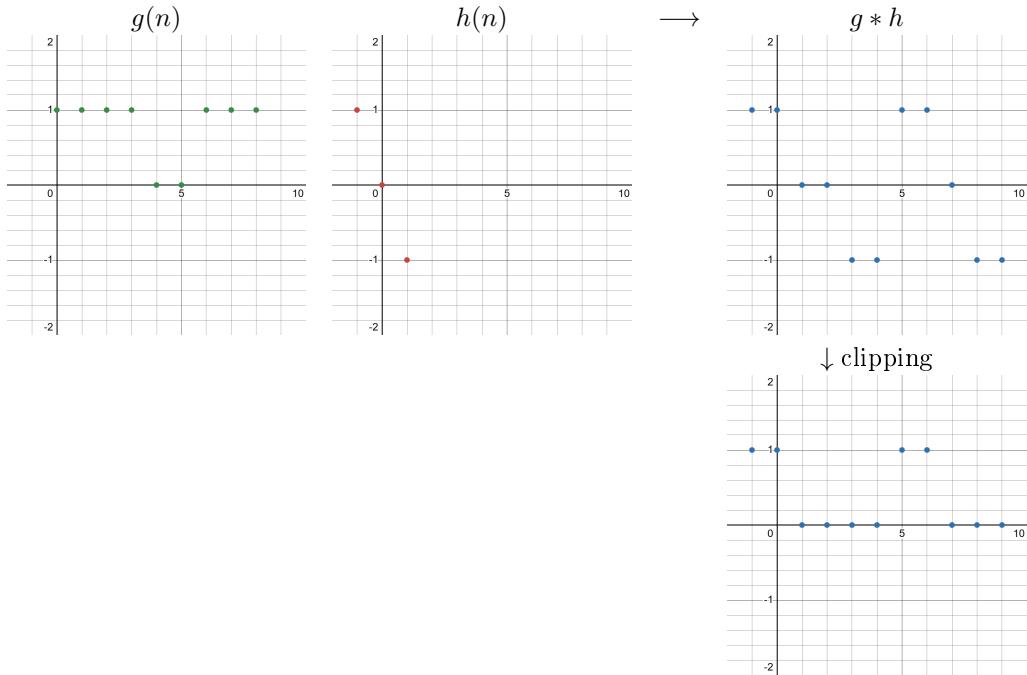
Przykład 1:



Przykład 2:



Przykład 3:



### 8.1.3 Splot a korelacja

- **Splot**

- operacja matematyczna polegająca na łączeniu dwóch sygnałów lub funkcji w celu wygenerowania nowego sygnału lub funkcji,
- operacja przemieniona, łączna i rozdzielna względem dodawania,
- operacja splotu w dziedzinie czasu/przestrzeni jest równoważna mnożeniu transformat Fouriera (tzn. operacją w dziedzinie częstotliwości) lub innych transformat wykorzystywanych w przetwarzaniu sygnałów.

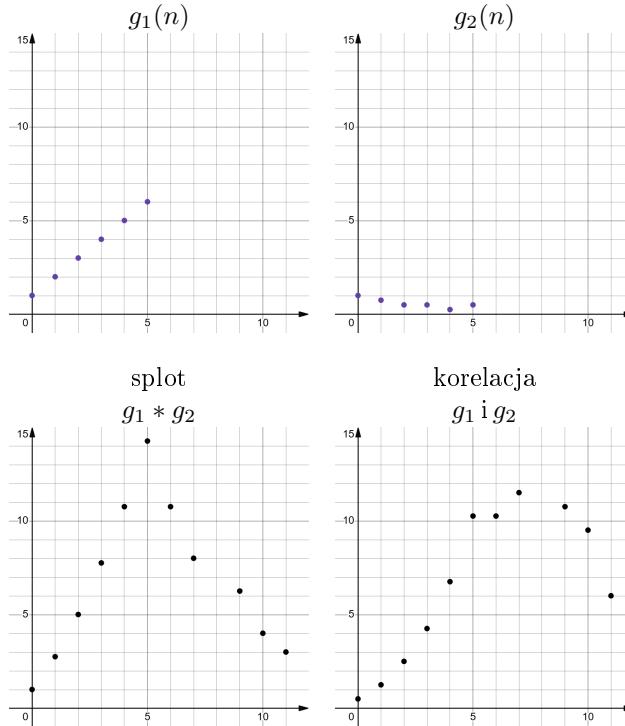
Splot jest powszechnie stosowany w przetwarzaniu sygnałów (cyfrowych, dyskretnych, ciągłych), cyfrowym przetwarzaniu obrazu, rozpoznawaniu obrazów, nauce o danych, transformacji sygnałów i wielu innych zastosowaniach. W pierwotnym znaczeniu filtrowanie sygnału to operacja na widmie i odpowiadający jej splot sygnałów w dziedzinie czasu (lub jak dla obrazów cyfrowych w przestrzeni obrazu).

- **Korelacja**

- operacja matematyczna, która mierzy lub kwantyfikuje podobieństwa i zależności między dwoma sygnałami lub funkcjami

Korelację stosuje się głównie do pomiaru podobieństwa między dwoma sygnałami, porównywania sygnałów, dopasowywania szablonów, rozpoznawania wzorców i zależności, analizy danych itp.

Przykład: operacje dla sygnałów  $g_1$  i  $g_2$ :



### Przykład: splot

$h(i)$ : jądro filtra

$h^*(i)$ : współczynniki wektora wag

$$h^*(-i) := h(i)$$

$$h^* \begin{array}{|c|c|c|c|c|} \hline h^*(-2) & h^*(-1) & h^*(0) & h^*(1) & h^*(2) \\ \hline 0 & 1 & 2 & 3 & 0 \\ \hline \end{array} \leftarrow h \begin{array}{|c|c|c|c|c|} \hline h(-2) & h(-1) & h(0) & h(1) & h(2) \\ \hline 0 & 3 & 2 & 1 & 0 \\ \hline \end{array}$$

$\underbrace{g * h}_{\text{splot}} := \text{kombinacja liniowa } g \text{ z wagami } h^*$

$$1. \quad g_1 = (g * h) * h$$

$$\begin{array}{ll} g(m) & \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \\ h & \begin{array}{|c|c|c|c|c|} \hline 0 & 3 & 2 & 1 & 0 \\ \hline \end{array} \\ \downarrow & \\ h^* & \begin{array}{|c|c|c|c|c|} \hline 0 & 1 & 2 & 3 & 0 \\ \hline \end{array} \\ g * h = g'(m) & \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 3 & 2 & 1 & 0 & 0 & 0 & 0 \\ \hline \end{array} \end{array}$$

$$(g * h) * h = g_1(m) \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 9 & 12 & 10 & 4 & 1 & 0 & 0 & 0 \\ \hline \end{array}$$

$$2. \quad g_2 = g * (h * h)$$

$$\begin{array}{ll} g(m) & \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \\ h & \begin{array}{|c|c|c|c|c|c|} \hline 0 & 3 & 2 & 1 & 0 \\ \hline \end{array} \\ \downarrow & \\ h^* & \begin{array}{|c|c|c|c|c|} \hline 0 & 1 & 2 & 3 & 0 \\ \hline \end{array} \\ h * h = h_1(m) & \begin{array}{|c|c|c|c|c|} \hline 9 & 12 & 10 & 4 & 1 \\ \hline \end{array} \\ \downarrow & \\ h_1^* & \begin{array}{|c|c|c|c|c|} \hline 1 & 4 & 10 & 12 & 9 \\ \hline \end{array} \end{array}$$

$$g * (h * h) = g_2(m) \quad \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 9 & 12 & 10 & 4 & 1 & 0 & 0 & 0 \\ \hline \end{array}$$

→  $g_1 = g_2$

→ splot (*convolution*) to operacja łączna

## 8.2 Operacje lokalne na obrazach

Zasadniczą właściwością operacji punktowych jest to, że nowa wartość piksela obrazu zależy wyłącznie od oryginalnej wartości piksela obrazu w tym samym miejscu. Operacje lokalne są podobne do operacji punktowych w tym sensie, że istnieje również mapowanie 1:1 współrzędnych obrazu, tzn. geometria obrazu nie ulega zmianie. Operacje lokalne nazywane są także operacjami filtrującymi (w skrócie filtrami). Filtry różnią się od operacji punktowych przede wszystkim tym, że wynik nie jest obliczany na podstawie pojedynczego oryginalnego piksela, ale na podstawie kilku pikseli oryginalnego obrazu. Współrzędne pikseli źródłowych są definiowane w odniesieniu do aktualnej pozycji  $(u, v)$  i tworzą ciągły obszar. Rozmiar obszaru filtra określa, ile oryginalnych pikseli bierze udział w obliczeniu nowej wartości piksela (zasięg przestrzenny filtra). Typowy rozmiar filtrów to  $3 \times 3$ , wyśrodkowany w sąsiedztwie ośmiospójnym na bieżącą współrzędną  $(u, v)$ . Kształt obszaru filtra nie musi być jednak kwadratowy, ale może przyjąć dowolny kształt.

### 8.2.1 Punkty położone na krawędzi obrazu

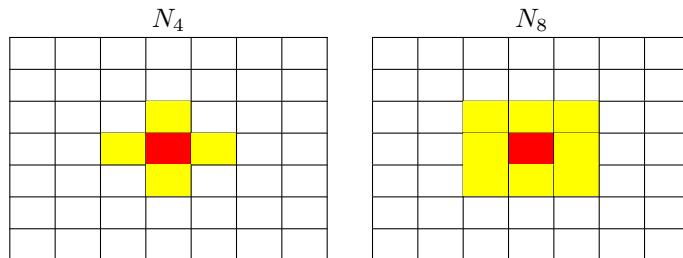
Metody:

- kontynuacja obrazu (dodanie do filtrowanego obrazu zduplikowanych pikseli znajdujących się na jego brzegu),
- pominięcie procesu filtracji dla punktów brzegowych,
- zmniejszenie obrazu po filtracji o punkty, dla których proces ten nie mógł być wykonany,
- przyjęcie założenia, że wszystkie wartości poza znanim sygnałem obrazu to wyłącznie wartości 0.

### 8.2.2 Sąsiedztwo

Sąsiedztwo to mały, określony obszar obrazu wokół piksela.

Operacje przetwarzania obrazu wykorzystują piksele w sąsiedztwie w celu obliczenia nowej wartości koloru lub szarości dla ich środka.



W przetwarzaniu obrazów najczęściej wykorzystywane sąsiedztwa punktu  $P(u, v)$  to:

- czterospójne (von Neumanna)

$$N_4(P) = \{(u, v), (u, v - 1), (u, v + 1), (u - 1, v), (u + 1, v)\}$$

Sąsiedztwo czterospójne składa się z pięciu pikseli.

- ośmiospójne (Moore'a)

$$N_8(P) = \{(u, v), (u, v - 1), (u, v + 1), (u - 1, v), (u + 1, v), (u - 1, v - 1), \\ (u - 1, v + 1), (u + 1, v - 1), (u + 1, v + 1)\}$$

Sąsiedztwo ośmiospójne składa się z dziewięciu pikseli.

Odległość najdalszych sąsiadów  $N$  od  $P$  jest określona poprzez zastosowaną metrykę. Na przykład w przypadku użycia metryki euklidesowej, odległość w sąsiedztwie ośmiospójnym wynosi  $\sqrt{2}$ , podczas gdy w przypadku użycia metryki Manhattan wynosi ona 2.

#### Sąsiedztwa zdefiniowane przez promień $r$ (ImageJ)

→ grafika 8.5

### 8.3 Formalne własności filtrów liniowych

Filtry liniowe łączą wartości pikseli w obszarze filtra w formie liniowej - poprzez sumowanie ważone. → grafika 8.6

Prostym przykładem jest uśrednianie lokalne, w którym wszystkie dziewięć pikseli obszaru filtra  $3 \times 3$  jest sumowanych z wagą  $\frac{1}{9}$ :

$$g'(m, n) = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 g(m+i, n+j)$$

Za pomocą tego samego mechanizmu można - zmieniając poszczególne wagę - zdefiniować dużą liczbę filtrów o różnych charakterystykach.

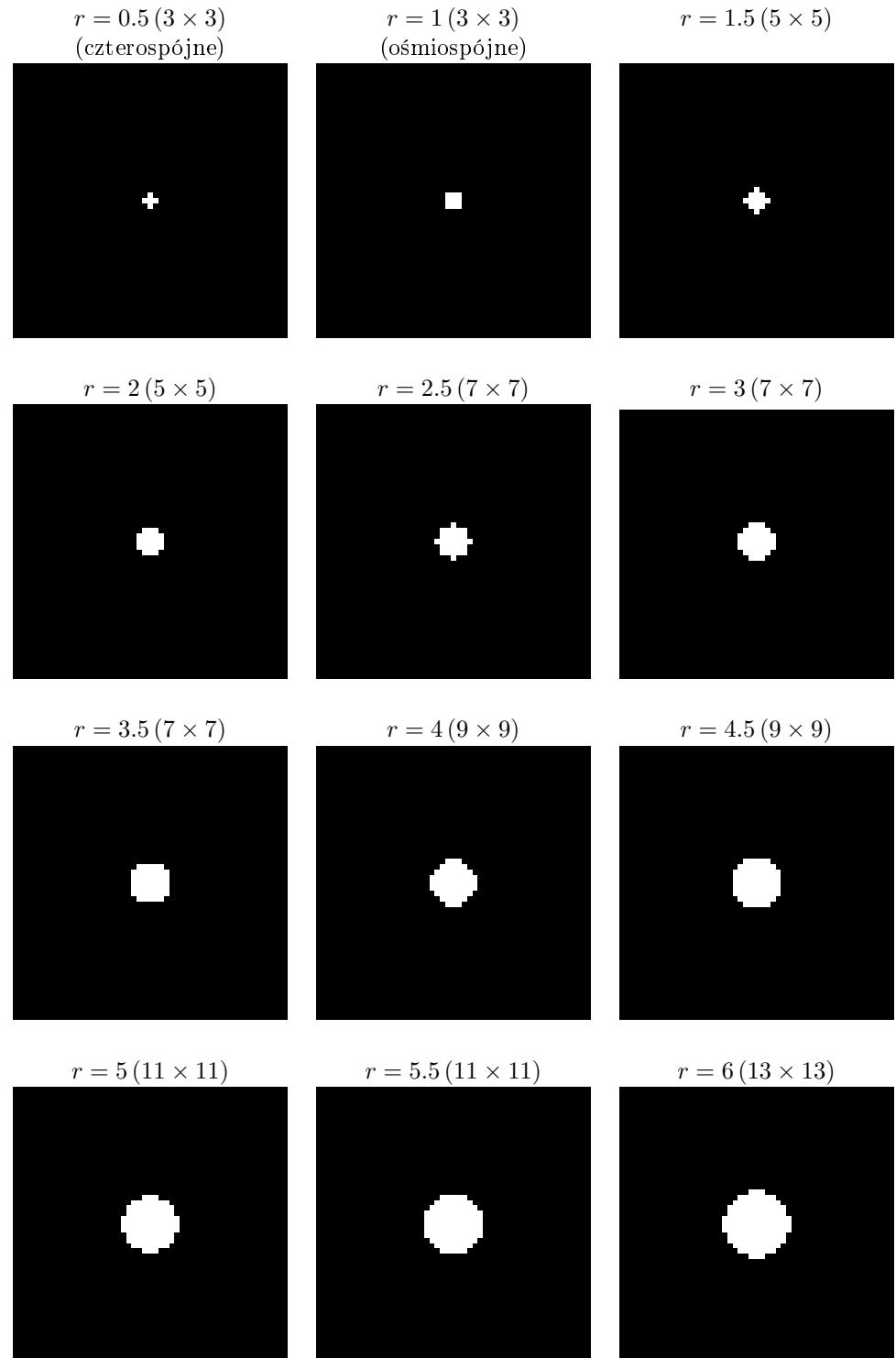
W przypadku filtrów liniowych wielkość i kształt obszaru filtra, oraz związane z nimi wag są określone przez macierz współczynników filtra, tzw. „jądro filtra”(ang. *kernel*), „macierz filtra” lub „maskę filtra”  $h(m, m)$ . Z jądra filtra wyznaczane są współczynniki wag filtra  $h^*(m, n)$ .

Zastosowanie współczynników wag filtra do obrazu jest – zgodnie z grafiką 8.6 – prostym procesem: przy każdej pozycji obrazu  $g(m, n)$  wykonywane są następujące kroki:

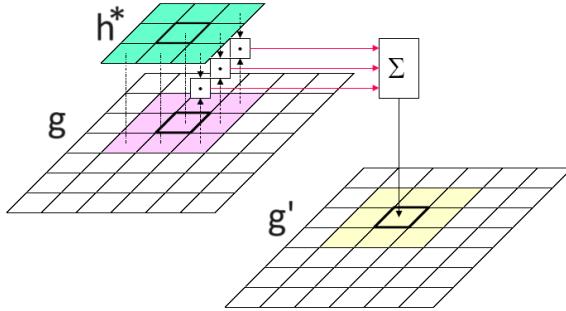
- Matryca współczynników wag filtra  $h^*$  jest umieszczona nad oryginalnym obrazem w taki sposób, że jej początek współrzędnych  $h^*(0, 0)$  przypada na aktualny element obrazu  $g(m, n)$ .
- Następnie wszystkie elementy obrazu są mnożone przez znajdujący się nad nimi współczynnik wag filtra, a wyniki są sumowane.
- Wynikowa suma jest zapisywana w odpowiedniej pozycji obrazu wyjściowego  $g'(m, n)$ .

Zatem wszystkie piksele dla opisanego wcześniej filtra  $3 \times 3$  nowego obrazu  $g'(m, n)$  obliczane są w następujący sposób:

$$g'(m, n) = \sum_{i=-1}^1 \sum_{j=-1}^1 g(m+i, n+j) \cdot h^*(i, j)$$



Grafika 8.5: Sąsiedztwa zdefiniowane przez promień  $r$



Grafika 8.6: Suma ważona pikseli w sąsiedztwie

### 8.3.1 Splot

W przypadku dyskretnych, dwuwymiarowych funkcji  $g$  i  $h$  operację splotu  $g(m, n) * h(m, n)$  definiuje się jako:

$$\begin{aligned} g'(m, n) = g(m, n) * h(m, n) &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(m-i, n-j) \cdot h(i, j) \\ &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(i, j) \cdot h(m-i, n-j) \\ &= h(m, n) * g(m, n) \end{aligned}$$

Ponieważ przyjmuje się, że współczynniki filtra poza macierzą filtra  $h(m, n)$  wynoszą zero, pozycje poza macierzą nie są istotne dla sumowania.

Przekształcenie

$$\begin{aligned} g'(m, n) = g(m, n) * h(m, n) &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(m-i, n-j) \cdot h(i, j) \\ &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(m+i, n+j) \cdot \underbrace{h(-i, -j)}_{h^*(i, j)} \\ &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(m+i, n+j) \cdot h^*(i, j) \end{aligned}$$

pozwala na wyznaczenie współczynników wag filtra  $h^*(m, n)$  z jądra filtra  $h(m, n)$  poprzez odbicie w poziomie i w pionie (lub obrócenie o  $180^\circ$ ).

Operacją matematyczną stojącą za wszystkimi filtrami liniowymi jest splot liniowy (\*), a wynik jest całkowicie i wyłącznie zdefiniowany przez jądro splotu (macierz filtra)  $h(m, n)$ . Dlatego operacja splotu nazywana jest też operacją „Black Box”.

**Przykład: splot w 2D**

Jądro splotu  $h(m, n)$ :

$$h(m, n) = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

obraz wejściowy:

		$m$		
		0	1	2
n	0	1	2	3
	1	4	5	6
	2	7	8	9

$g(m, n)$

		$m$		
		-1	0	1
n	-1	-1	-2	-1
	0	$h(0, 0)$	0	0
	1	1	2	1

$h(m, n)$

- $h(0, 0)$  jest środkiem jądra splotu.
- założenie:  $g(m, n) = 0$  dla  $m, n \notin \{0, 1, 2\}$

**Metody obliczenia splotu:**

1. Obliczenie splotu z definicji

$$\begin{aligned} g'(m, n) = g(m, n) * h(m, n) &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(i, j) \cdot h(m-i, n-j) \\ &= \sum_{i=0}^2 \sum_{j=0}^2 g(i, j) \cdot h(m-i, n-j) \end{aligned}$$

$$\begin{aligned}
g'(0,0) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h(0-i, 0-j) \\
&= g(0,0) \cdot h(0,0) + g(0,1) \cdot h(0,-1) + g(0,2) \cdot \underbrace{h(0,-2)}_{=0} + \\
&\quad + g(1,0) \cdot h(-1,0) + g(1,1) \cdot h(-1,-1) + g(1,2) \cdot \underbrace{h(-1,-2)}_{=0} + \\
&\quad + g(2,0) \cdot \underbrace{h(-2,0)}_{=0} + g(2,1) \cdot \underbrace{h(-2,-1)}_{=0} + g(2,2) \cdot \underbrace{h(-2,-2)}_{=0} \\
&= 1 \cdot 0 + 4 \cdot (-2) + 2 \cdot 0 + 5 \cdot (-1) \\
&= -13 \\
g'(1,0) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h(1-i, 0-j) \\
&= g(0,0) \cdot h(1,0) + g(0,1) \cdot h(1,-1) + g(0,2) \cdot \underbrace{h(1,-2)}_{=0} + \\
&\quad + g(1,0) \cdot h(0,0) + g(1,1) \cdot h(0,-1) + g(1,2) \cdot \underbrace{h(0,-2)}_{=0} + \\
&\quad + g(2,0) \cdot h(-1,0) + g(2,1) \cdot h(-1,-1) + g(2,2) \cdot \underbrace{h(-1,-2)}_{=0} \\
&= 1 \cdot 0 + 4 \cdot (-1) + 2 \cdot 0 + 5 \cdot (-2) + 3 \cdot 0 + 6 \cdot (-1) \\
&= -20 \\
g'(2,0) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h(2-i, 0-j) \\
&= -17 \\
g'(0,1) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h(0-i, 1-j) \\
&= -18 \\
g'(1,1) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h(1-i, 1-j) \\
&= -24 \\
&\vdots
\end{aligned}$$

obraz wyjściowy:

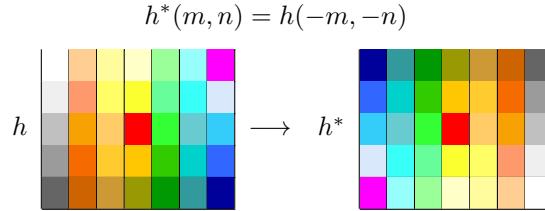
		m		
		0	1	2
n		0	-13	-20
		1	-18	-24
2		13	20	17

$$g'(m, n)$$

2. Wyznaczenie splotu z wykorzystaniem współczynników wag filtra  $h^*(m, n)$

$$\begin{aligned}
 g'(m, n) = g(m, n) * h(m, n) &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(i, j) \cdot h(m - i, n - j) \\
 &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(i, j) \cdot h(-(i - m), -(j - n)) \\
 &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(i, j) \cdot \underbrace{h(-(i - m), -(j - n))}_{=h^*(i-m, j-n)}
 \end{aligned}$$

tzn.



→ Jądro splotu zostaje jest odbite symetrycznie zarówno w kierunku poziomym, jak i pionowym.

$$\begin{aligned}
 g'(m, n) &= g(m, n) * h(m, n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} g(i, j) \cdot h^*(i - m, j - n) \\
 &= \sum_{i=0}^2 \sum_{j=0}^2 g(i, j) \cdot h^*(i - m, j - n)
 \end{aligned}$$

- (a) wyznaczenie  $h^*(m, n)$  (symetryczne odbicie  $h(m, n)$  w kierunku poziomym i pionowym lub obrócenie o  $180^\circ$ )

			<i>m</i>
			-1      0      1
			<i>n</i>
			-1      -1      -1
			0 <i>h(0,0)</i> 0
			1      2      1
			→
			<i>n</i>
			-1      1      2
			0 <i>h*(0,0)</i> 0
			1      -1      -2      -1
			<i>h(m, n)</i>
			<i>h*(m, n)</i>

- (b)  $h^*(m, n)$  zostaje przesunięte na obraz wejściowy. Jeśli  $h^*(0, 0)$  jest wyśrodkowane (wyrównane) dokładnie w interesującym punkcie  $(x, y)$ , należy pom-

nożyć wartości jądra przez nakładające się dane wejściowe.

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

$$\begin{aligned}
 g'(0,0) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h^*(i,j) \\
 &= 1 \cdot 0 + 4 \cdot (-2) + 2 \cdot 0 + 5 \cdot (-1) \\
 &= -13
 \end{aligned}$$

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

$$\begin{aligned}
 g'(1,0) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h^*(i-1,j) \\
 &= 1 \cdot 0 + 4 \cdot (-1) + 2 \cdot 0 + 5 \cdot (-2) + 3 \cdot 0 + 6 \cdot (-1) \\
 &= -20
 \end{aligned}$$

1	2	1
0	2	0
4	-1	-2

$$\begin{aligned}
 g'(2,0) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h^*(i-2,j) \\
 &= 2 \cdot 0 + 5 \cdot (-1) + 3 \cdot 0 + 6 \cdot (-2) \\
 &= -17
 \end{aligned}$$

1	2	1	3
0	0	0	6
-1	-2	-1	9

$$\begin{aligned}
 g'(0,1) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h^*(i,j-1) \\
 &= 1 \cdot 2 + 4 \cdot 0 + 7 \cdot (-2) + 2 \cdot 1 + 5 \cdot 0 + 8 \cdot (-1) \\
 &= -18
 \end{aligned}$$

1	2	1
0	5	0
-1	-2	-1

$$\begin{aligned}
 g'(1,1) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h^*(i-1,j-1) \\
 &= 1 \cdot 1 + 4 \cdot 0 + 7 \cdot (-2) + 2 \cdot 2 + 5 \cdot 0 + 8 \cdot (-2) + 3 \cdot 1 + 6 \cdot 0 + 9 \cdot \\
 &= -24
 \end{aligned}$$

1	2	3
4	0	0
7	-1	-1

$$\begin{aligned}
 g'(2,1) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h^*(i-2,j-1) \\
 &= -18
 \end{aligned}$$

1	2	3
2	4	5
0	7	8

$$\begin{aligned}
 g'(0,2) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h^*(i,j-2) \\
 &= 13
 \end{aligned}$$

	1	2	3
1	4	2	1
4	5	6	
0	8	0	9

$$\begin{aligned} g'(1,2) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h^*(i-1, j-2) \\ &= 20 \end{aligned}$$

	1	2	3
4	1	2	6
7	0	8	0
-1	-2	-1	

$$\begin{aligned} g'(2,2) &= \sum_{i=0}^2 \sum_{j=0}^2 g(i,j) \cdot h^*(i-2, j-2) \\ &= 17 \end{aligned}$$

**Przykład: splot obrazu z filtrem wykrywającym poziome krawędzie**  
 → grafika 8.7

### 8.3.2 Własności splotu

1. Operacja splotu liniowego jest przemienna

$$g * h = h * g$$

2. Liniowość splotu

- (a) mnożenie obrazu przez stałą skalarną  $a$

$$(a \cdot g) * h = g * (a \cdot h) = a \cdot (g * h)$$

- (b) suma dwóch obrazów

$$(g_1 + g_2) * h = g_1 * h + g_2 * h$$

Filtrowanie sumy obrazów daje taki sam efekt jak suma pojedynczo filtrowanych obrazów.

3. Splot liniowy jest asocjacyjny

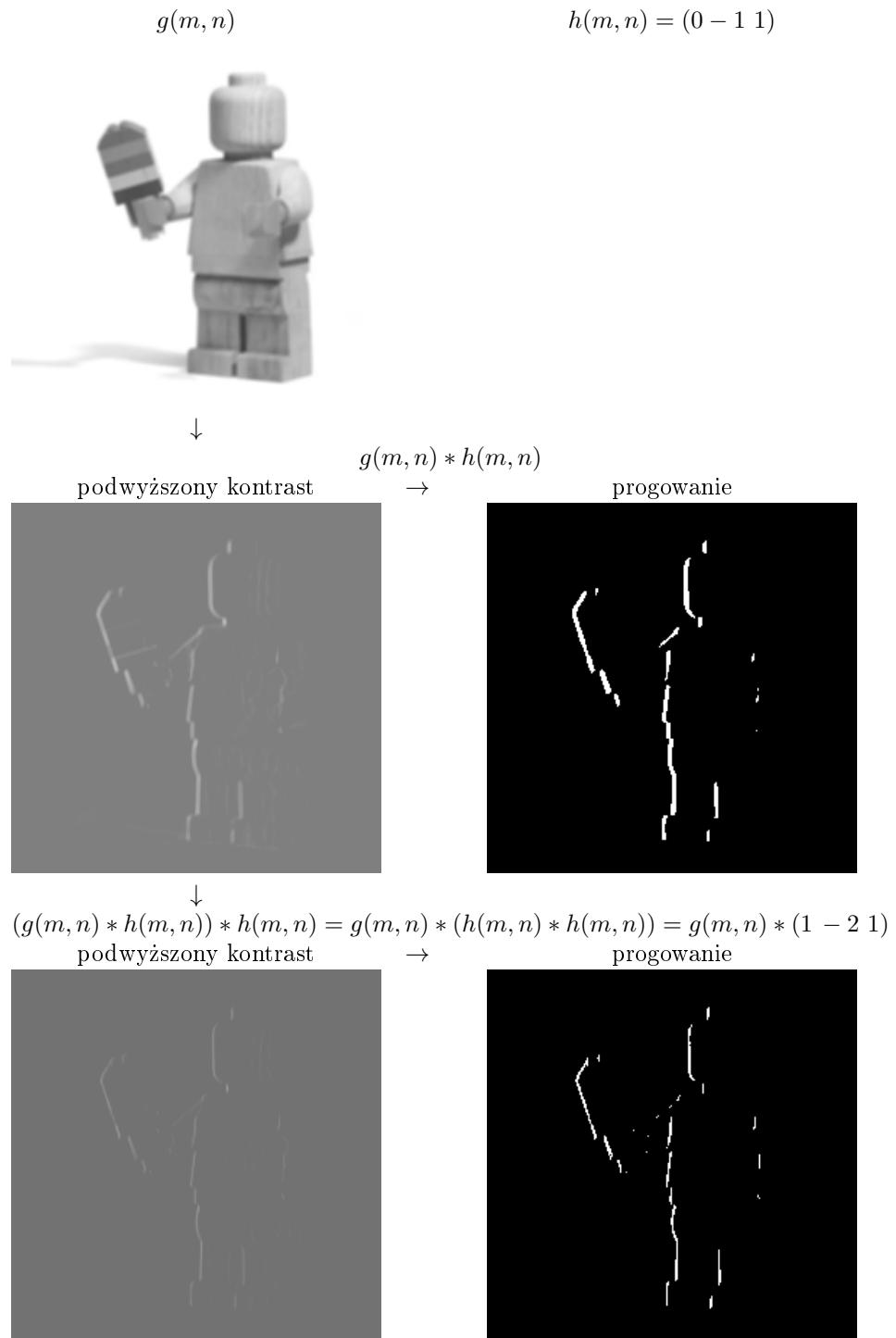
$$g * (h_1 * h_2) = (g * h_1) * h_2$$

Kolejność operacji filtracyjnych wykonywanych jedna po drugiej nie ma znaczenia. Można zatem dowolnie zmieniać kolejność dla kilku kolejnych filtrów, a także łączyć kilka filtrów w nowe filtry.

Bezpośrednią konsekwencją własności 3 jest możliwość definicji jądra filtru  $h(m, n)$  jako splotu dwóch lub większej liczby (mniejszych) jąder splotu  $h_1, h_2, \dots, h_n$  w taki sposób, że  $h = h_1 * h_2 \dots * h_n$ .

**Separacja**, tzn. operacja filtrowania  $g * h$  jako seria „mniejszych” operacji filtrujących, może prowadzić do redukcji złożoności obliczeniowej.

Ważną formą separacji jest możliwość rozdzielenia dwuwymiarowego filtru  $h$  na dwa jednowymiarowe filtry  $h_x$  i  $h_y$ .



Grafika 8.7: Wykrywanie poziomych krawędzi w obrazie

### Przykład

Wynikiem operacji splotu dwóch jednowymiarowych filtrów  $h_x$  i  $h_y$

$$h_x = \frac{1}{5} \cdot (1 \ 1 \ 1 \ 1 \ 1) \quad \text{i} \quad h_y = \frac{1}{3} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

jest

$$h_{xy} = h_x * h_y = \frac{1}{15} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

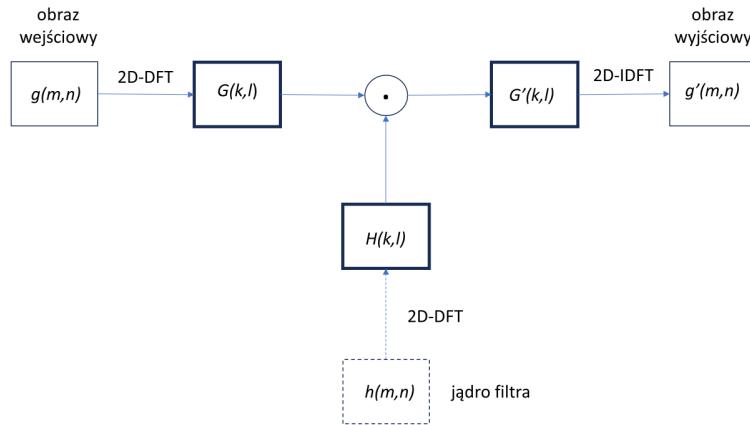
Dwuwymiarowy filtr uśrednienia lokalnego  $h_{xy}$  można podzielić na dwa jednowymiarowe filtry. W powyższym przypadku cała operacja splotu wymaga  $3 \cdot 5 = 15$  operacji arytmetycznych na piksel, w przypadku separacji splotu tylko  $5 + 3 = 8$  operacji na piksel.

### 8.3.3 Filtracja w dziedzinie widma (spektrum Fouriera)

Dla transformaty Fouriera zachodzi twierdzenie

$$T\{g * h\} = T\{g\} \cdot T\{h\}$$

Transformata splotu dwóch funkcji w przestrzeni odpowiada mnożeniu ich transformat w przestrzeni Fouriera. —> grafika 8.8



Grafika 8.8: Filtracja obrazu w dziedzinie widma

Przykład: —> grafika 8.9

### Znaczenie dla przetwarzania obrazu:

- operację splotu obrazu  $g(m, n)$  z jądrem filtra  $h(m, n)$  można zastąpić mnożeniem transformaty Fouriera obrazu przez transformatę jądra filtra  $h(m, n)$ , a dla wyniku mnożenia wykonać odwrotną DFT,
- definicje filtrów w przestrzeni i dziedzinie częstotliwości są równoważne,
- w przypadku dużych jąder filtrów szybsza może być operacja filtrowania poprzez transformację Fouriera.

## 8.4 Charakterystyki filtrów liniowych

→ grafika 8.10

### 8.4.1 Filtry dolnoprzepustowe

Filtry dolnoprzepustowe odfiltrowują wysokie częstotliwości i przepuszczają niskie częstotliwości. Dlatego nadają się do redukcji szumu lub jako opery wygładzające.

Idealne filtry dolnoprzepustowe → grafika 8.11

Rzeczywiste filtry dolnoprzepustowe są aproksymacją filtrów idealnych, ich celem jest redukcja efektu Gibbsa. W przetwarzaniu obrazów najczęściej stosuje się aproksymację w przestrzeni obrazu, a nie w przestrzeni transformaty Fouriera. Jednak niemożliwa jest definicja filtra, który usunie cały szum, pozostawiając nienaruszone ważne struktury w obrazie, ponieważ żaden filtr nie jest w stanie rozróżnić, które struktury są ważne dla odbiorcy, a które nie.

### Filtr zdefiniowane w dziedzinie przestrzeni

→ grafika 8.12

Najczęściej stosowane liniowe filtry dolnoprzepustowe to filtr uśredniający, filtr Gaussa, filtr o współczynnikach dwumianu Newtona.

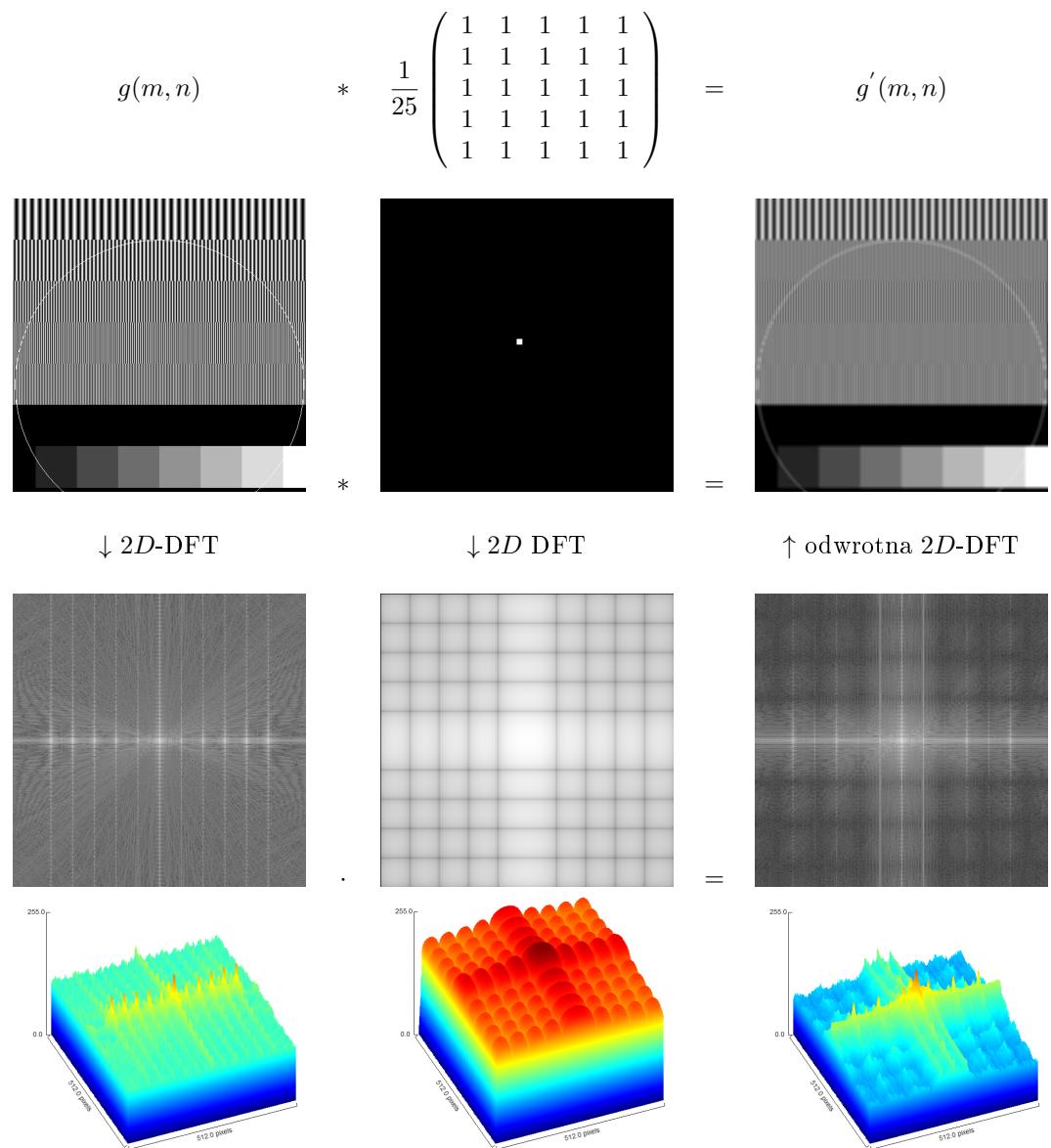
### Filtr uśredniający

→ jądro filtra składa się z 1-ek pomnożonych przez współczynnik normalizacji:

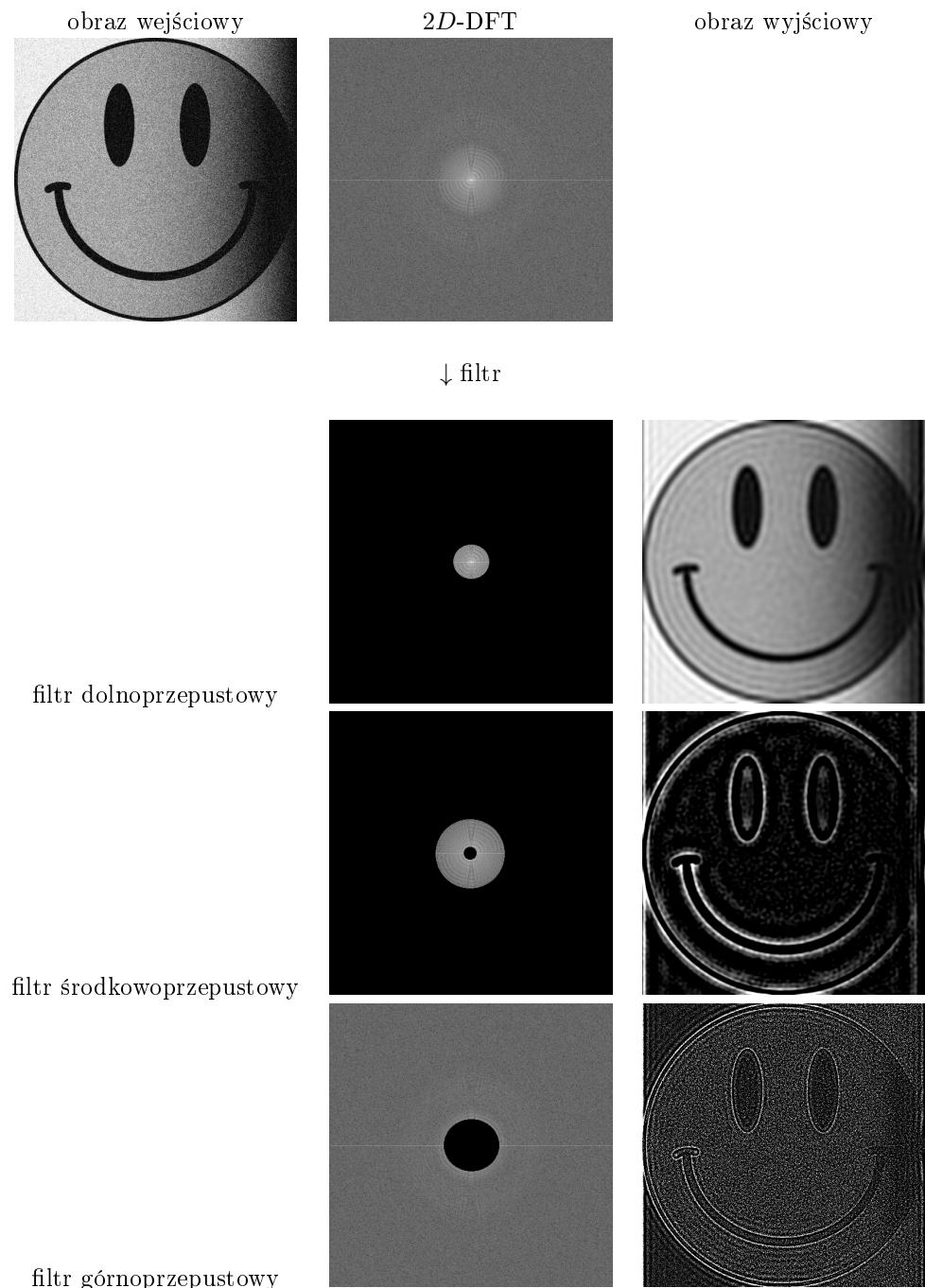
$$h = \frac{1}{9} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad h = \frac{1}{25} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Filtr uśredniający nie jest to optymalnym rozwiązaniem ze względu na ostre krawędzie jądra i związane z tym zachowanie w dziedzinie częstotliwości (→ grafika 8.13). Dodatkowo wszystkim elementom obrazu przypisuje się tę samą wagę, a środek nie jest ważony mocniej niż krawędzie.

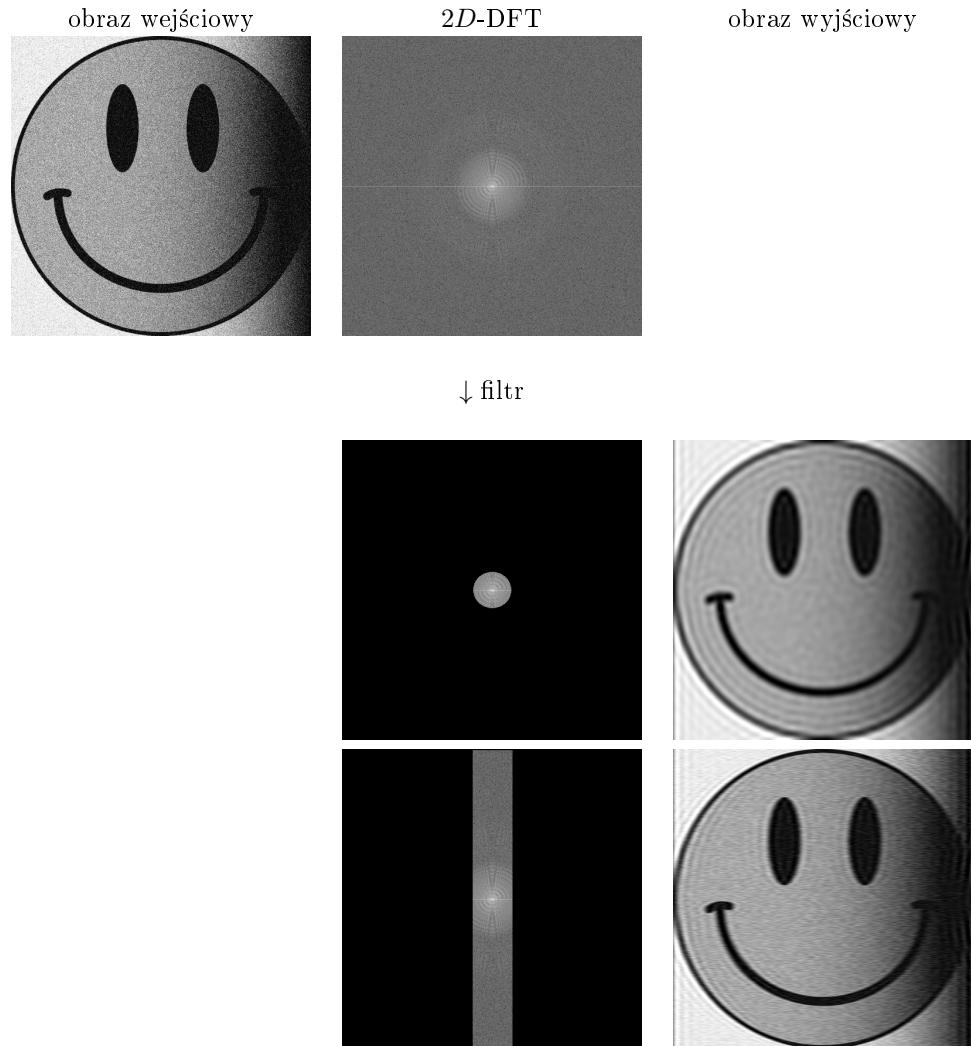
Redukcja szumu przez filtr uśredniający → grafika 8.14



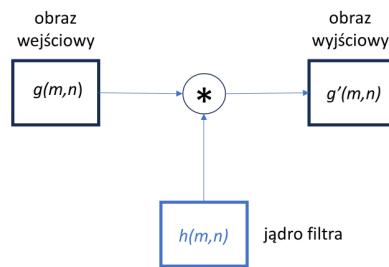
Grafika 8.9: Filtracja w spektrum Fouriera

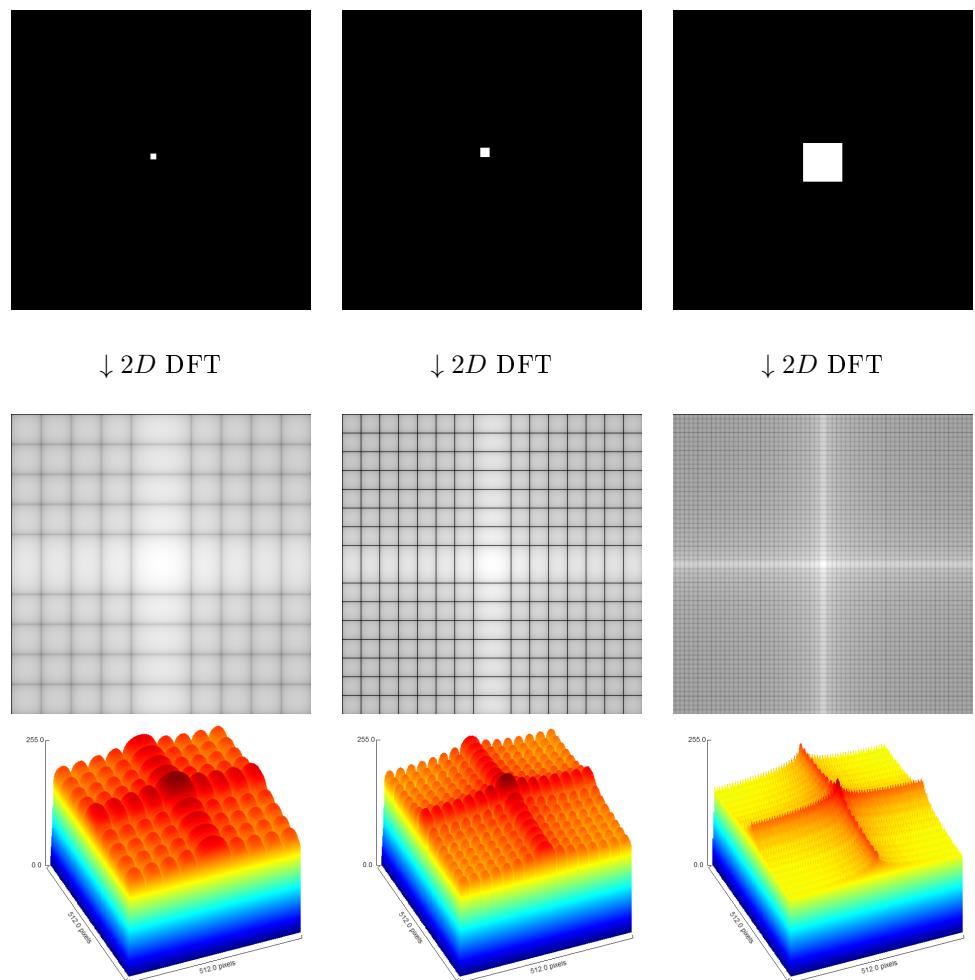


Grafika 8.10: Charakterystyki idelanych filtrów (filtr środkowoprzepustowy i górnoprzepustowy z powiększonym kontrastem)

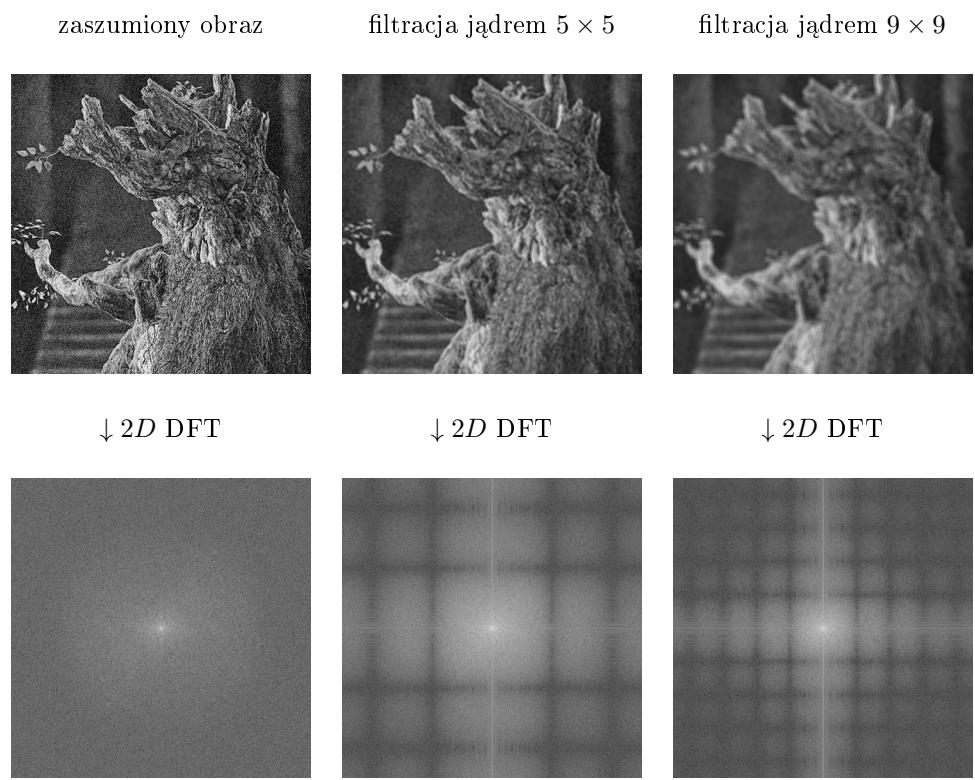


Grafika 8.11: Idealne filtry dolnoprzepustowe

Grafika 8.12: Definicja filtra  $h(m,n)$  w dziedzinie przestrzeni



Grafika 8.13: Transformata Fouriera dla jąder filtra uśredniającego



Grafika 8.14: Redukcja szumu przez filtr uśredniający

### Filtr Gaussa

→ współczynniki jądra filtra są obliczone z (dyskretnej, dwuwymiarowej) funkcji Gaussa

$$G_\sigma(x, y) = \exp^{-\frac{x^2+y^2}{2\sigma^2}},$$

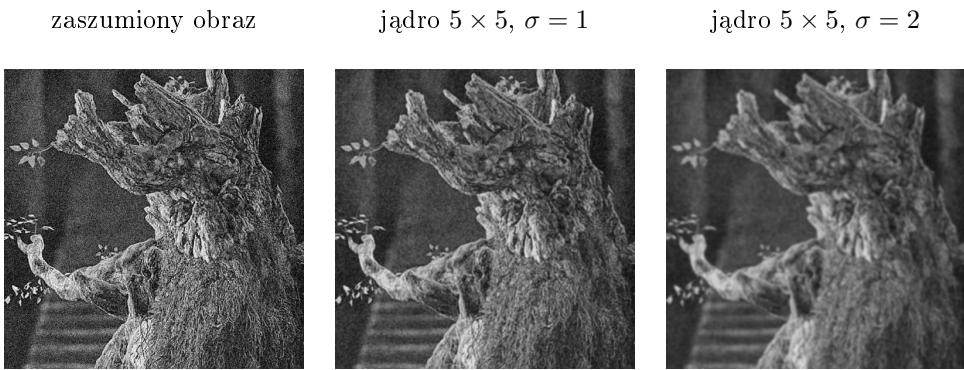
gdzie odchylenie standardowe  $\sigma$  określa "promień" funkcji dzwonowej.

Środkowy element obrazu otrzymuje maksymalną wagę, wartości pozostałych współczynników maleją w sposób ciągły i równomiernie we wszystkich kierunkach (izotropowo) wraz ze wzrostem odległości od środka.

Jądro filtra dla filtru Gaussa  $3 \times 3$  z  $\sigma = 0.5$  otrzymuje się po podstawieniu dla parametrów  $x$  i  $y$  wartości ze zbioru  $\{-1, 0, 1\}$  i normalizacji (ponieważ jasność obrazu nie powinna się zmieniać, suma wszystkich współczynników musi wynosić 1, co osiąga się poprzez podzielenie wszystkich współczynników przez ich sumę):

$$h_{0.5} = \begin{pmatrix} 0.011 & 0.084 & 0.011 \\ 0.084 & 0.619 & 0.084 \\ 0.011 & 0.084 & 0.011 \end{pmatrix}$$

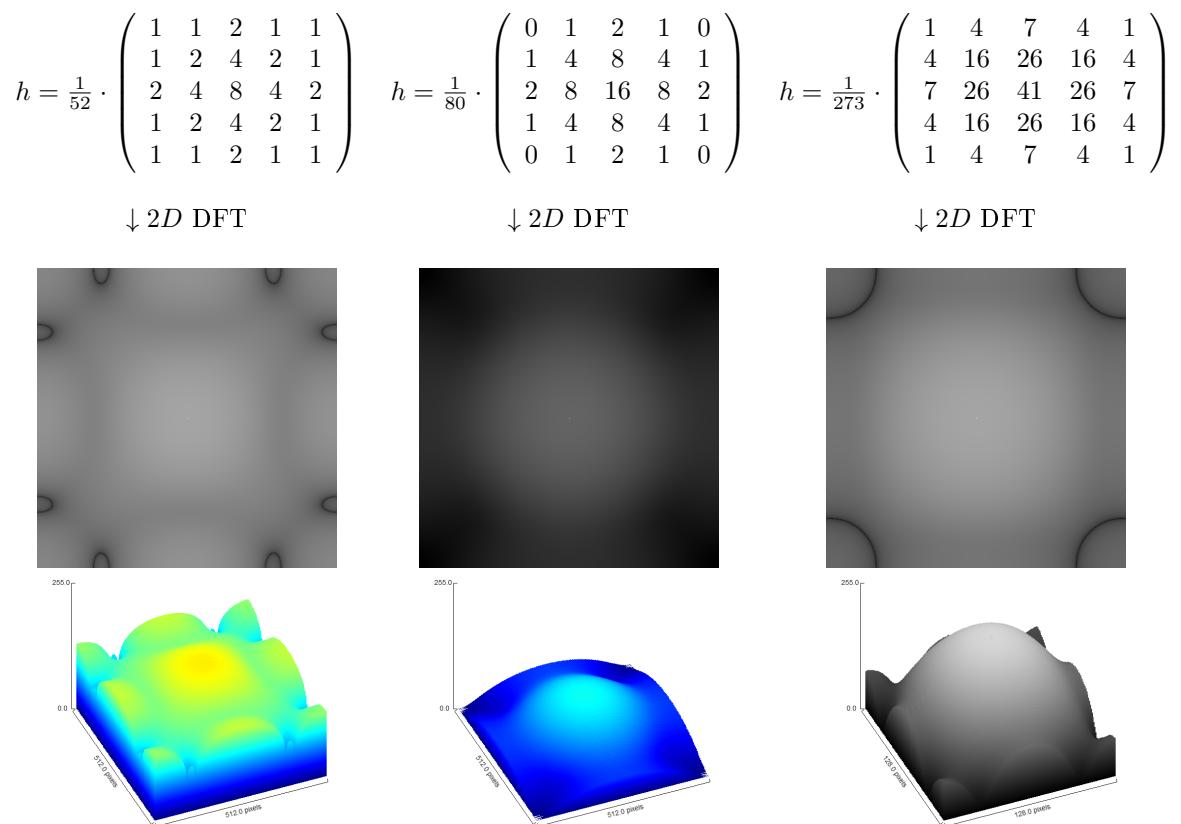
W przypadku filtru Gaussa większe jądro filtra przy zachowaniu parametru  $\sigma$  zapewnia lepsze przybliżenie funkcji Gaussa, ale w rzeczywistości nie powoduje znacznego polepszenia funkcji wygładzania obrazu. (→ grafika 8.15)



Grafika 8.15: Redukcja szumu przez filtr Gaussa

Skalę wygładzania można zmieniać za pomocą odchylenia standardowego  $\sigma$ . Standardowe filtry Gaussa o rozmiarze jądra  $5 \times 5$ , w których skalowanie transformaty Fouriera regulowane jest poprzez parametr  $\sigma$  → grafika 8.16

Realizacja filtra Gaussa w programach graficznych → grafika 8.17

Grafika 8.16: Transformaty Fouriera dla filtrów Gaussa o wymiarach  $5 \times 5$ Grafika 8.17: Filtr Gaussa ( $3 \times 3$ ) w oprogramowaniu Blender. Autor: P. Arłukowicz

### Filtr o współczynnikach dwumianu Newtona (*binomial filter*)

→ (znormalizowany) splot dwóch filtrów jednowymiarowych, których współczynniki są dwumianami Newtona:

$$h_x = \frac{1}{2^n} \cdot ((\binom{n}{0} \ \binom{n}{1} \ \binom{n}{2} \ \dots \ \binom{n}{n}) \quad \text{i} \quad h_y = \frac{1}{2^n} \cdot \begin{pmatrix} \binom{n}{0} \\ \binom{n}{1} \\ \vdots \\ \binom{n}{n} \end{pmatrix}$$

Przykłady:

1. jądro  $3 \times 3$

$$h_x = \frac{1}{2^2} \cdot ((\binom{2}{0} \ \binom{2}{1} \ \binom{2}{2})) = \frac{1}{4} \cdot (1 \ 2 \ 1) \quad \text{i} \quad h_y = \frac{1}{2^2} \cdot \begin{pmatrix} \binom{2}{0} \\ \binom{2}{1} \\ \binom{2}{2} \end{pmatrix} = \frac{1}{4} \cdot \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

Obliczenie splotu  $3 \times 3$  z definicji:

$$h_x = (h(-1, 0) \ h(0, 0) \ h(1, 0)) = \frac{1}{4} \cdot (1 \ 2 \ 1) \quad \text{i} \quad h_y = \begin{pmatrix} h(0, -1) \\ h(0, 0) \\ h(0, 1) \end{pmatrix} = \frac{1}{4} \cdot \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

$$\begin{aligned} h(m, n) = h_x(m, n) * h_y(m, n) &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h_x(i, j) \cdot h_y(m-i, n-j) \\ &= \sum_{i=-1}^1 h_x(i, 0) \cdot h_y(m-i, n) \end{aligned}$$

$$h(-1, -1) = \sum_{i=-1}^1 h_x(i, 0) \cdot h_y(-1-i, -1) = h_x(-1, 0) \cdot h_y(0, -1) = \frac{1}{16} \cdot 1$$

$$h(-1, 0) = \sum_{i=-1}^1 h_x(i, 0) \cdot h_y(-1-i, 0) = h_x(-1, 0) \cdot h_y(0, 0) = \frac{1}{16} \cdot 2$$

$$h(-1, 1) = \sum_{i=-1}^1 h_x(i, 0) \cdot h_y(-1-i, 1) = h_x(-1, 0) \cdot h_y(0, 1) = \frac{1}{16} \cdot 1$$

$$h(0, -1) = \sum_{i=-1}^1 h_x(i, 0) \cdot h_y(0-i, -1) = h_x(0, 0) \cdot h_y(0, -1) = \frac{1}{16} \cdot 2$$

$$h(0, 0) = \sum_{i=-1}^1 h_x(i, 0) \cdot h_y(0-i, 0) = h_x(0, 0) \cdot h_y(0, 0) = \frac{1}{16} \cdot 4$$

⋮

$$h = h_x * h_y = \left[ \frac{1}{4} \cdot (1 \ 2 \ 1) \right] * \left[ \frac{1}{4} \cdot \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \right] = \frac{1}{16} \cdot \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

## 2. jądro $5 \times 5$

$$h_x = \frac{1}{2^4} \cdot ((\begin{pmatrix} 4 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 4 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 4 \\ 2 \end{pmatrix} \quad \begin{pmatrix} 4 \\ 3 \end{pmatrix} \quad \begin{pmatrix} 4 \\ 4 \end{pmatrix})) = \frac{1}{16} \cdot (1 \ 4 \ 6 \ 4 \ 1) \quad \text{i} \quad h_y = \frac{1}{2^4} \cdot \begin{pmatrix} \begin{pmatrix} 4 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 4 \\ 1 \end{pmatrix} \\ \begin{pmatrix} 4 \\ 2 \end{pmatrix} \\ \begin{pmatrix} 4 \\ 3 \end{pmatrix} \\ \begin{pmatrix} 4 \\ 4 \end{pmatrix} \end{pmatrix} = \frac{1}{16} \cdot \begin{pmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{pmatrix}$$

$$h = h_x * h_y = \frac{1}{256} \cdot \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

Filtry dwumianowe to bardzo przydatne filtry w grafice komputerowej, przetwarzaniu obrazu i przetwarzaniu sygnałów.

Korzyści:

- łatwe i precyzyjne obliczenie współczynników bez liczb zmiennoprzecinkowych
- separacja (możliwość wykonania operacji filtrowania jako serii prostych splotów jednowymiarowych)
- charakterystyka filtra bardzo zbliżona do filtra Gaussa (filtr Gaussa jest idealnie promienisty i izotropowy, podczas gdy filtr dwumianowy ma bardziej „prostokątną” transformatę Fouriera) → grafika 8.18

Porównanie filtrów dolnoprzepustowych o wymiarach  $5 \times 5$  → grafika 8.19.

Ogólne wnioski dla liniowych filtrów dolnoprzepustowych zdefiniowanych w dziedzinie przestrzeni:

- Liniowe filtry dolnoprzepustowe eliminują (redukują) szумy, ale rozmywają krawędzie.
- Im większe jądro filtra, tym wyraźniejszy efekt operacji filtrowania.

## Filtrowanie różnych rodzajów szumu

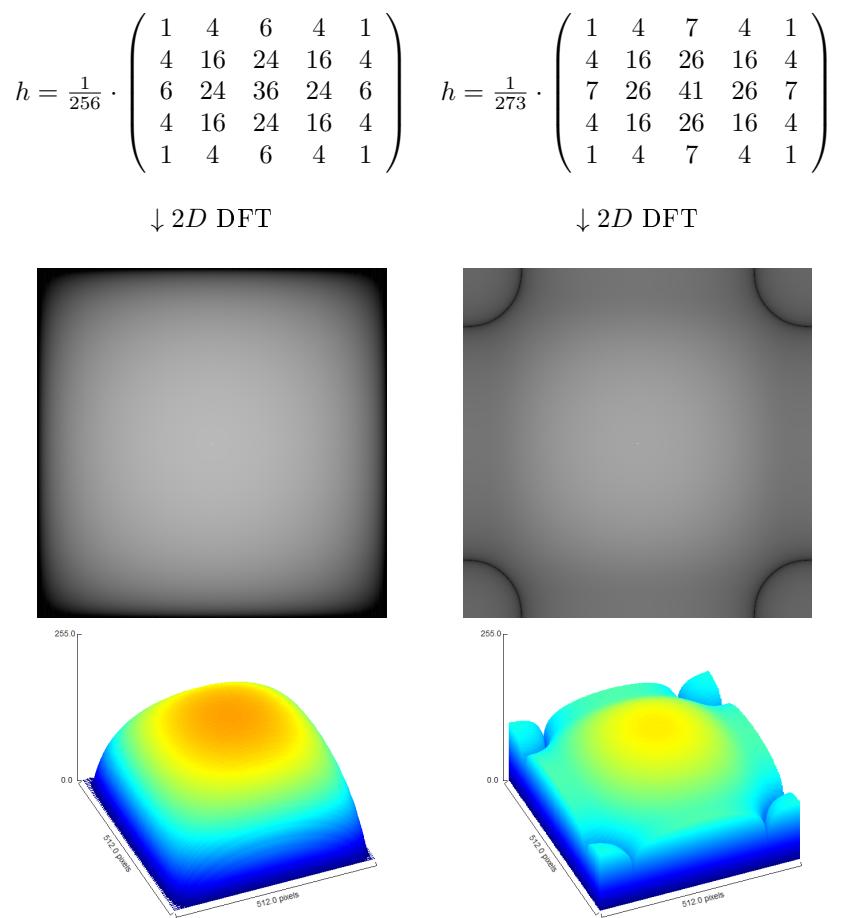
→ grafika 8.20

- Filtrowanie liniowe działa dobrze w przypadku szumu addytywnego ze średnią wartością 0.
- Filtrowanie liniowe działa źle w przypadku szumu nieaddytywnego.

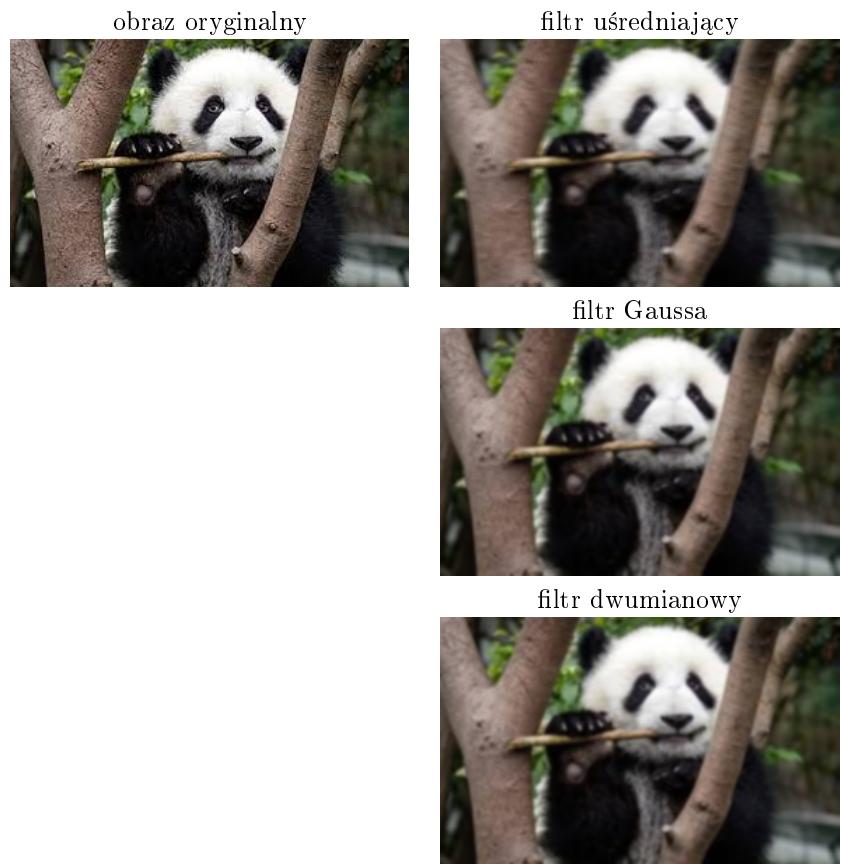
## Filtry zdefiniowane w dziedzinie częstotliwości

→ grafika 8.21

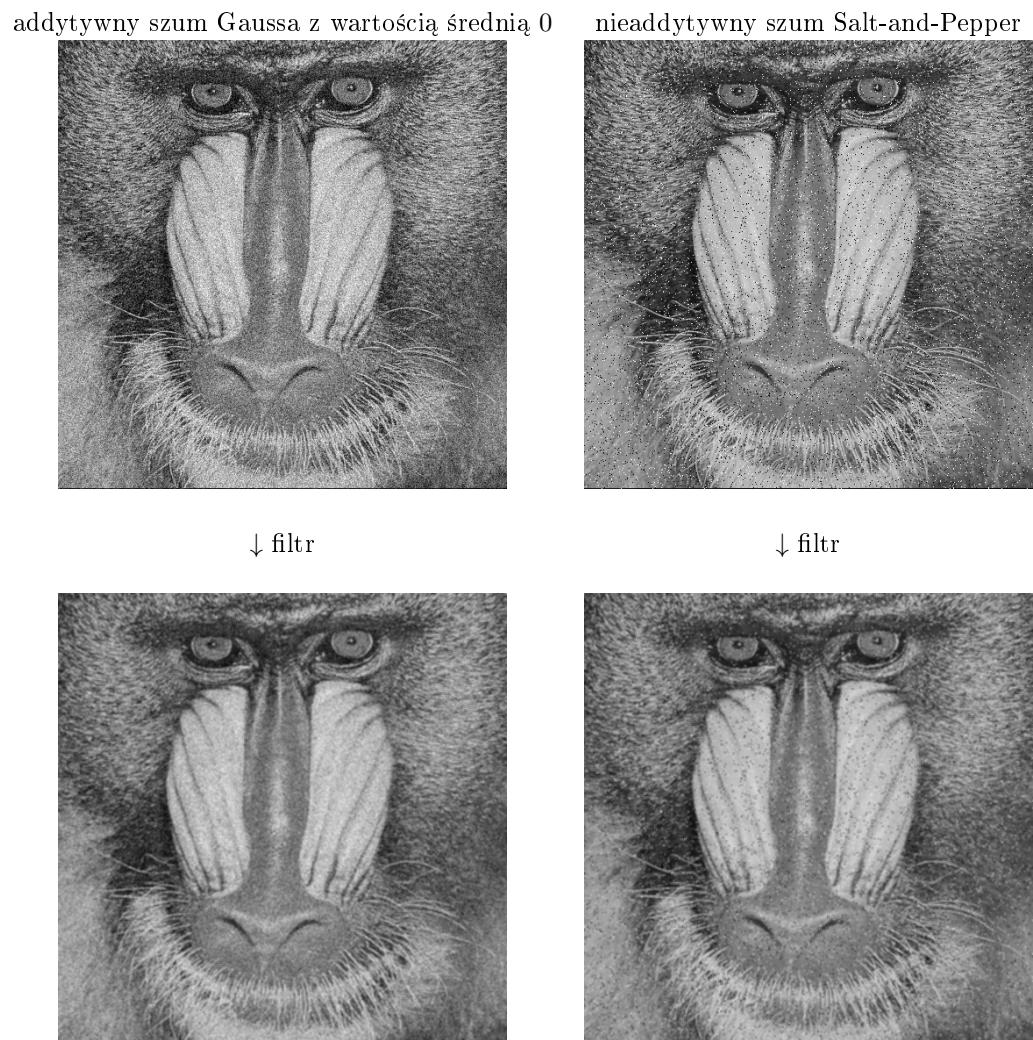
Najczęściej stosowane liniowe filtry dolnoprzepustowe zdefiniowane w dziedzinie częstotliwości to filtr idealny, filtr Butterwortha, filtr Czebyszewa i filtr Gaussa.



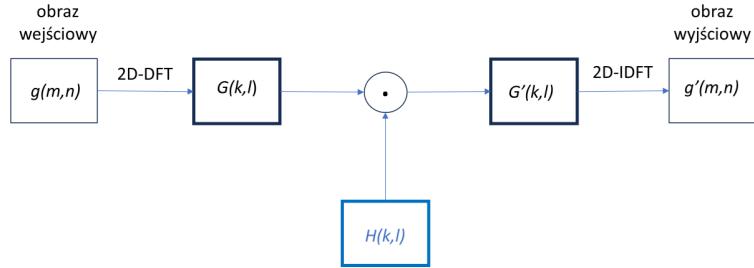
Grafika 8.18: Transformaty Fouriera dla filtrów dwumianowego i Gaussa o wymiarach  $5 \times 5$



Grafika 8.19: Porównanie filtrów dolnoprzepustowych o wymiarach  $5 \times 5$



Grafika 8.20: Filtrowanie szumów filtrem uśredniającym  $3 \times 3$



Grafika 8.21: Definicja filtra  $H(k,l)$  w dziedzinie widma

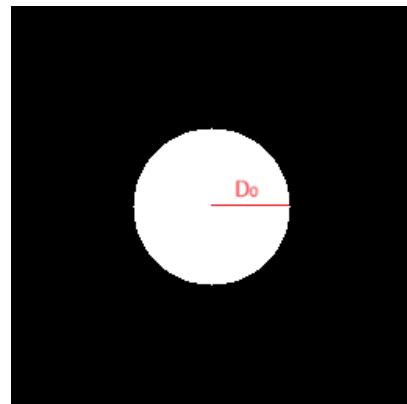
### Idealny filtr dolnoprzepustowy

Definicja (w przypadku filtra symetrycznego rotacyjnie) → grafika 8.22:

$$H(k,l) = \begin{cases} 1 & : D(k,l) \leq D_0 \\ 0 & : D(k,l) > D_0 \end{cases}$$

$$D(k,l) = \sqrt{k^2 + l^2}$$

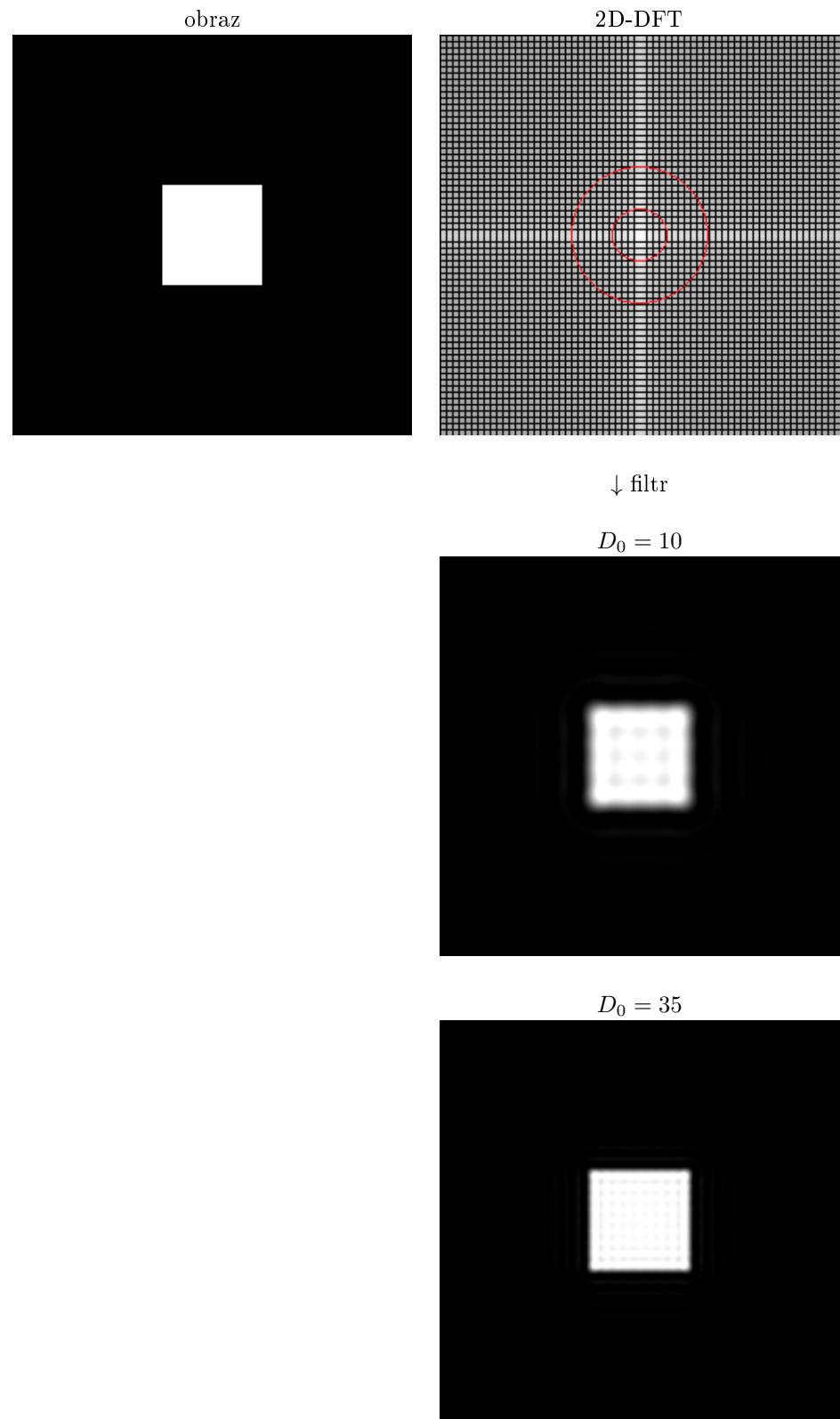
Przykład: → grafika 8.23



Grafika 8.22: Idealny, symetryczny rotacyjnie filtr dolnoprzepustowy

### Filtr Butterwortha

→ grafika 8.24



Grafika 8.23: Obraz filtrowany idealnym filtrem dolnoprzepustowym o promieniu  $D_0$

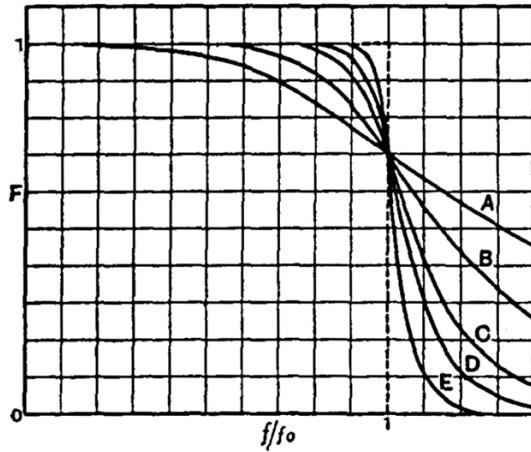


Fig. 3.

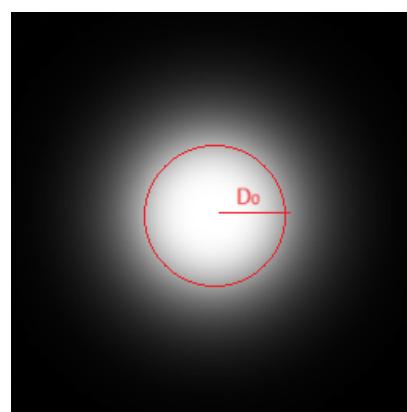
Grafika 8.24: Butterworth, S. (1930). "On the Theory of Filter Amplifiers", Experimental Wireless and the Wireless Engineer. 7: 536–541

Definicja filtra stopnia  $n \rightarrow$  grafika 8.25:

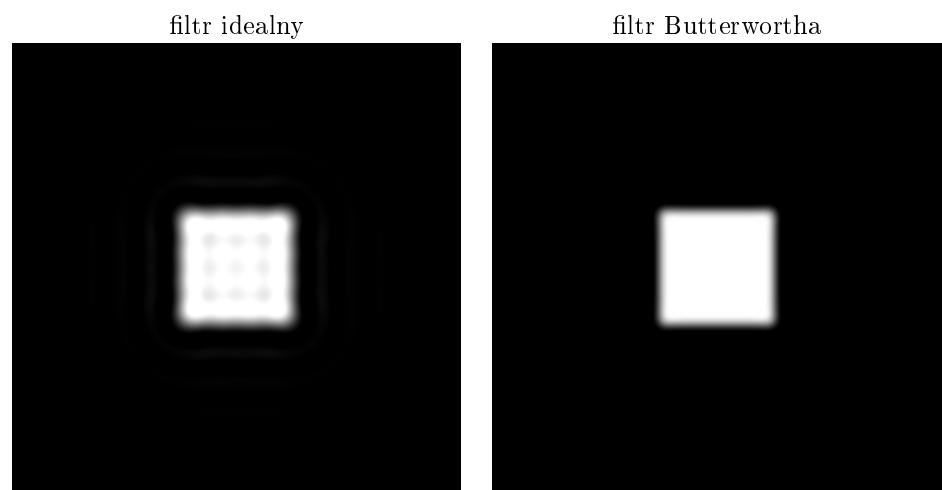
$$H(k, l) = \frac{1}{1 + (\sqrt{2} - 1) \left[ \frac{D(k, l)}{D_0} \right]^{2n}}, \quad n \in \mathbb{N}$$

$$D(k, l) = \sqrt{k^2 + l^2}$$

Przykład:  $\rightarrow$  grafika 8.26



Grafika 8.25: Filtr Butterwortha



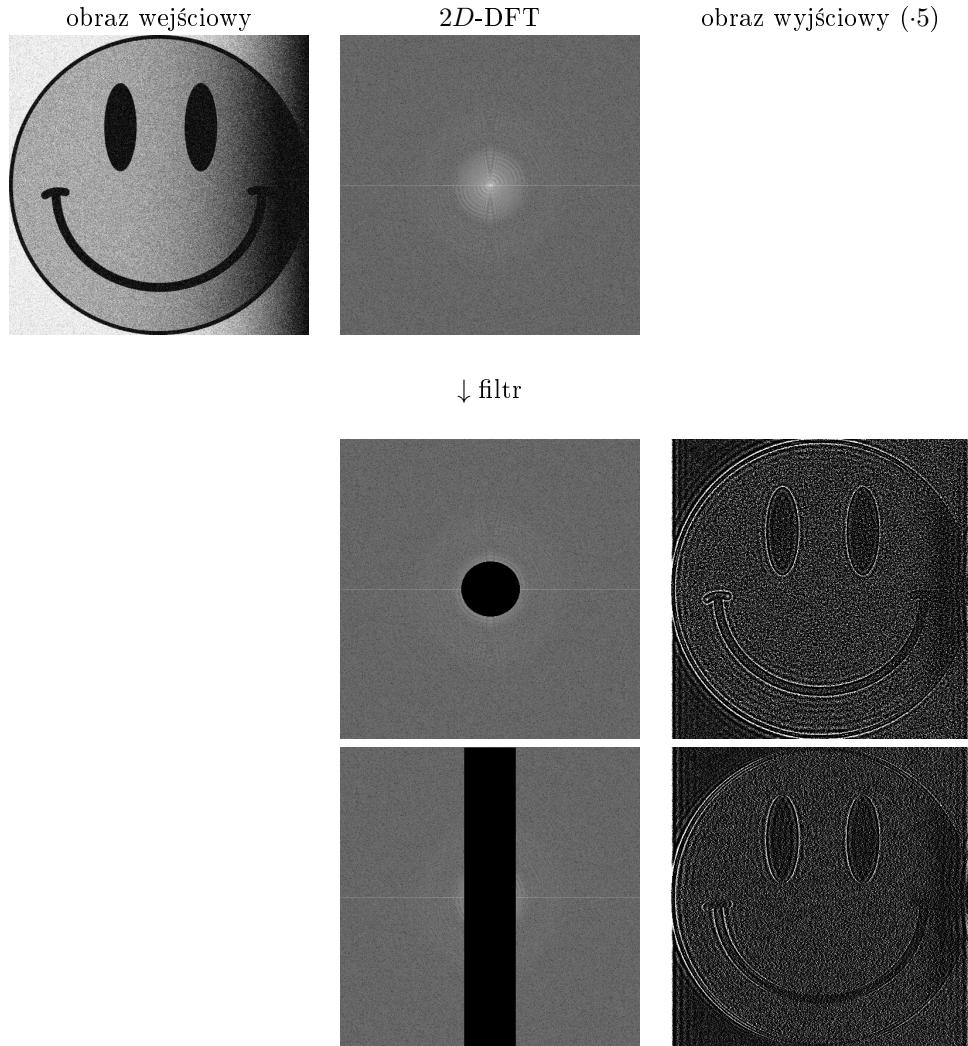
Grafika 8.26: Porównanie filtrów idealnego i Butterwortha o tej samej wartości granicznej  $D_0$

### 8.4.2 Filtry górnoprzepustowe

Filtry górnoprzepustowe odfiltrowują niskie częstotliwości i przepuszczają wysokie częstotliwości. Dlatego nadają się na przykład do wykrywania krawędzi czy wyostrzenia obrazu. Tłumieniu natomiast ulegają elementy o niskiej częstotliwości, a ilość szumów zostaje zwiększena.

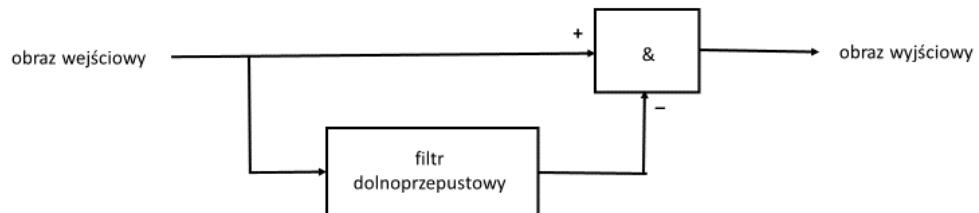
Idealne filtry górnoprzepustowe → grafika 8.27.

Rzeczywiste filtry górnoprzepustowe są aproksymacją filtrów idealnych.



Grafika 8.27: Idealne filtry górnoprzepustowe

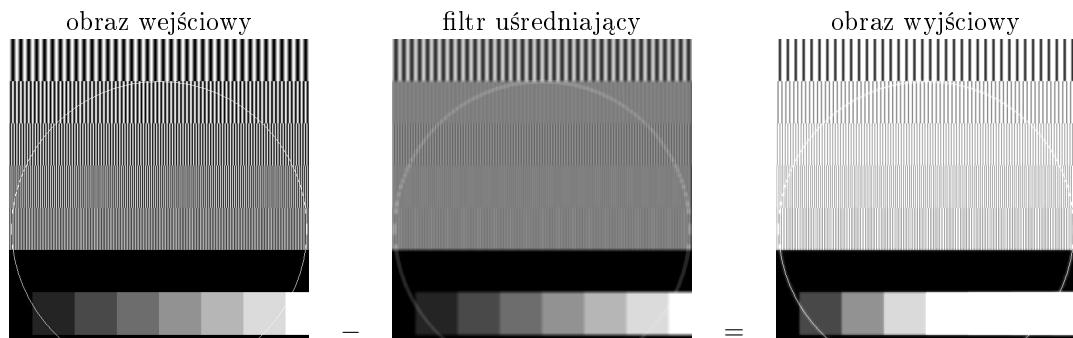
### Filtrowanie poprzez różnicę z obrazem wstępnie przetworzonym filtrem dolnoprzepustowym



Grafika 8.28: Filtracja obrazu poprzez różnicę obrazu wejściowego z obrazem filtrowanym filtrem dolnoprzepustowym

Wysokie częstotliwości obrazu można otrzymać poprzez odjęcie od obrazu wejściowego obrazu, w którym odfiltrowane zostały wysokie częstotliwości (filowanego przez filtr dolnoprzepustowy) → grafika 8.28.

Przykład → grafika 8.29.



Grafika 8.29: Różnica obrazu wyjściowego i filtrowanego filtrem uśredniającym z grafiki 8.9

### Filtry zdefiniowane w dziedzinie przestrzeni

→ grafika 8.12

### Filtry różnicowe

Jeżeli poszczególne współczynniki filtra są ujemne, działanie filtra można zinterpretować jako różnicę dwóch sum: sumy ważonej wszystkich pikseli z powiązanymi współczynnikami dodatnimi minus suma ważona pikseli o ujemnych współczynnikach w ob-

szarze filtra. Tworzenie różnic lokalnych wzmacnia ich widoczność. Dlatego obszarami zastosowań filtrów różnicowych są przede wszystkim uwydatnianie krawędzi i konturów oraz wyostrzanie obrazów.

Filtryle różnicowe są filtrami górnoprzepustowymi. Filtry górnoprzepustowe opierają się na pierwszej lub drugiej pochodnej funkcji obrazu  $g(m, n)$  w różnych kierunkach z różnym ważeniem poszczególnych pikseli:

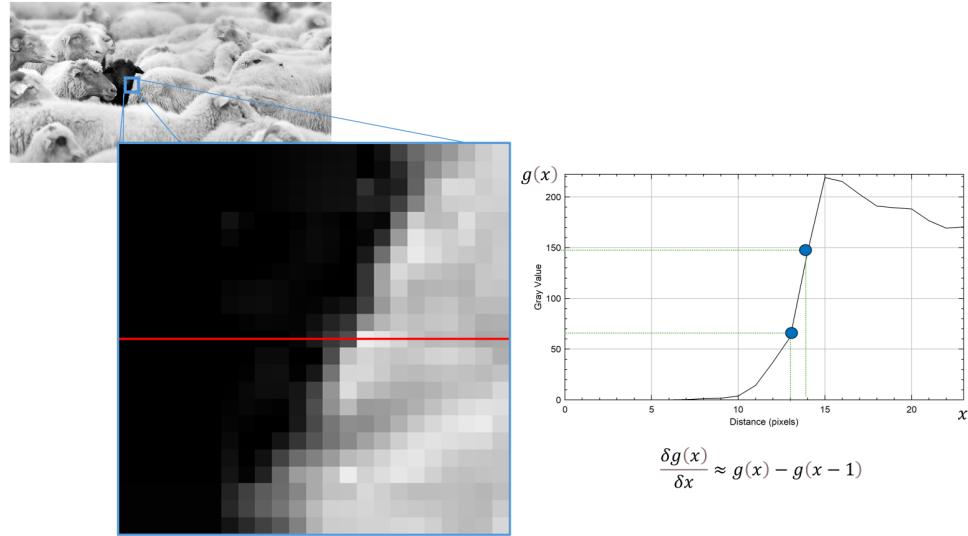
- filtry gradientowe  
→ filtry oparte na pierwszej pochodnej obrazu  $g(m, n)$
- filtry Laplace'a  
→ filtry oparte na drugiej pochodnej obrazu  $g(m, n)$

Po filtrowaniu obraz wyjściowy zwykle zawiera dodatnie i ujemne wartości szarości. Należy go znormalizować do dodatniego przedziału wartości szarości, na przykład do przedziału  $\{0, \dots, 255\}$ , stosując n.p. operację punktową, sprowadzić wartość do 0 lub wyznaczyć  $|g(m, n)|$ .

Najczęściej stosowane filtry różnicowe to gradientowe filtry kierunkowe, operator Prewitta, operator Sobela, filtr kompasowy, operator Kircha, filtr Laplaca i filtry wyostrzające lub uwypuklające obraz.

### Gradientowe filtry kierunkowe (*gradient directional*)

Motywacja → grafika 8.30



Grafika 8.30: Profil liniowy (wycinka) obrazu: przybliżenie pochodnej w kierunku  $x$ .

### 1. operatory różnicowe (poziomy, pionowy)

(a) poziomy  $\rightarrow$  grafika 8.31

$$\begin{aligned}\frac{\partial}{\partial x}g(x,y) &\approx g(x,y) - g(x-1,y) \\ \frac{\partial}{\partial x}g(x,y) &\approx g(x+1,y) - g(x,y)\end{aligned}$$

obraz wejściowy

$$\begin{aligned}h &= (0 \ 1 \ -1) \\ h &= (1 \ -1 \ 0)\end{aligned}$$



Grafika 8.31: Poziomy filtr gradientowy (operator różnicowy)

W ten sposób zdefiniowany gradient poziomy wykrywa (w większości programów przetwarzających obraz, w tym ImageJ) wyłącznie krawędzie dla  $g(x,y) > g(x-1,y)$ , tzn. kiedy pochodna przybiera wartość większą niż 0. Filtr gradientowy wykrywający krawędzie w przypadku  $g(x,y) < g(x-1,y)$  można zdefiniować analogicznie poprzez zmianę znaków współczynników filtra poziomego ( $\rightarrow$  grafika 8.32):

$$\begin{aligned}-\frac{\partial}{\partial x}g(x,y) &\approx g(x-1,y) - g(x,y) \\ -\frac{\partial}{\partial x}g(x,y) &\approx g(x,y) - g(x+1,y)\end{aligned}$$

(b) pionowy  $\rightarrow$  grafika 8.33

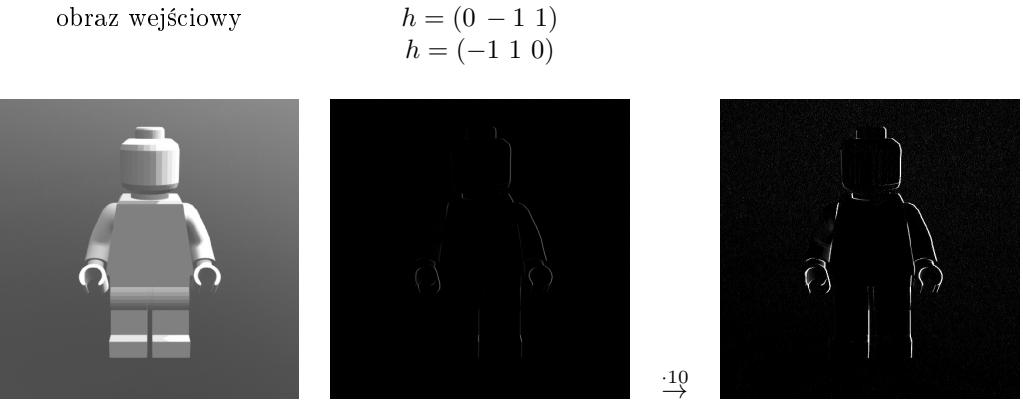
$$\begin{aligned}\frac{\partial}{\partial y}g(x,y) &\approx g(x,y) - g(x,y-1) \\ \frac{\partial}{\partial y}g(x,y) &\approx g(x,y+1) - g(x,y)\end{aligned}$$

Pionowy filtr gradientowy wykrywający krawędzie w przypadku  $g(x,y) < g(x,y-1)$   $\rightarrow$  grafika 8.34

## 2. operator Roberts'a (ukośny)

$\rightarrow$  wykrywa ukośne zmiany w obrazie  $\rightarrow$  grafika 8.35

- Wada: zarówno pionowe/poziome filtry różniczkowe jak i operatory Roberts'a są bardzo podatne na szum  $\rightarrow$  grafika 8.36



Grafika 8.32: Poziomy filtr gradientowy wykrywajacy ujemną pochodną

obraz wejściowy

$$h = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}, h = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$$



Grafika 8.33: Pionowy filtr gradientowy (operator różnicowy)

obraz wejściowy

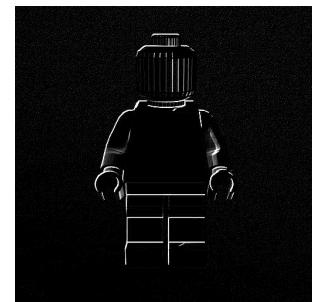
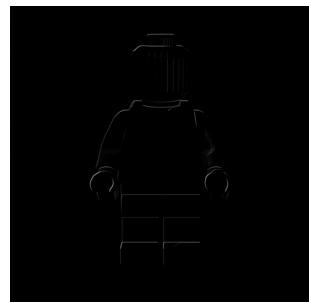
$$h = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}, h = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}$$



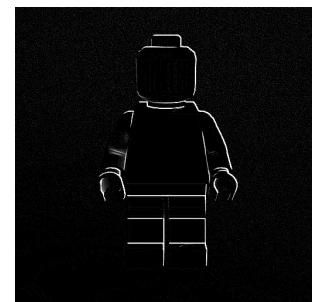
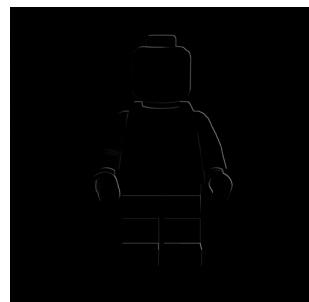
Grafika 8.34: Pionowy filtr gradientowy wykrywajacy ujemną pochodną

obraz wejściowy

$$h = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$



$$h = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$$



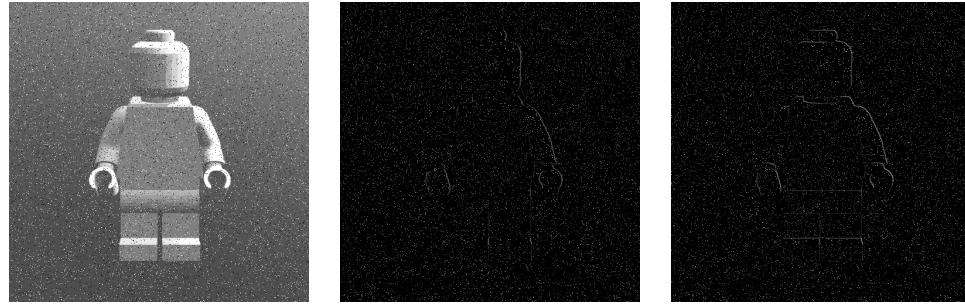
Grafika 8.35: Operatory Roberts'a

→ Potrzebne filtry, które jednocześnie uśredniają i obliczają pochodną (patrz poniżej)

obraz wejściowy

$$h = (0 \ -1 \ 1)$$

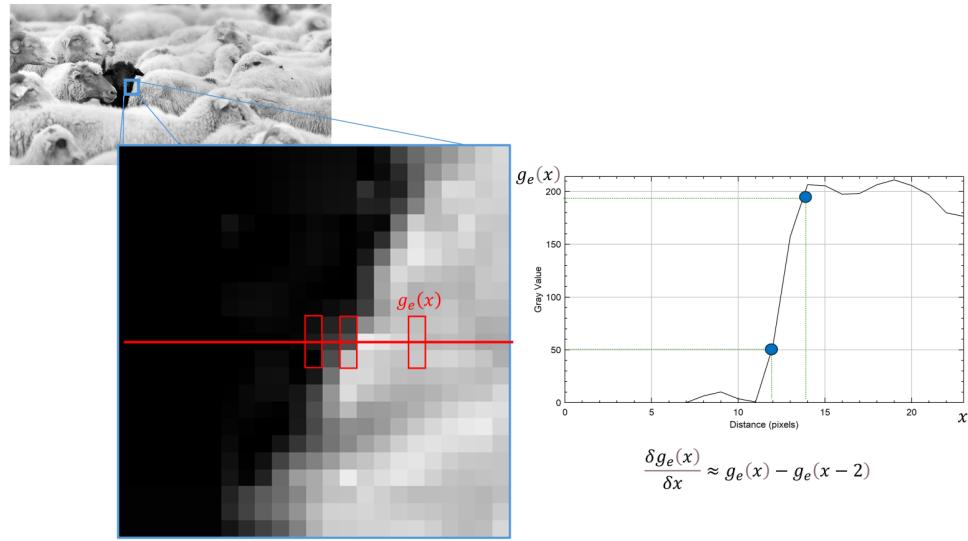
$$h = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$$



Grafika 8.36: Zaszumiony obraz filtrowany poziomym filtrem gradientowym i filtrem Roberts'a

### Operator Prewitta

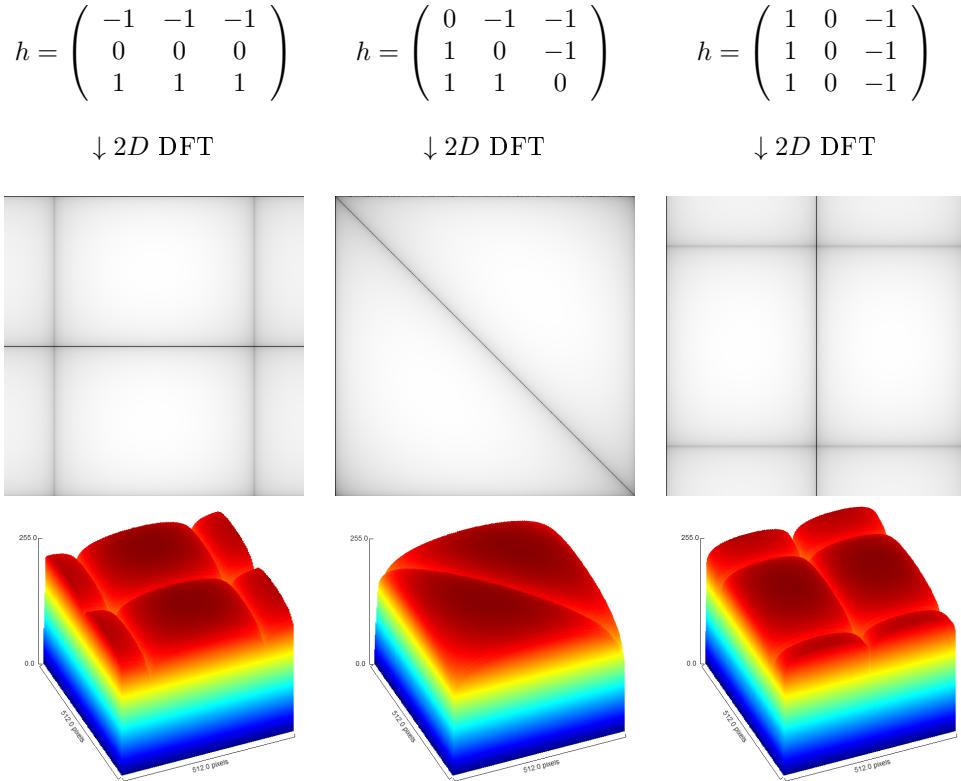
Motywacja → grafika 8.37



Grafika 8.37: Profil liniowy (wycinka) obrazu: przybliżenie pochodnej w kierunku  $x$  dla uśrednionych pikseli w kierunku  $y$

→ filtr gradientowy, który przybliża pochodną poprzez 2 piksele i uśrednia jak filtr uśredniający.

Istnieje 8 wariantów poprzez obrót o  $45^\circ$ . Przykłady → grafika 8.38



Grafika 8.38: Amplitudy transformaty Fouriera dla filtrów Prewitta

→ Operator Prewitta skutkuje "bardziej stabilnymi" krawędziami (grafika 8.39).  
Filtr Prewitta nazywany jest także **filtrem kompasowym**, ponieważ selektywnie wyodrębnia komponenty kierunkowe z obrazów (8 kierunków odpowiada kierunkom geograficznym, które są wykryte na obrazie wynikowym: wschód, południowy-zachód, itd.)

Realizacja filtra Prewitta w programach graficznych → grafika 8.40

### Operator Sobela

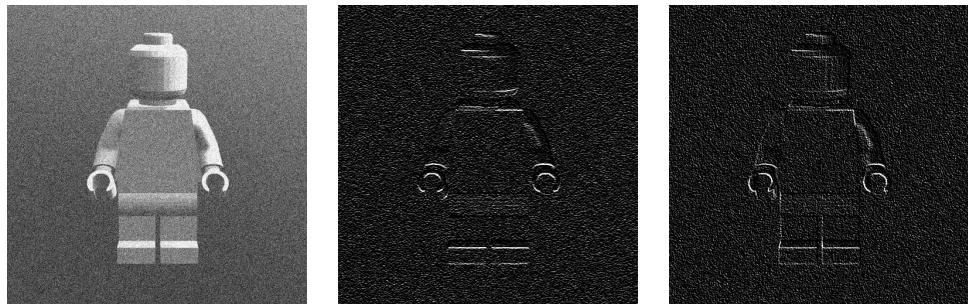
→ filtr gradientowy, który przybliża pochodną poprzez 2 piksele i uśrednia jak filtr ze współczynnikami dwumianowymi.

Istnieje 8 wariantów poprzez obrót o  $45^\circ$ . Przykłady → grafika 8.41

→ Operator Sobela skutkuje "stabilnymi" krawędziami (grafika 8.42).  
Filtr Sobela nazywany jest także **filtrem kompasowym**, ponieważ selektywnie wyodrębnia komponenty kierunkowe z obrazów (8 kierunków odpowiada kierunkom geograficznym, które są wykryte na obrazie wynikowym: wschód, południowy-zachód, itd.)

obraz wejściowy

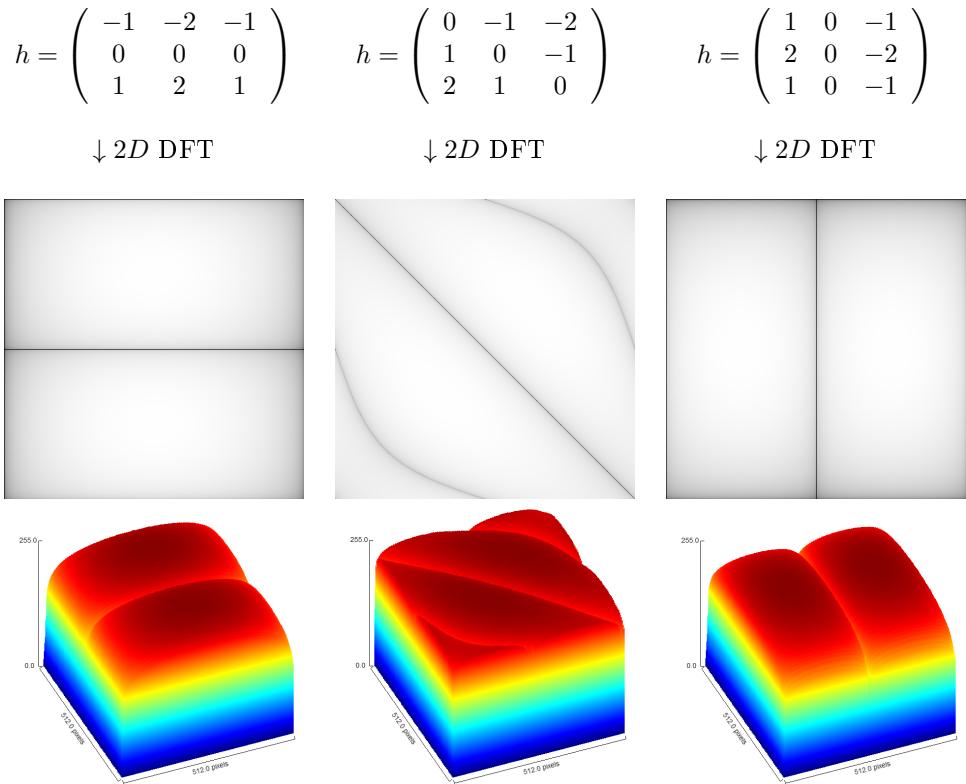
$$h = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad h = \begin{pmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{pmatrix}$$



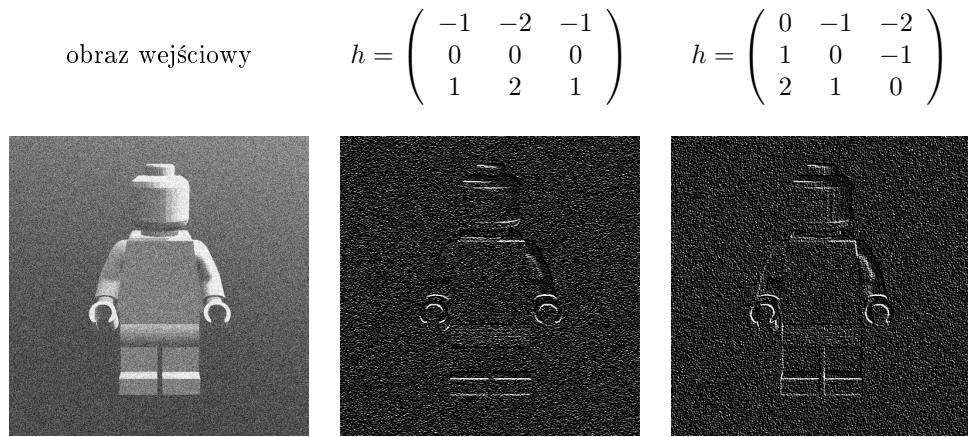
Grafika 8.39: Zaszumiony obraz filtrowany filtrami Prewitta



Grafika 8.40: Filtr Prewitta w oprogramowaniu Blender. Autor: P. Arłukowicz



Grafika 8.41: Amplitudы трансформати Fourierа для фільтрів Sobela



Grafika 8.42: Zaszumiony obraz filtrowany filtrami Sobela

**Zastosowanie: znajdywanie krawędzi w ImageJ ("Find Edges")**

→ grafika 8.43

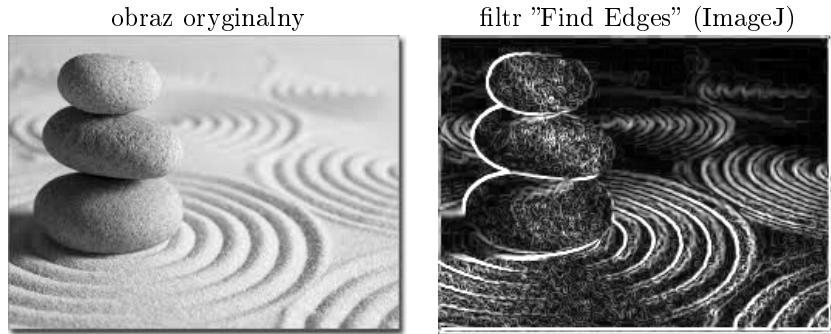
Dwa jądra splotu  $h_1$  i  $h_2$  generują pochodne pionowe i poziome w obrazie wejściowym  $g$ :

$$h_1 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad h_2 = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

Obraz wyjściowy  $f$  obliczany jest ze wzoru

$$f = \sqrt{(g * h_1)^2 + (g * h_2)^2}$$

**Znajdywanie krawędzi w ImageJ ("Find Edges") - problemy**



Grafika 8.43: Filtr "Find Edges" w ImageJ

→ grafiki 8.44, 8.45

Realizacja filtra Sobela w programach graficznych → grafika 8.46

**Filtr kompasowy**

→ filtr gradientowy, który wyodrębnia komponenty kierunkowe z obrazów.

Istnieje 8 wariantów poprzez obrót o  $45^\circ$ . Przykład → grafika 8.47

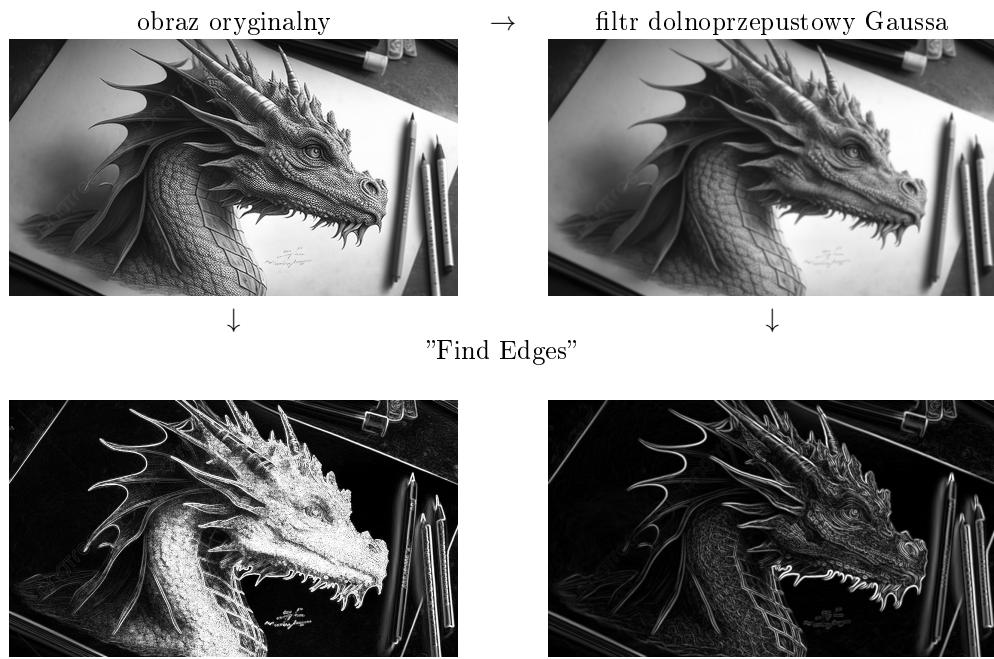
$$h = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{pmatrix}$$

**Operator Kirsch'a**

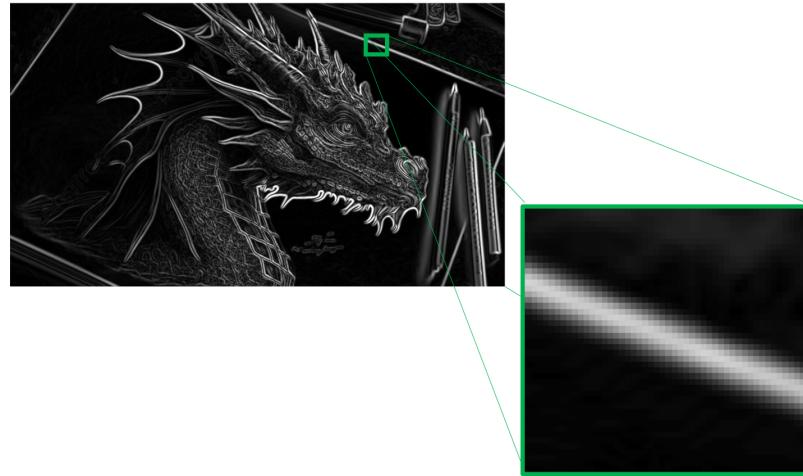
→ filtr gradientowy, który wyodrębnia komponenty kierunkowe z obrazów.

Istnieje 8 wariantów poprzez obrót o  $45^\circ$ . Przykład → grafika 8.47

$$h = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}$$



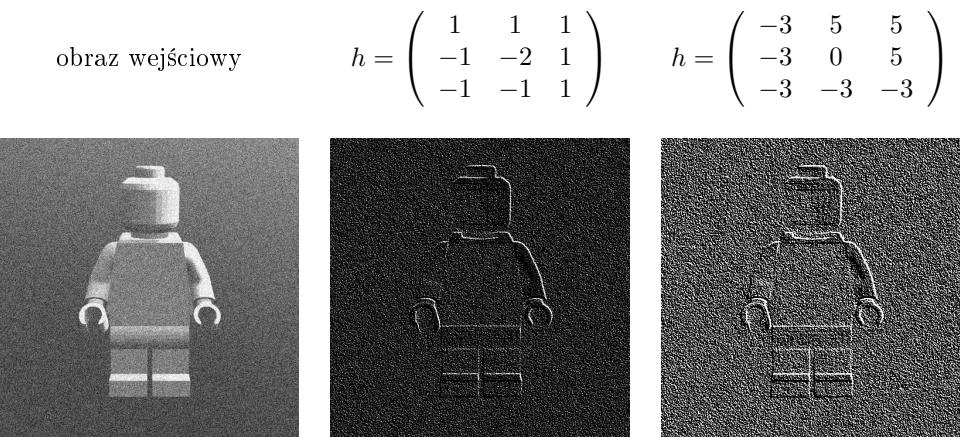
Grafika 8.44: Filtr "Find Edges" w ImageJ dla obrazu bez przetwarzania wstępnie i przetworzonego wstępnie filtrem dolnoprzepustowym Gaussa



Grafika 8.45: Filtr "Find Edges" - grubość krawędzi



Grafika 8.46: Filtr Sobela w oprogramowaniu Blender. Autor: P. Arłukowicz



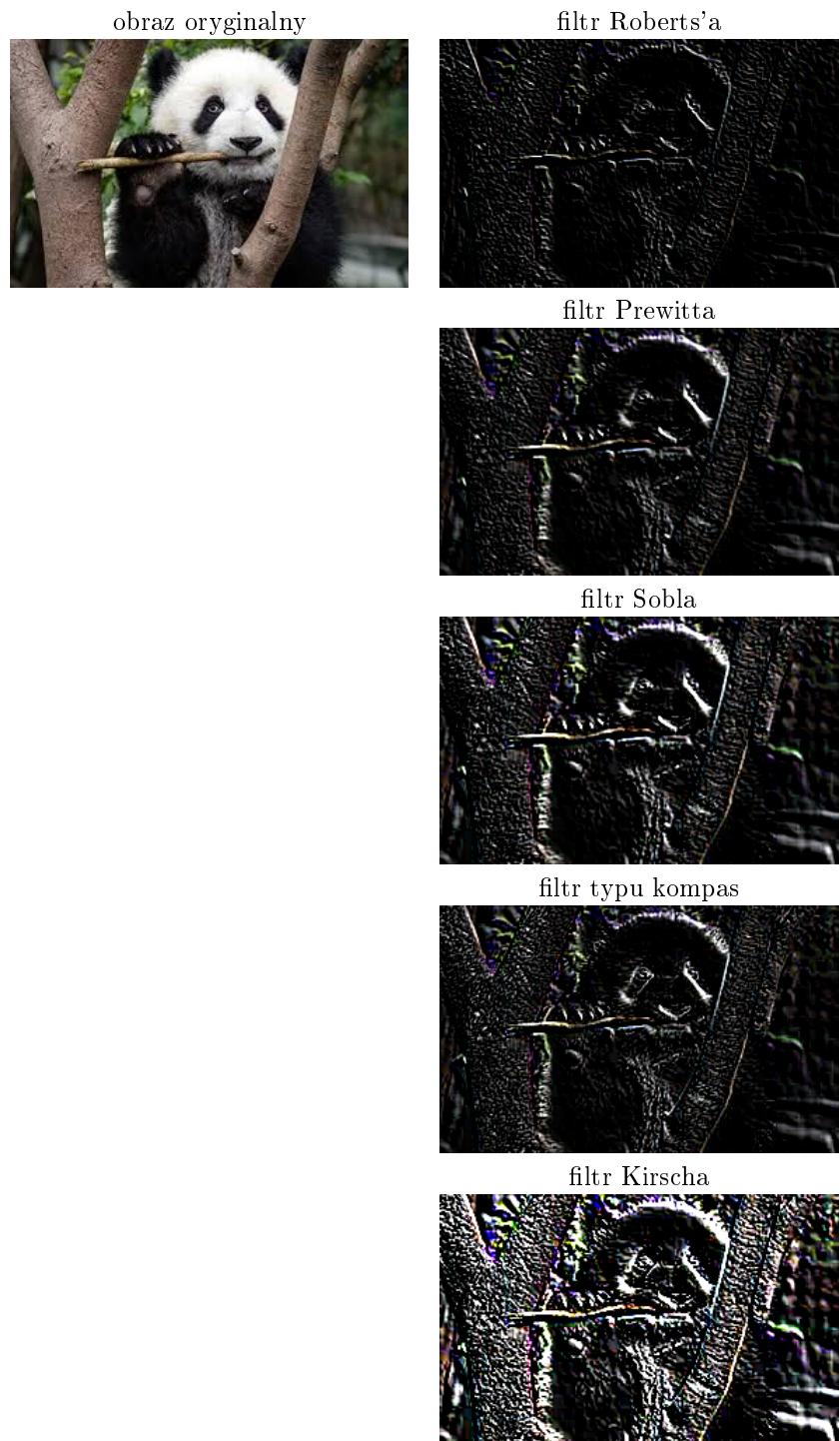
Grafika 8.47: Zaszumiony obraz filtrowany filtrami południowy-zachód typu kompas i Kirsch

Realizacja filtra Kirscha w programach graficznych → grafika 8.48



Grafika 8.48: Filtr Kirscha w oprogramowaniu Blender. Autor: P. Arłukowicz

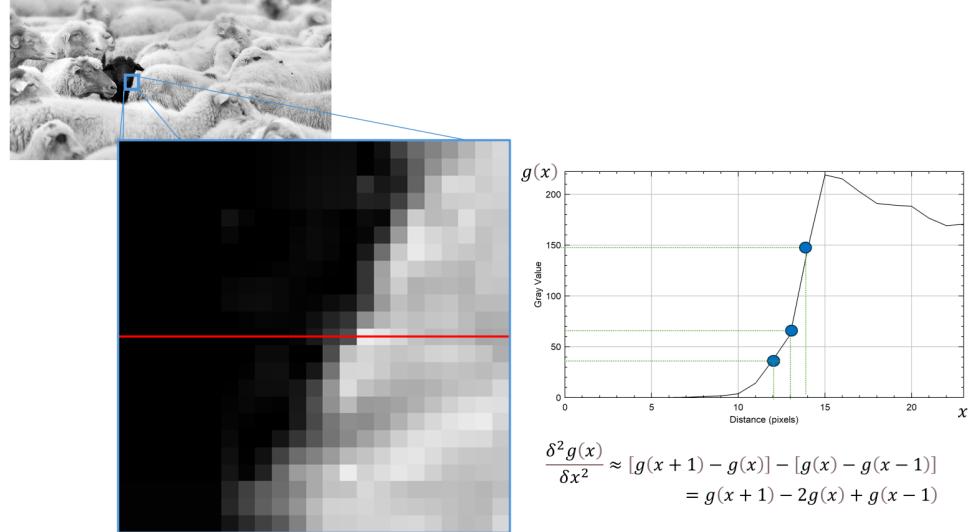
Porównanie filtrów gradientowych → grafika 8.49



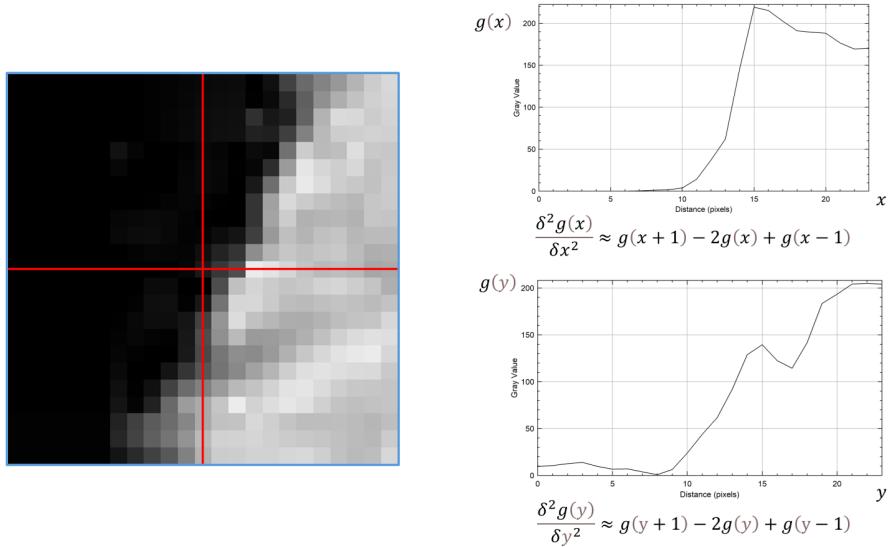
Grafika 8.49: Porównanie gradientowych filtrów południowy-zachód

## Filtryle Laplace'a

Motywacja → grafiki 8.50, 8.51



Grafika 8.50: Profil liniowy (wycinka) obrazu: przybliżenie drugiej pochodnej w kierunku  $x$



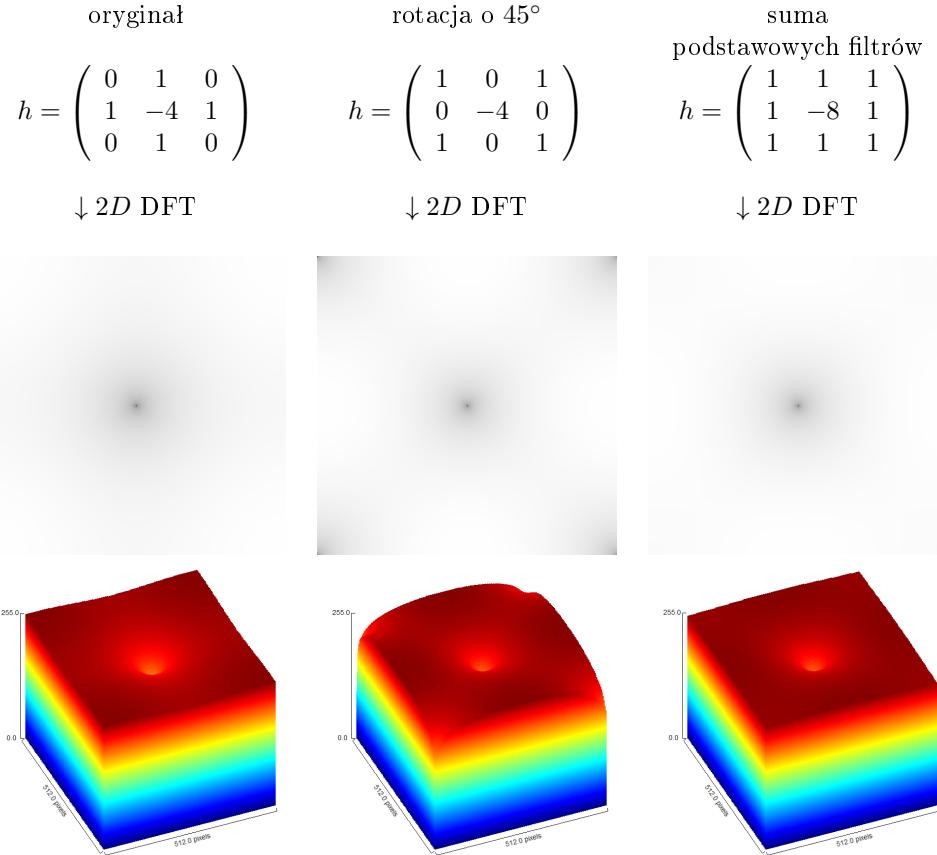
Grafika 8.51: Profile liniowe (wycinki) obrazu: przybliżenie drugiej pochodnej w kierunkach  $x$  i  $y$

→ współczynniki jądra filtra są obliczone z przybliżenia drugiej pochodnej funkcji

obrazu  $g(x, y)$

Istnieją trzy podstawowe warianty filtrów Laplace'a  $\rightarrow$  grafika 8.52

$\rightarrow$  Filtr Laplace'a znajduje punkty o dużej krzywiźnie (grafika 8.53). W odróżnieniu



Grafika 8.52: Filtry Laplace'a i ich transformaty Fouriera

do wymienionych wyżej filtrów filtry Laplace'a cechuje wielokierunkowość (wykrywają krawędzie we wszystkich kierunkach).

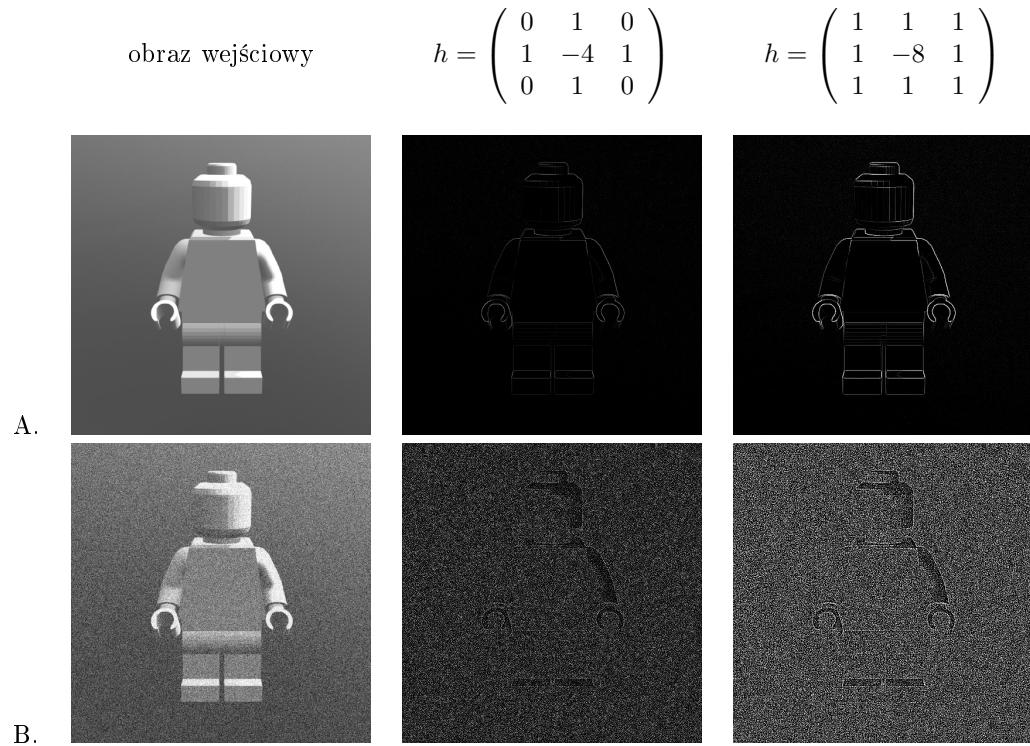
**Przykład: wyostrzanie obrazu za pomocą operatora Laplace'a**

$\rightarrow$  grafika 8.54

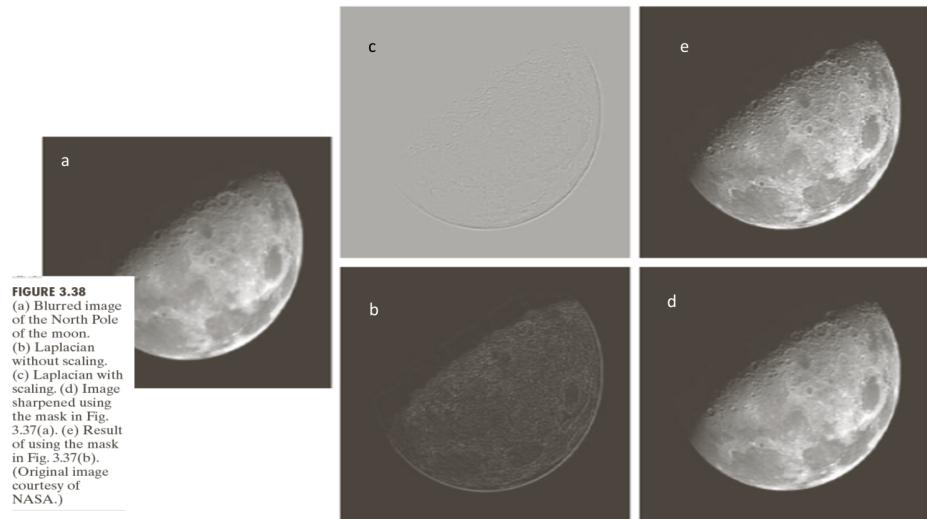
**Przykład: Highboost filtering - maskowanie nieostre**

$\rightarrow$  grafika 8.55

- Wygładzenie oryginalnego obrazu (filtr dolnoprzepustowy)
- Odejmowanie wygładzonego obrazu od oryginału (powoduje powstanie nieostrej maski)



Grafika 8.53: Obraz filtrowany filtrami Laplace'a: A. obraz bez szumu, B. obraz zaszumiony



Grafika 8.54: Wyostrzanie obrazu za pomocą operatora Laplace'a. Zródło: R. C. Gonzalez, R. E. Woods, Digital Image Processing

- Dodanie (skalowanej) maski do obrazu wejściowego

Realizacja filtra Laplace'a w programach graficznych —> grafika 8.56

### Filtry górnoprzepustowe wystrzające obraz

#### 1. filtr usuwający średnią (ang. *mean removal*)

→ współczynniki jądra filtra są obliczone z przybliżenia drugiej pochodnej funkcji obrazu  $g(x, y)$  (analogicznie do filtra Laplace'a), jednak ich średnia wynosi 1, filtr powoduje znaczne wyostrzenie obrazu, ale także wzmacnienie wszelkich szumów i zakłóceń —> grafika 8.57

$$h = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -9 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

#### 2. filtry uwypuklające (*embossing*) —> wprowadzają złudzenie wypukłości i wklęsłości w tych miejscach na obrazie, w których znajdują się krawędzie.

Istnieje 8 wariantów poprzez obrót o  $45^\circ$ . Przykład —> grafika 8.58

$$h = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 1 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

#### 3. filtr uwypuklający "Sharpen" (ImageJ) —> grafika 8.59

$$h = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 12 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

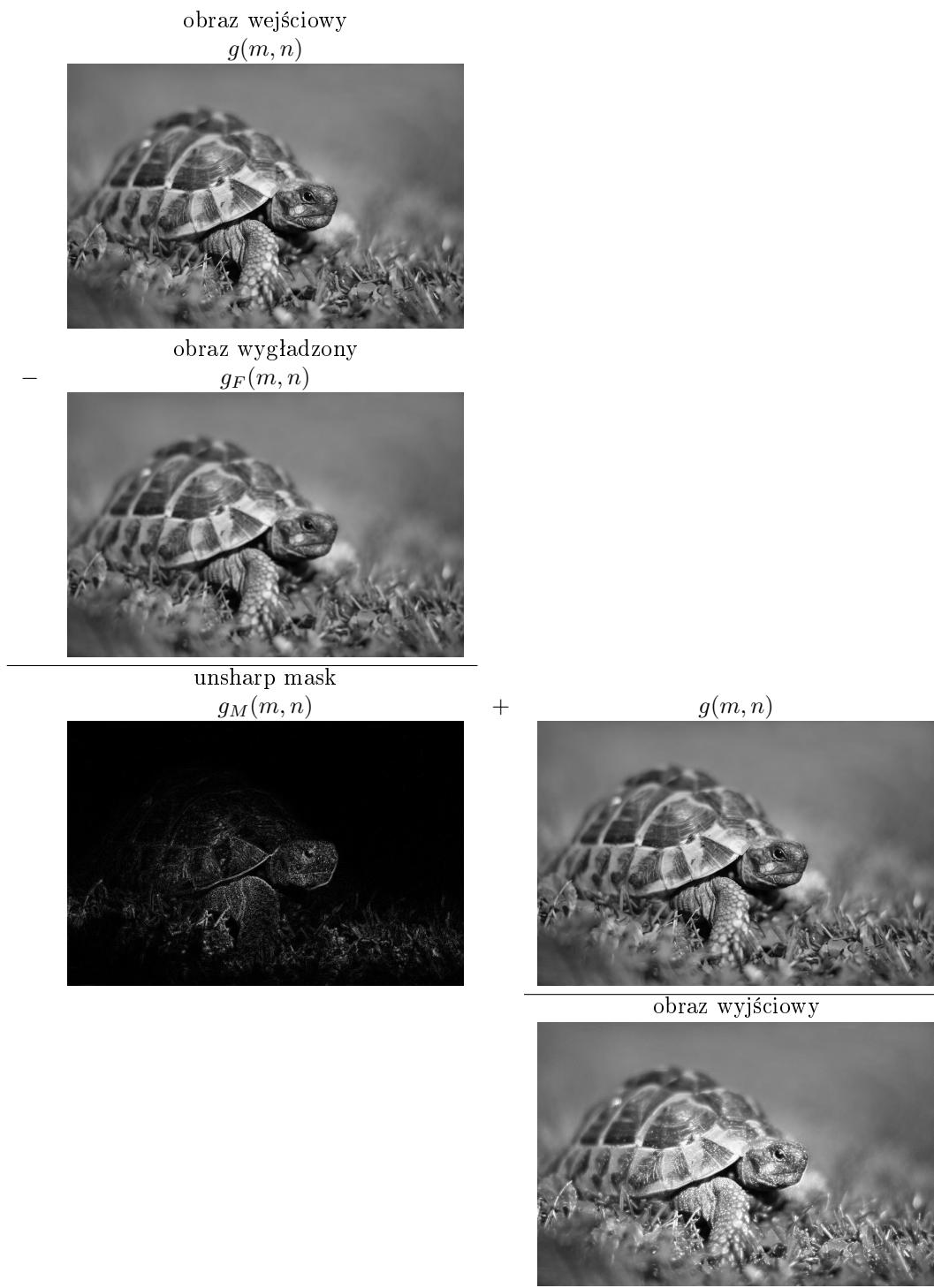
#### 4. efekt cienia "Shadows" (ImageJ)

Efekt cienia światła pochodzącego z kierunku, w którym tworzony jest gradient obrazu.

Istnieje 8 wariantów poprzez obrót o  $45^\circ$ . Przykład —> grafika 8.60

$$h_N = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Realizacja filtrów wyostrzających obraz w programach graficznych —> grafika 8.61

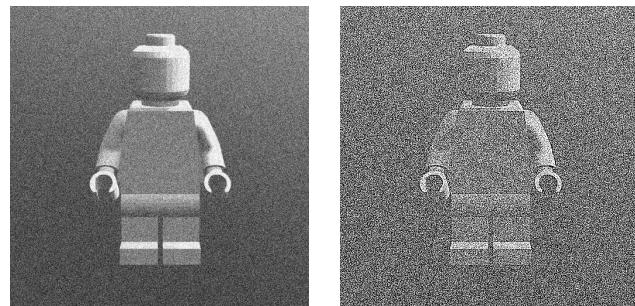


Grafika 8.55: Wyostrzanie obrazu: Highboost filtering



Grafika 8.56: Filtr Laplace'a w oprogramowaniu Blender. Autor: P. Arłukowicz

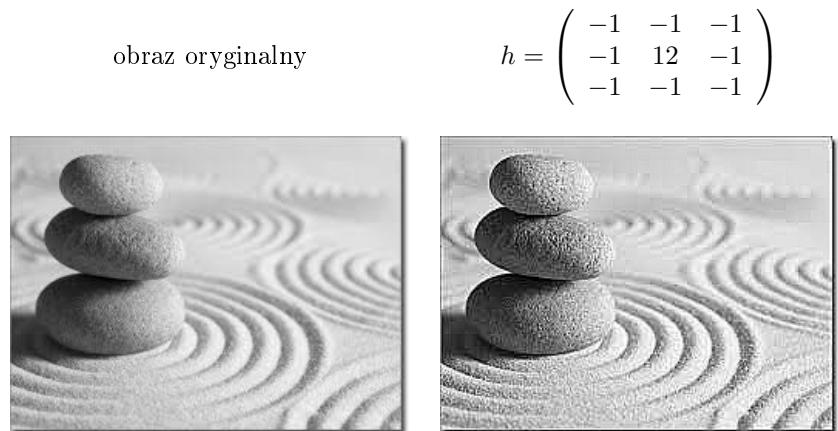
$$\text{obraz wejściowy} \quad h = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -9 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$



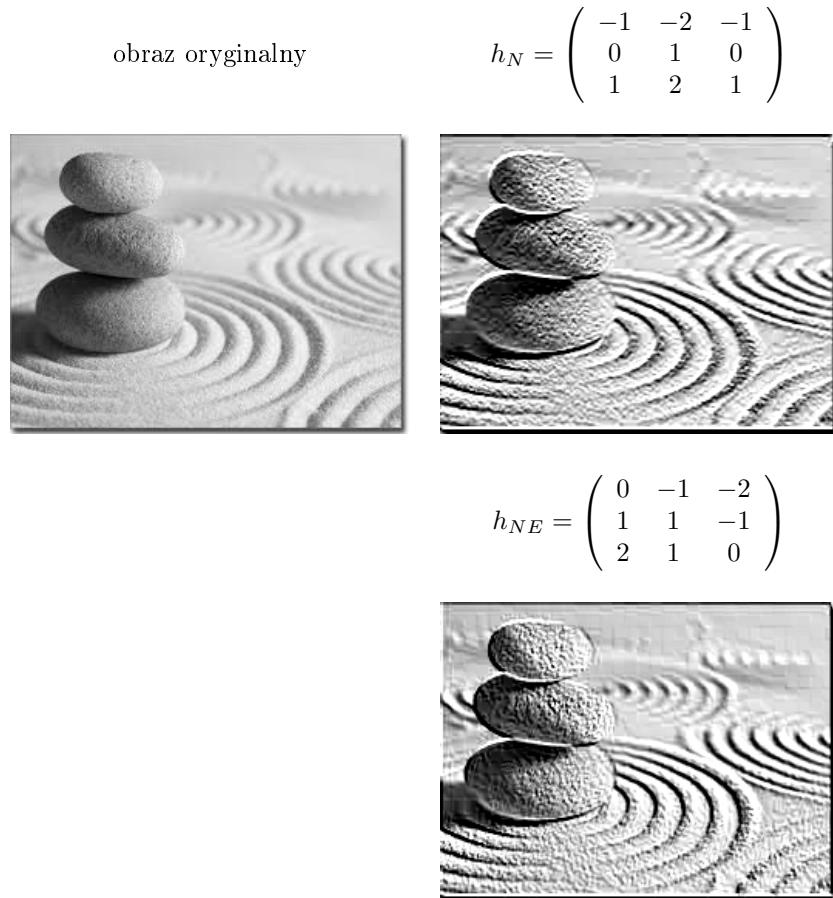
Grafika 8.57: Zaszumiony obraz filtrowany filtrem usuwającym średnią



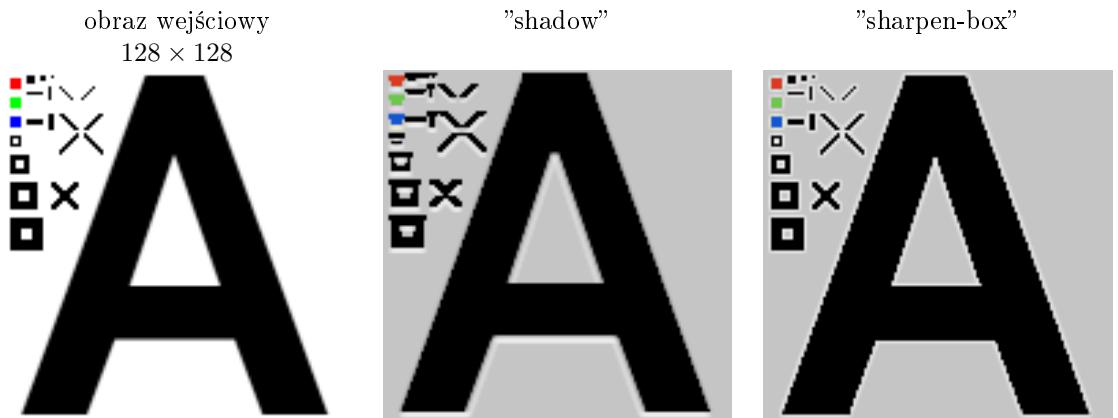
Grafika 8.58: Porównanie filtrów wyostrzających obraz: usuwającego średnią i uwypuklającego



Grafika 8.59: Filtr "Sharpen" w ImageJ



Grafika 8.60: Filtr "Shadows" w ImageJ (North i Northeast)



Grafika 8.61: Filtry wyostrzające obraz w oprogramowaniu Blender. Autor: P. Arłukowicz

### Filtry zdefiniowane w dziedzinie widma

→ grafika 8.21

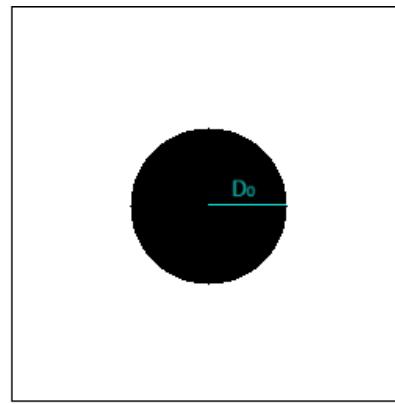
#### Idealny filtr górnoprzepustowy

Definicja (w przypadku filtra symetrycznego rotacyjnie) → grafika 8.62:

$$H(k, l) = \begin{cases} 0 & : D(k, l) \leq D_0 \\ 1 & : D(k, l) > D_0 \end{cases}$$

$$D(k, l) = \sqrt{k^2 + l^2}$$

Przykład: → grafika 8.63



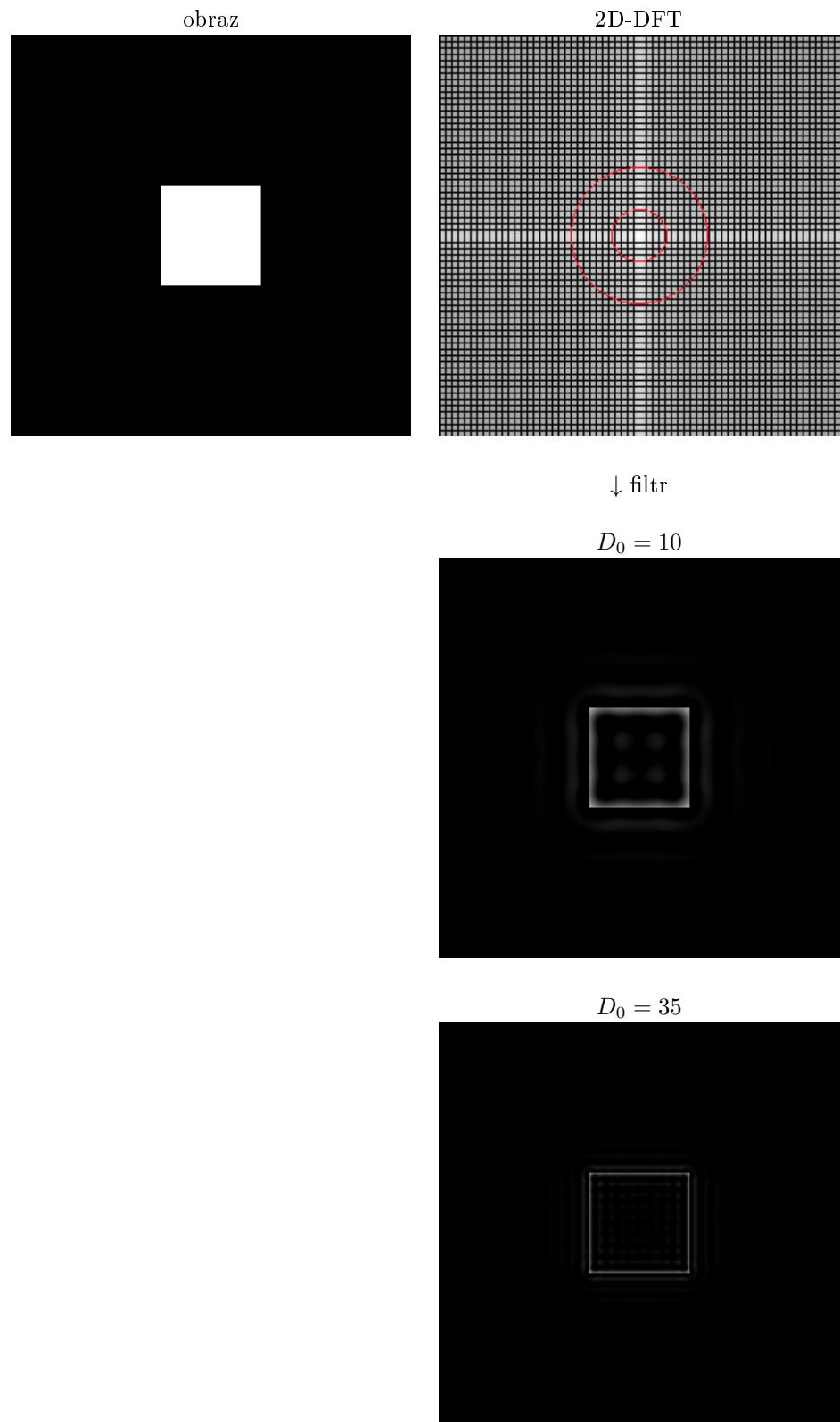
Grafika 8.62: Idealny, symetryczny rotacyjnie filtr górnoprzepustowy

#### Filtr górnoprzepustowy Butterwortha

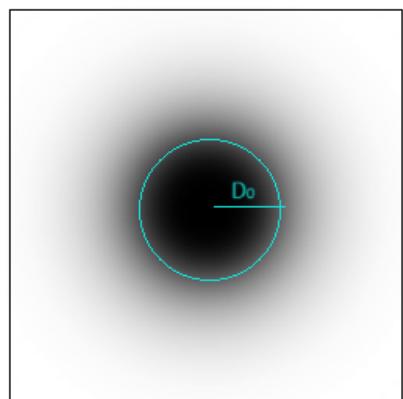
Definicja filtra stopnia  $n$  → grafika 8.64:

$$H(k, l) = \frac{1}{1 + (\sqrt{2} - 1) \left[ \frac{D_0}{D(k, l)} \right]^{2n}}, \quad n \in \mathbb{N}$$

$$D(k, l) = \sqrt{k^2 + l^2}$$



Grafika 8.63: Obraz filtrowany idealnym filtrem górnoprzepustowym o promieniu  $D_0$



Grafika 8.64: Filtr górnoprzepustowy Butterwortha

## 8.5 Dekonwolucja

$$\begin{array}{ccc}
 g(m, n) & * & h(m, n) \\
 \text{?} & & \\
 & & h = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 12 & -1 \\ -1 & -1 & -1 \end{pmatrix} \\
 & & \text{Eiffel Tower image}
 \end{array}$$

Grafika 8.65: Problem dekonwolucji

Dekonwolucja (rozplot, *deconvolution*) to odwrócenie operacji splotu ( $\rightarrow$  grafika 8.65).

Splot można zawsze obliczyć, natomiast jego odwrócenie nie zawsze jest możliwe, ponieważ splot może spowodować utratę informacji, których nie będzie można odzyskać. Aby nadal móc obliczyć możliwie najlepsze operacje dekonwolucji, opracowano algorytmy rozplotu.

**Dane:**

$g'(m, n)$ : obraz wyjściowy

$h(m, n)$ : jądro splotu

$$g'(m, n) = g(m, n) * h(m, n)$$

**Szukane:**

$g(m, n)$ : obraz wejściowy

**Rozwiążanie w teorii:**  $\rightarrow$  grafika 8.66

$g'(m, n)$ : obraz wyjściowy  $\rightarrow$  2D-DFT  $G(k, l)$

$h(m, n)$ : jądro splotu  $\rightarrow$  2D-DFT  $H(k, l)$

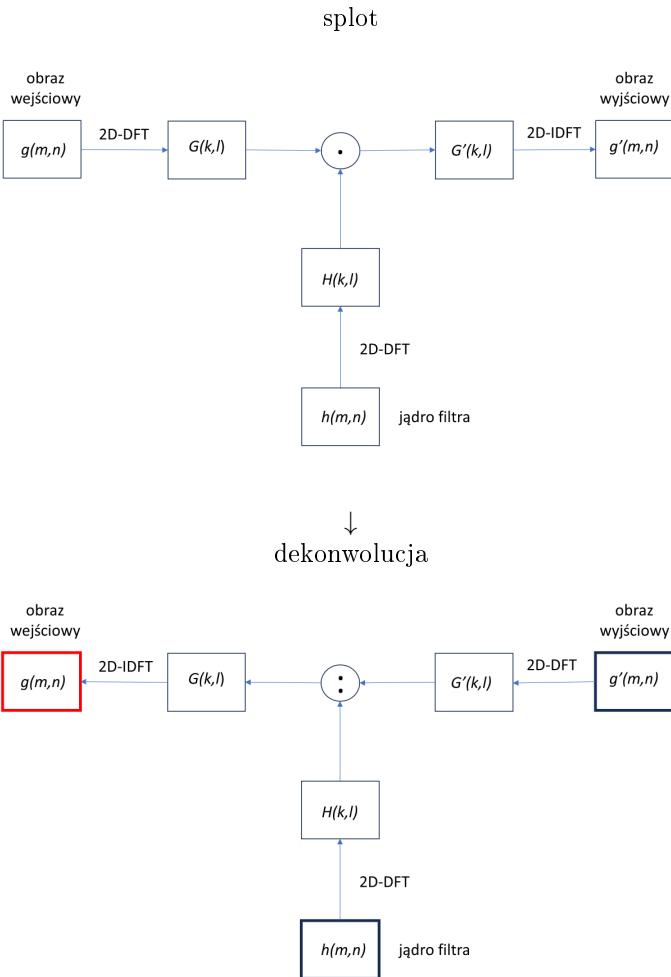
$$\begin{aligned}
 g'(m, n) &= g(m, n) * h(m, n) \\
 \rightarrow G'(k, l) &= G(k, l) \cdot H(k, l)
 \end{aligned}$$

$$G(k, l) = \frac{G'(k, l)}{H(k, l)}$$

$g(m, n)$ : obraz wejściowy  $\leftarrow$  2D-IDFT  $G(k, l)$

To ogólne podejście zwykle nie ma zastosowania, ponieważ

- $g$  nie musi być jednoznaczny,
- $H$  może posiadać miejsca zerowe,
- na rzeczywiste dane zwykle wpływa szum addytywny.



Grafika 8.66: Splot i dekonwolucja (w teorii)

Ponieważ nie istnieje jednoznaczne rozwiązańe analityczne, stosuje się różne metody, które próbują określić najbardziej prawdopodobny wynik dla  $g$ .

### Jądro filtra $h(m, n)$

- odpowiedź punktowa systemu (*point spread function, PSF*)  
→ mikroskopy, astrofizyka
- (znane) jądro zastosowane do przetwarzania obrazu wejściowego

#### 8.5.1 Odpowiedź punktowa systemu (*point spread function, PSF*)



### Odpowiedź punktowa w teorii

Dla obrazu wejściowego  $g(m, n) = \delta(m, n)$

$$\delta(m, n) = \begin{cases} 1 & : m, n = 0 \\ 0 & : m \neq 0 \text{ lub } n \neq 0 \end{cases}$$

obraz wyjściowy  $g'(m, n)$  wyznaczony ze splotu z jądrem  $h(m, n)$  wynosi

$$g'(m, n) = g(m, n) * h(m, n) = \delta(m, n) * h(m, n) = h(m, n)$$

→ grafika 8.67

$h(m, n)$  jest odpowiedzią systemu na sygnał  $\delta(m, n)$  → **odpowiedź punktowa  $h_{PSF}$**

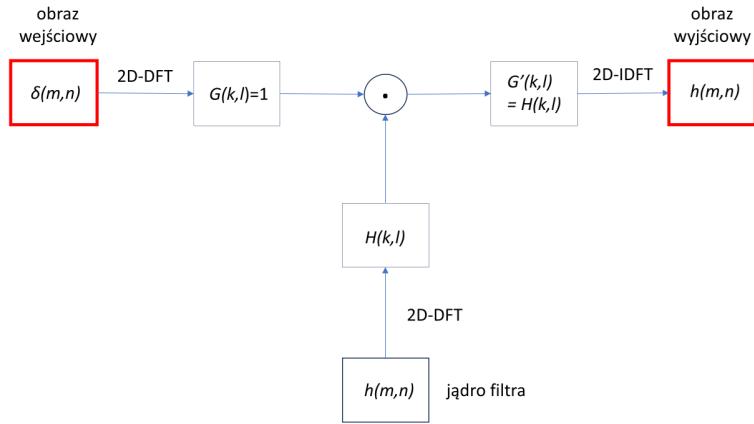
### Odpowiedź punktowa w praktyce

Odpowiedź punktowa (PSF) w optyce i przetwarzaniu obrazu opisuje wpływ czynników takich jak

- zjawiska dyfrakcyjne
- błędy odwzorowania
- wpływ obszaru czujnika lub przysłony

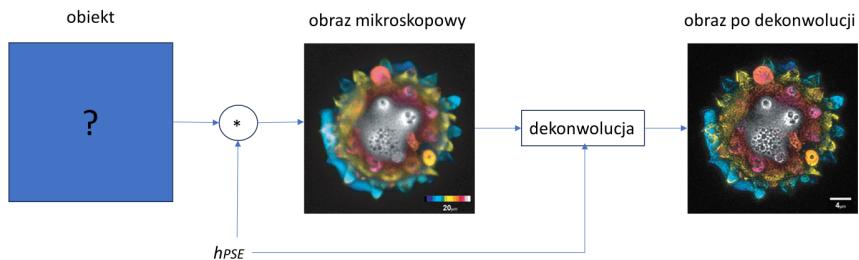
PSE opisuje, jak wyidealizowany obiekt punktowy byłby reprezentowany przez system. W mikroskopach szerokość PSF ogranicza rozdzielcość, jaką można osiągnąć.

- każdy obraz to splot rzeczywistych źródeł światła z odpowiedzią punktową  $h_{PSF}$  systemu akwizycji obrazu



Grafika 8.67: Odpowiedź punktowa

- z reguły przyjmuje się  $h_{PSF} = \delta$  (idealny system)
- korekta konieczna w wysokiej jakości obrazach mikroskopowych i astrofizyce  
→ grafika 8.68



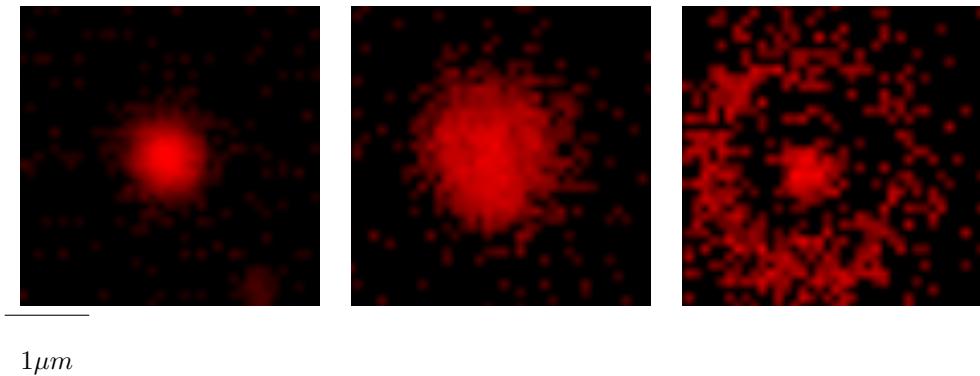
Grafika 8.68: Dekonwolucja obrazu mikroskopowego (obraz pyłku).

#### Pomiar eksperymentalny:

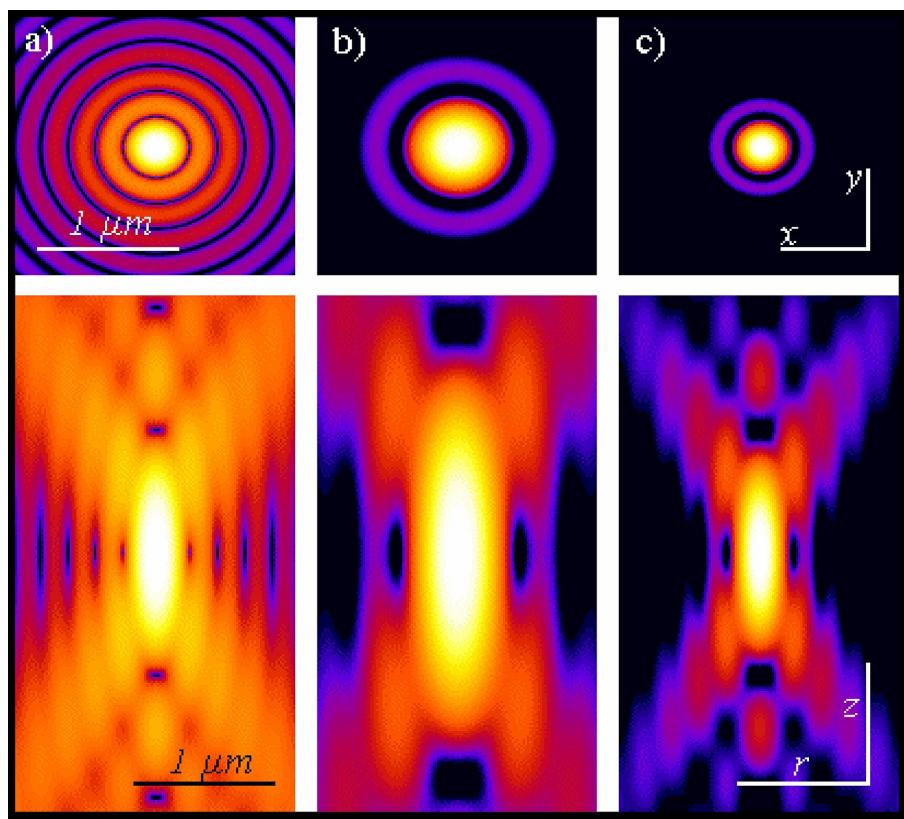
Zwykle rejestruje się obraz mikroskopowy małej kulistej struktury, na przykład obraz mikroskopowy bardzo małych fluorescencyjnych kulek lateksowych. → grafika 8.69

#### Modelowanie matematyczne:

Biomedical Imaging Group, École Polytechnique fédérale de Lausanne:  
<https://bigwww.epfl.ch/algorithms/psfgenerator/> → grafika 8.70



Grafika 8.69: Odpowiedź punktowa  $h_{PSE}$  mikroskopu wyznaczona eksperymentalnie: obraz kulki fluorescencyjnej o średnicy  $0.1\mu m$  (w płaszczyźnie ogniskowej)



Grafika 8.70: Porównanie obliczonych odpowiedzi punktowych  $h_{PSE}$  w a) normalnej (szerokim polu) mikroskopii fluorescencyjnej, b) mikroskopii dwufotonowej i c) mikroskopii konfokalnej, u góry w płaszczyźnie ogniskowej i u dołu wzduż osi optycznej.  
 Źródło: PSF Generator EPFL

### 8.5.2 Algorytmy dekonwolucji

Dekonwolucję wykorzystuje się na przykład do wyostrzania obrazów w astrofotografii i mikroskopii. Filtry dekonwolucyjne próbują matematycznie opisać i cofnąć rozmycie obrazu.

Algorytmy dokonwolucji

- algorytm dekonwolucji Van-Citterta
- dekonwolucja Wienera
- blind deconvolution
- ...

#### Algorytm dekonwolucji Van-Citterta

**Dane:**

$g'(m, n)$ : obraz wyjściowy  $\longrightarrow$  2D-DFT  $G(k, l)$

$h(m, n)$ : jądro splotu  $\longrightarrow$  2D-DFT  $H(k, l)$

(np.  $h_{PSE}(m, n)$ : odpowiedź punktowa systemu  $\longrightarrow$  2D-DFT  $H_{PSE}(k, l)$ )

**Szukane:**

$g(m, n)$ : obraz wejściowy  $\longleftarrow$  2D-IDFT  $G(k, l)$

$$\begin{array}{rcl} g' & = & g * h \\ \rightarrow & G' & = G \cdot H \end{array}$$

$$G = \frac{G'}{H}$$

By uniemożliwić ewentualne dzielenie przez 0 wprowadza się funkcję  $\tilde{H}(k, l) = 1(k, l) - H(k, l)$  (rozwnięcie w szereg Taylora):

$$G = \frac{G'}{H} = \frac{G'}{1 - \tilde{H}} \approx \left(1 + \tilde{H} + \tilde{H}^2 + \tilde{H}^3 + \dots\right) \cdot G'$$

Transformacja odwrotna:

$$g \approx \left(\delta + \tilde{h} + \tilde{h} * \tilde{h} + \tilde{h} * \tilde{h} * \tilde{h} + \dots\right) * g'$$

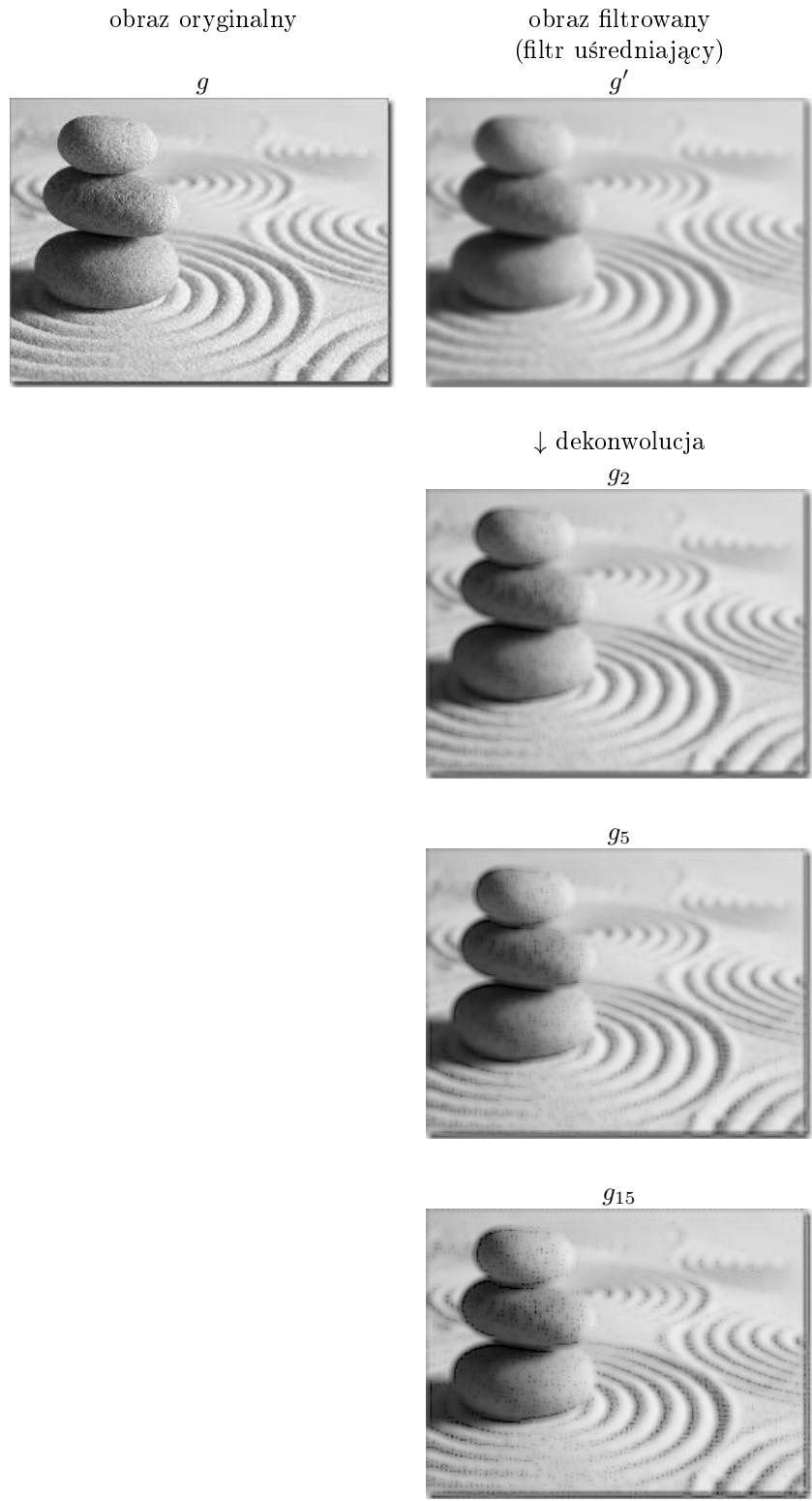
gdzie  $\tilde{h}(m, n) = \delta(m, n) - h(m, n)$  to transformacja odwrotna  $\tilde{H}(k, l)$ .

Po zastosowaniu schematu Hornera otrzymuje się wzór rekurencyjny do obliczenia obrazu wejściowego  $g$ .

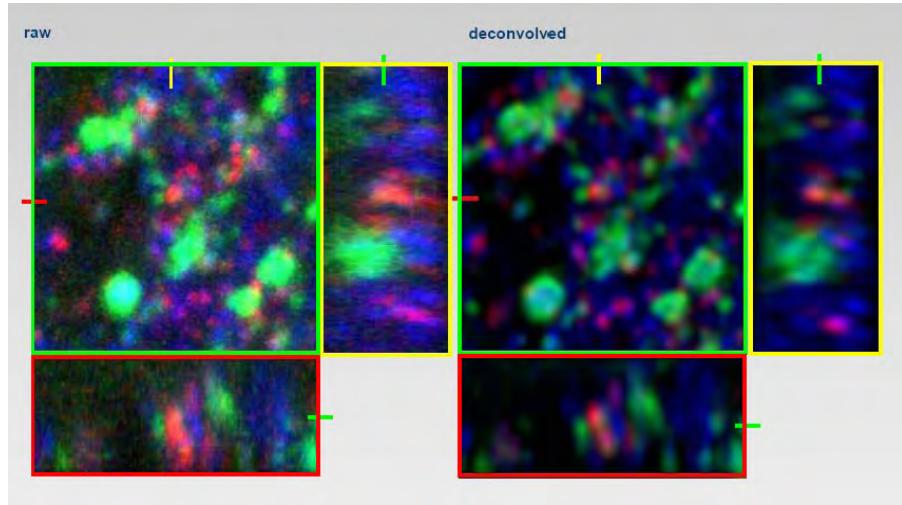
#### Dekonwolucja Van-Citterta

$$\begin{array}{rcl} g_0 & = & g' \\ g_{k+1} & = & g' + (\delta - h) * g_k \end{array}$$

Przykłady: grafiki 8.71, 8.72



Grafika 8.71: Algorytm dekonwolucji Van-Citterta



Grafika 8.72: Mikroskop z odpowiedzią punktową  $h_{PSE}$ : dekonwolucja obrazu

### Dekonwolucja Wienera

#### Dane:

$g'(m, n)$ : obraz wyjściowy  $\rightarrow$  2D-DFT  $G(k, l)$

$h(m, n)$ : jądro splotu  $\rightarrow$  2D-DFT  $H(k, l)$

(np.  $h_{PSE}(m, n)$ : odpowiedź punktowa systemu  $\rightarrow$  2D-DFT  $H_{PSE}(k, l)$ )

$n(m, n)$ : szum (niezależny od obrazu wejściowego)  $\rightarrow$  widmowa gęstość szumu  $N$   $\rightarrow$  widmowa gęstość typowego obrazu  $S$

#### Szukane:

$g(m, n)$ : obraz wejściowy  $\leftarrow$  2D-IDFT  $G(k, l)$

$$g' = g * h + n$$

#### Cel:

znalenienie  $x(m, n)$ , takiego że

$$\tilde{g} = x * g'$$

gdzie  $\tilde{g}$  jest oszacowaniem  $g$  ze zminimalizowanym błędem kwadratowym.

**Filtr Wienera** (definicja w przestrzeni Fouriera):

$$X = \frac{H^* S}{|H|^2 S + N}$$

Powyższe równanie zakłada, że znane są wartości widmowe typowego obrazu oraz szumu.