

Cyfrowe Przetwarzanie Obrazu

Filtry splotowe

Wojciech Kowalewski

fraktal@amu.edu.pl

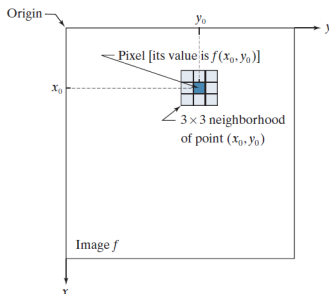


Wydział Matematyki i Informatyki UAM
Uniwersytet im. Adama Mickiewicza w Poznaniu

Wiosna 2021

Operacje w dziedzinie przestrzennej obrazu

- **Przypomnienie:** Dość ogólną grupę operacji w dziedzinie przestrzennej można zdefiniować następująco:
 - niech $f(x, y)$ oznacza obraz źródłowy, natomiast $g(x, y)$ obraz wynikowy
 - przekształcenie $g(x, y) = T(f(x, y))$ działa na pewnym *sąsiedztwie* piksela (x, y) obrazu $f(x, y)$. Pokazuje to poniższy rysunek dla piksela (x_0, y_0) i jego otoczenia wymiaru 3 (otoczenia są zwykle kwadratami o nieparzystej długości krawędzi - w ogólności otoczenia mogą być prostokątami):



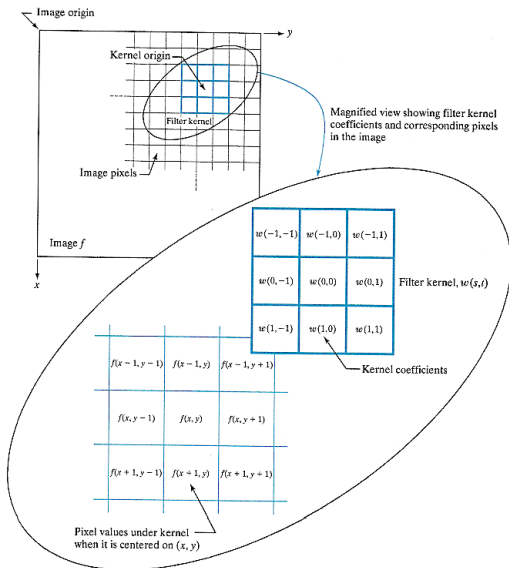
- Przykładem tak rozumianego przekształcenia jest *operacja splotu* i bardzo do niej podobna *operacja korelacji krzyżowej*.

Przykład ideowy

- ▶ W każdym pikselu (x, y) obrazu $f(x, y)$ zaczepiamy centralnie jądro filtra $w(s, t)$
- ▶ Następnie w każdym pikselu obrazu, który jest pokryty przez jądro filtra wykonujemy mnożenie wartości filtra i wartości obrazu.
- ▶ Finalnie sumujemy otrzymane iloczyny i zastępujemy wartość $f(x, y)$ obliczoną sumą.

$$T(f(x, y)) = \sum_{m=-1}^1 \sum_{n=-1}^1 f(x+m, y+n) \cdot w(m, n)$$

- ▶ Takie przekształcenie jest źródłem bardzo wielu klasycznych filtrów obrazowych i jest w istocie *operacją korelacji krzyżowej*

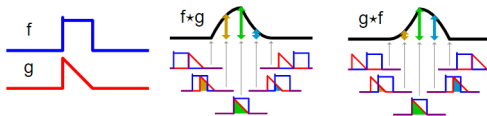


Korelacja krzyżowa

- Załóżmy, że f , g są ciągłymi rzeczywistymi funkcjami jednej zmiennej. *Korelacją krzyżową* funkcji f i g nazywamy funkcję zdefiniowaną przez formułę

$$(f \star g)(s) = \int_{-\infty}^{\infty} f(t)g(s+t) dt \quad (1)$$

- Odpowiednikiem *jądra* z poprzedniego slajdu jest tu funkcja f
- Operacja korelacji krzyżowej nie jest przemienalna - pokazuje to poniższy rysunek (na podstawie Wikipedii)



- *Korelacja krzyżowa* funkcji f i g może być równoważnie zdefiniowana jako

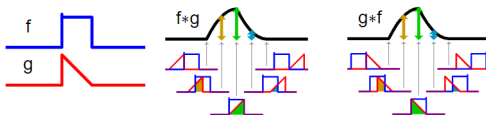
$$(f \star g)(s) = \int_{-\infty}^{\infty} f(t-s)g(t) dt \quad (2)$$

Splot funkcji

- Załóżmy, że f , g są ciągłymi rzeczywistymi funkcjami jednej zmiennej. *Splotem* funkcji f i g nazywamy funkcję zdefiniowaną przez formułę

$$(f * g)(s) = \int_{-\infty}^{\infty} f(t)g(s - t) dt \quad (3)$$

- Operacja splotu jest przemienalna - pokazuje to poniższy rysunek (na podstawie Wikipedii) - formalny dowód przemienności jest oczywisty (zamiana zmiennych pod całką)



- *Splot* funkcji f i g może być zatem równoważnie zdefiniowany jako

$$(f * g)(s) = \int_{-\infty}^{\infty} f(s - t)g(t) dt \quad (4)$$

Własności korelacji krzyżowej i splotu

► Łączność splotu

$$f * (g * h) = (f * g) * h \quad (5)$$

► Rozdzielność splotu względem dodawania

$$f * (g + h) = (f * g) + (f * h) \quad (6)$$

► *Relacja splotu i korelacji krzyżowej* - spłot jest równoważny korelacji krzyżowej, jeżeli wykres jednej z funkcji w splocie odbijemy względem osi Y:

$$(f(t) \star g(t))(t) = (f(-t) * g(t))(t) \quad (7)$$

► *Splot korelacji i korelacja splotu*

$$f \star (g * h) = (f \star g) * h \quad (8)$$

► *Pewna przewaga splotu nad korelacją krzyżową* - spłot w przeciwieństwie do korelacji krzyżowej jest *operacją przemenną*

Korelacji krzyżowa i spłot funkcji dyskretnej

- Załóżmy, że f , g są wektorami rzeczywistymi. Wówczas *Korelacja krzyżowa* ma postać formuły

$$(f \star g)[n] = \sum_{m=-\infty}^{\infty} f[m] \cdot g[n + m] \quad (9)$$

natomiast *splot* jest definiowany przez formułę

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m] \cdot g[n - m] \quad (10)$$

- W wersji dwuwymiarowej wzory te mają postać:

$$(f \star g)[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] \cdot g[m + k, n + l] \quad (11)$$

oraz

$$(f * g)[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] \cdot g[m - k, n - l] \quad (12)$$

- ▶ Przemienność spłotu powoduje, że jest o częściej używany jako ogólna koncepcja filtracji sygnałów (jedno i wielowymiarowych)
- ▶ W praktyce oczywiście nośniki zarówno sygnałów jak i jąder filtrów są skończone, zatem wzory (9) oraz (12) przyjmują postaci

$$(f * g)[n] = \sum_{m=-M_1}^{M_2} f[m] \cdot g[n - m], \quad n = N_1, \dots, N_2 \quad (13)$$

oraz

$$(f * g)[n, m] = \sum_{k=-K_1}^{K_2} \sum_{l=-L_1}^{L_2} f[k, l] \cdot g[m - k, n - l], \quad m = M_1, \dots, M_2, \quad n = N_1, \dots, N_2 \quad (14)$$

- ▶ W przypadku obrazu $f[x, y]$ o wymiarach $M \times N$ oraz kwadratowego jądra filtru w o nieparzystym wymiarze S otrzymujemy ($s = S/2 + 1$)

$$(w * f)[n, m] = \sum_{k=-s}^s \sum_{l=-s}^s w[k, l] \cdot f[m - k, n - l], \quad m = 0, \dots, M-1, \quad n = 0, \dots, N-1 \quad (15)$$

- ▶ Wzór (7) pokazuje, że jeżeli $w[x]$ jest jądrem korelacji krzyżowej, to odpowiadające mu jądro splotu będzie obrócone o 180 stopni względem swojego środka (odbicie w pionie)
- ▶ Analogicznie jest dla przypadku dwuwymiarowego: jeżeli $w[x, y]$ jest jądrem korelacji krzyżowej, to odpowiadające mu jądro splotu będzie obrócone o 180 stopni względem swego środka (odbicie w pionie i następnie odbicie w poziomie).
- ▶ Ponadto jeżeli jądro ma nieparzysty wymiar K , to zaczepienie go w pozycjach $0, \dots, K/2$ oraz $N - K/2 - 1, \dots, N - 1$ filtrowanego sygnału jednowymiarowego powoduje brak $K/2$ wartości w sygnale z lewej i prawej strony. Analogicznie jest w przypadku sygnału dwuwymiarowego.
- ▶ Powoduje to potrzebę *rozszerzania sygnału o $K/2$* (tzw. padding) na końcach dziedziny w każdym jej kierunku.
- ▶ Kolejne dwa slajdy pokazują to w przypadku jedno i dwuwymiarowym, zarówno dla korelacji krzyżowej jak i splotu.

Przykład filtracji jednowymiarowej

Correlation

Origin f w
 0 0 0 1 0 0 0 0 1 2 4 2 8

↓
 0 0 0 1 0 0 0 0
 1 2 4 2 8

↑ Starting position alignment

Zero padding
 0 0 0 0 0 1 0 0 0 0 0 0
 1 2 4 2 8

↑ Starting position

0 0 0 0 0 1 0 0 0 0 0 0
 1 2 4 2 8

↑ Position after 1 shift

0 0 0 0 0 1 0 0 0 0 0 0
 1 2 4 2 8

↑ Position after 3 shifts

0 0 0 0 0 1 0 0 0 0 0 0
 1 2 4 2 8

Final position

Correlation result

0 8 2 4 2 1 0 0

Extended (full) correlation result

0 0 0 8 2 4 2 1 0 0 0 0

Convolution

Origin f w rotated 180°
 0 0 0 1 0 0 0 0 8 2 4 2 1

0 0 0 1 0 0 0 0
 8 2 4 2 1

↑ Starting position alignment

Zero padding
 0 0 0 0 0 1 0 0 0 0 0 0
 8 2 4 2 1

↑ Starting position

0 0 0 0 0 1 0 0 0 0 0 0
 8 2 4 2 1

↑ Position after 1 shift

0 0 0 0 0 1 0 0 0 0 0 0
 8 2 4 2 1

↑ Position after 3 shifts

0 0 0 0 0 1 0 0 0 0 0 0
 8 2 4 2 1

Final position

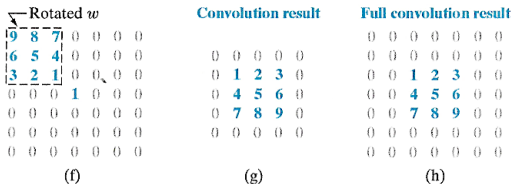
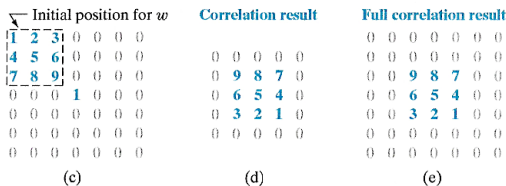
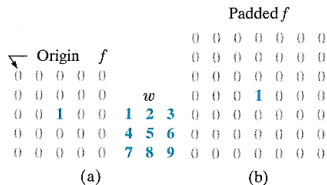
Convolution result

0 1 2 4 2 8 0 0

Extended (full) convolution result

0 0 0 1 2 4 2 8 0 0 0 0

Przykład filtracji dwuwymiarowej



- Rozpatrzmy filtr z jądrem o rozmiarze 3 i postaci:

1	1	1
1	1	1
1	1	1

- Jasne jest, że takie jądro łatwo może spowodować przekroczenie zakresu jasności kanału w pikselu: jeżeli zastosujemy je do fragmentu obrazu o wartościach

255	255	255
255	255	255
255	255	255

otrzymamy wartość $9 \cdot 255 = 2295$, co znacznie przekracza zakres $[0, 255]$ i spowoduje automatyczne obcięcie do wartości **255**.

- W celu uniknięcia takich obcięć można filtr poddać *normalizacji*, która w tym przypadku może być najprościej wykonana przez podzielenie każdej wartości filtra przez sumę jego wszystkich wartości, tzn. w tym przypadku przez 9:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

► Filtry

-1	-1	-1
-1	8	-1
-1	-1	-1

-1	0	1
-1	0	1
-1	0	1

nie wymagają tak rozumianej normalizacji, ale łatwo mogą doprowadzić do przekroczenia zakresu wartości piksela poniżej wartości zero.

► Przykładowo prawy z nich zastosowany do fragmentu obrazu o wartościach

255	255	0
255	255	0
255	255	0

da wartość -765 , co spowoduje automatyczne obcięcie do zera. Z drugiej strony filtr taki może wygenerować również wartość $+765$, co sugeruje co najmniej jego przeskalowanie przez 3.

► Dodatkowo można w takich sytuacjach dodać (poza operacją splotu) do działania filtra stałą wartość. Przykładowo prawy filtr można by zmodyfikować do postaci

$-1/6$	0	$1/6$
$-1/6$	0	$1/6$
$-1/6$	0	$1/6$

co implikuje wynik jego działania z zakresu $[-127, 127]$ i następnie dodać do wartości piksela wartość $+127$. Ostatecznie wartości będą z zakresu $[0, 254]$.

Filtry wygładzające (blur)

► Wygładzanie liniowe i jego modyfikacje

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

1/12	1/12	1/12
1/12	4/12	1/12
1/12	1/12	1/12

Lewy i środkowy filtr wygładzają jednostajnie (środkowy znacznie mocniej), natomiast prawy filtr preferuje piksel centralny w średniej (wariant nieliniowy)

► Rozmycie nieliniowe Gaussa - generatorem jest funkcja ([link do wykresu](#))

$$f(x, y) = \frac{1}{\sqrt{2\pi}} e^{-\left(\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right)}$$

Próbkowanie jej symetrycznej ($\mu_x = \mu_y = 0$, $\sigma_x = \sigma_y = 1$) postaci na siatce 5x5 daje filtr

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

W tym przypadku piksele są uwzględniane w średniej proporcjonalnie do ich odległości od piksela centralnego.

Kierunkowe filtry wygładzające (blur)

0	0	0	0	0
0	0	0	0	0
1/5	1/5	1/5	1/5	1/5
0	0	0	0	0
0	0	0	0	0

0	0	1/5	0	0
0	0	1/5	0	0
0	0	1/5	0	0
0	0	1/5	0	0
0	0	1/5	0	0

1/5	0	0	0	0
0	1/5	0	0	0
0	0	1/5	0	0
0	0	0	1/5	0
0	0	0	0	1/5

0	0	0	0	1/5
0	0	0	1/5	0
0	0	1/5	0	0
0	1/5	0	0	0
1/5	0	0	0	0

Lewy górny wygładza w poziomie, a prawy górny w pionie. Dolne wygładzają wzdłuż kierunków nachylonych pod kątem 45 oraz 135 stopni.

Obliczanie pochodnych kierunkowych w obrazie

- *Pochodna kierunkowa pozioma z różnic centralnych :*

$$\frac{\partial f}{\partial x}(x, y) \approx \frac{f(x+1, y) - f(x-1, y)}{2}$$

Może być odwzorowana w obrazie wynikowym przez filtr (dodatkowo odpowiednio znormalizowany)

-1	0	1
-1	0	1
-1	0	1

- Analogicznie *pochodna kierunkowa pionowa z różnic centralnych :*

$$\frac{\partial f}{\partial y}(x, y) \approx \frac{f(x, y+1) - f(x, y-1)}{2}$$

może być reprezentowana przez filtr typu

-1	-1	-1
0	0	0
1	1	1

- Podobnie *pochodna kierunkowa w kierunku [1,-1] z różnic centralnych :*

$$\frac{\partial f}{\partial y}(x, y) \approx \frac{f(x+1, y+1) - f(x-1, y-1)}{2\sqrt{2}}$$

może być reprezentowana przez filtr typu

-1	-1	0
-1	0	1
0	1	1

- *Laplasjan obrazu $f(x,y)$* reprezentuje drugą pochodną w 2D :

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

może być przybliżony przez

$$\nabla^2 f(x, y) \approx f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

- Zatem może być przybliżony przez jeden z filtrów

0	-1	0
-1	4	-1
0	-1	0

0	1	0
1	-4	1
0	1	0

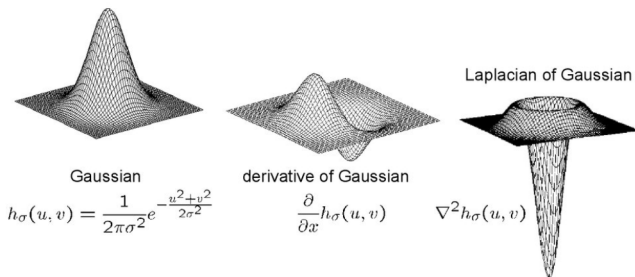
- Jedną z wad takiego przybliżenia jest jego anizotropowość - bardziej izotropowe wersje są reprezentowane przykładowo przez filtry

-1	-1	-1
-1	8	-1
-1	-1	-1

1/6	2/3	1/6
2/3	-10/3	2/3
1/6	2/3	1/6

Obliczanie laplasjanu obrazu

- ▶ Inną wadą *laplasjanu* jest jego wrażliwość na szum - w celu jej zniwelowania działanie laplasjanu poprzedza się filtrem wygładzającym Gaussa
- ▶ Z łączności spłotu wynika, że można zbudować filtr wynikowy, będący spłotem filtrów laplasjanu i Gaussa i do obrazu zastosować ten wynikowy filtr - nosi on nazwę *Laplacian of Gauss* i jest oznaczany symbolem *LoG*.



- ▶ Jednym z filtrów używanych do jego reprezentacji jest

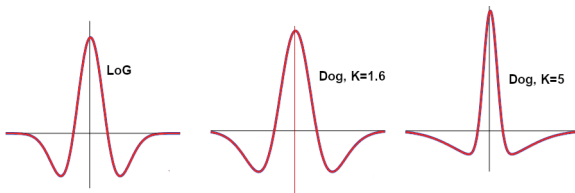
0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	0
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-50	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	0
0	1	1	2	2	2	1	1	0

LoG jako różnica filtrów Gaussa

- **LoG** można przybliżyć różnicą 2 filtrów Gaussa o różnym odchyleniu standardowym - taki filtr wynikowy oznaczony jest symbolem **DoG** (Difference of Gauss)

$$DoG(x, y, \sigma, K) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} - \frac{1}{2\pi K^2\sigma^2} e^{-\frac{x^2+y^2}{2K^2\sigma^2}}$$

- dla $K \approx 1.6$ DoG najlepiej przybliża LoG
- dla $K \approx 5$ DoG, wg pewnych teorii, przybliża sposób widzenia krawędzi przez człowieka



Wykrywanie krawędzi

- **Krawędź** w obrazie reprezentowana jest przez istotną zmianę jasności w ustalonym kierunku przy braku zmiany jasności w kierunku prostopadłym do ustalonego
- Naturalnym jest zatem szukanie krawędzi w obszarach z maksymalną wartością pochodnej (w 2D można liczyć jej moduł z sumy pochodnych kierunkowych)
- Punkty z maksimum pierwszej pochodnej są punktami z zerową drugą pochodną
- Z powyższych faktów wynika kilka różnych koncepcji szukania krawędzi w obrazie
- **Filtr Robertsza** - używane są tu dwa składowe filtry obliczające pochodne kierunkowe w kierunkach $[1,1]$ oraz $[-1,1]$, reprezentowane przez jądro

$$r_1 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad r_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Za ich pomocą oblicza się wektor pochodnych kierunkowych w każdym pikselu obrazu, a następnie jego moduł w metryce l_1 (prościej) lub l_2 . Finalny obraz jest reprezentowany przez te moduły. Filtr ten jest dość wrażliwy na szum, daje dość nieprecyzyjne i grube krawędzie, ale jego zaletą jest prostota i szybkość obliczeń.

- **Filtry Sobela i Prewita** - działają identycznie jak filtr Robertsza, ale każdy używa innego zestawu jądra. Zarówno Sobel jak i Prewit reprezentuje pochodne kierunkowe obliczone wg różnic centralnych, ale w przypadku Sobela z większymi wagami przyłożonymi dla kierunku przechodzącego przez analizowany punkt

$$s_1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad s_2 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$
$$p_1 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad p_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Operator Sobela jest nieco bardziej izotropowy, a oba są bardziej odporne na szum niż filtr Robertsza.

- **Filtr Kirscha** - stosuje 8 postaci jądra do obliczenia pochodnych kierunkowych (w klasycznych kierunkach geograficznych) w pikselu i wybiera ostatecznie tą, która daje maksymalną odpowiedź.

5	5	5
-3	0	-3
-3	-3	-3

5	5	-3
5	0	-3
-3	-3	-3

5	-3	-3
5	0	-3
5	-3	-3

-3	-3	-3
5	0	-3
5	5	-3

-3	-3	-3
-3	0	-3
5	5	5

-3	-3	-3
-3	0	5
-3	5	5

-3	-3	5
-3	0	5
-3	-3	5

-3	5	5
-3	0	5
-3	-3	-3

- **Filtr oparty na obliczeniu Laplasjanu** - wagi jądra laplasjanu sumują się do zera, zatem laplasjan daje odpowiedź zero w obszarach jednostajnej jasności. Jego istotą jest rozróżnianie dodatnich i ujemnych wartości drugiej pochodnej po przeciwnych stronach krawędzi. Filtr ten jest dość wrażliwy na szum, więc częściej stosuje się wersję LoG lub DoG.
- **Detekcja krawędzi przez przejście przez zero drugiej pochodnej** - oblicza się tu (jak powyżej) obraz wartości laplasjanu i następnie stosuje się tu jedną z trzech ogólnie koncepcji - druga i trzecia będzie możliwa do realizacji w późniejszej części kursu:

- **Szukanie sąsiednich pikseli o przeciwnych znakach:**

jeżeli $\text{LoG}(x, y) \cdot \text{LoG}(x+i, y+j) < 0$ and $|\text{LoG}(x, y)| < |\text{LoG}(x+i, y+j)| \Rightarrow \text{piksel} \in \text{krawędź}$

- **Progowanie wartością zero**

- **Operacje morfologii matematycznej**

- *Filtr Frei'a Chena* - stosuje 9 postaci jądra:

-1	$-\sqrt{2}$	-1
0	0	0
1	$\sqrt{2}$	1

-1	0	-1
$-\sqrt{2}$	0	$\sqrt{2}$
-1	0	1

0	-1	$\sqrt{2}$
1	0	-1
$-\sqrt{2}$	1	0

$\sqrt{2}$	-1	0
-1	0	1
0	1	$-\sqrt{2}$

0	1	0
-1	0	1
1	-1	0

-1	0	1
0	0	0
1	0	-1

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

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

1	1	1
1	1	1
1	1	1

Za wykrywanie krawędzi odpowiadają pierwsze 4 jądra. Kolejne 4 służą do wykrywania linii. Całość stanowi bazę ortogonalną 9-cio wymiarowej przestrzeni.

- *Uwaga:* Do zagadnienia wykrywania krawędzi wrócimy na zajęciach poświęconych filtracji w przestrzeni częstotliwości.

Filtry wyostrzające

- Naturalną koncepcją jest sumowanie wartości obrazu i jakiegoś filtra krawędziowego - przykładowo filtra laplasjanu:

$$\begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 9 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|c|} \hline -1 & -1 & -1 & -1 & -1 \\ \hline -1 & -1 & -1 & -1 & -1 \\ \hline -1 & -1 & 24 & -1 & -1 \\ \hline -1 & -1 & -1 & -1 & -1 \\ \hline -1 & -1 & -1 & -1 & -1 \\ \hline \end{array} + \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|} \hline -1 & -1 & -1 & -1 & -1 \\ \hline -1 & -1 & -1 & -1 & -1 \\ \hline -1 & -1 & 25 & -1 & -1 \\ \hline -1 & -1 & -1 & -1 & -1 \\ \hline -1 & -1 & -1 & -1 & -1 \\ \hline \end{array}$$

Filtry drugiej pochodnej można w powyższych formułach dodatkowo skalować przez ustaloną stałą.

- Można również zastosować koncepcję dwukrokową ($f(x, y)$ oznacza oryginalny obraz)

$$g(x, y) = f(x, y) - f_{smooth}(x, y)$$

$$f_{sharp}(x, y) = f(x, y) + k \cdot g(x, y)$$

gdzie f_{smooth} oznacza obraz po działaniu jakiegoś filtra wygładzającego. Wartość k dobrze jest wziąć z zakresu $[0.2, 0.7]$ (wyższa oznacza silniejsze wyostanie).