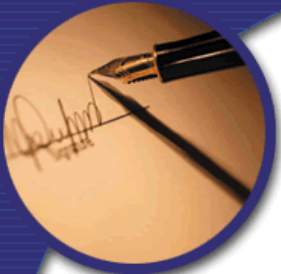


Grafika komputerowa

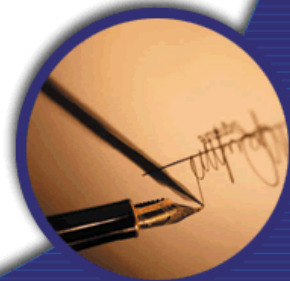
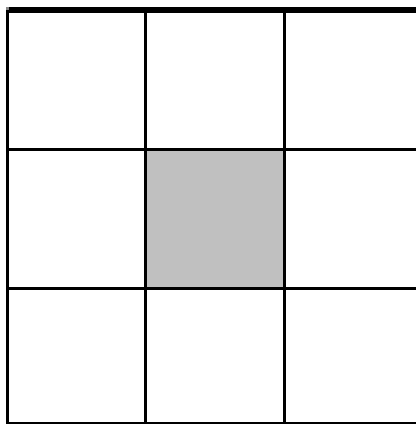
Dr inż. Michał Kruk

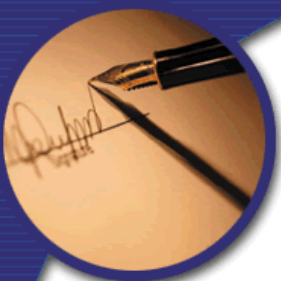




Operacje kontekstowe

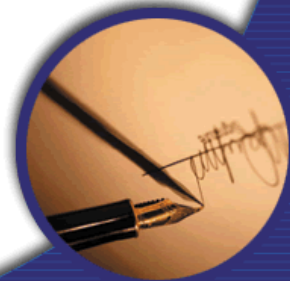
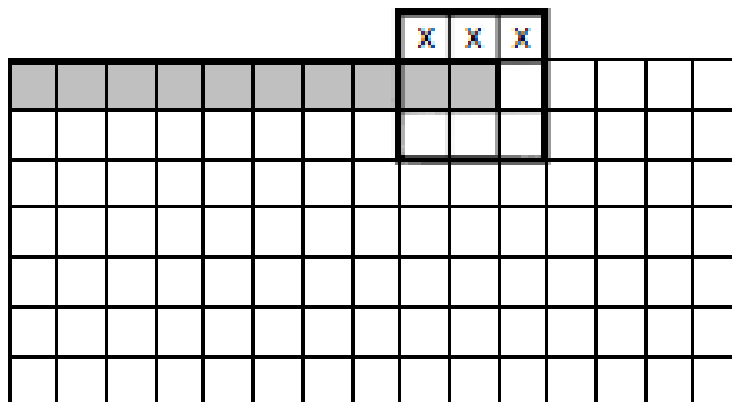
- Z reguły filtry używane do analizy obrazów zakładają, że wykonywane na obrazie operacje będą **kontekstowe**
- Polega to na wyznaczeniu wartości funkcji, której argumentami są wartości piksela o tym samym położeniu na obrazie źródłowym oraz wartości pikseli z jego otoczenia K , które w ogólnym przypadku może mieć różną formę, ale najczęściej utożsamiane jest z kwadratowym „**oknem**” otaczającym symetrycznie aktualnie przetwarzany punkt obrazu

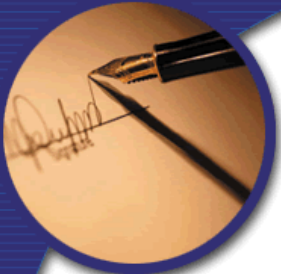




Operacje kontekstowe

- Z powodu kontekstowości wykonywanych operacji filtracja z reguły nie może dotyczyć pikseli znajdujących się bezpośrednio na brzegu obrazu



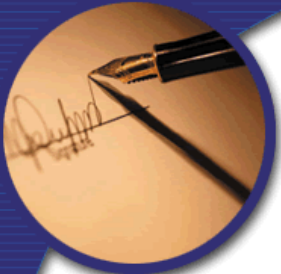


Praktyczne zastosowania

W praktycznych zastosowaniach filtry wykorzystywane są zazwyczaj do realizacji następujących celów:

- **Stłumienie w obrazie niepożądanego szumu.** Przy braku konkretnych przesłanek na temat istoty szumu realizujący tę funkcję filtr działa zazwyczaj na zasadzie lokalnych średnich. Każdemu z punktów obrazu przypisywana jest w takim przypadku średnia wartości jego otoczenia.
- **Wzmocnienie** w obrazie pewnych elementów zgodnych z posiadanym wzorcem. W tym przypadku dany punkt zostanie wzmocniony w stopniu zależnym od spełniania przez jego otoczenie określonych warunków.
- **Usunięcie** określonych wad z obrazu. Na przykład: usunięcie wad powstałych wskutek zarysowania kliszy fotograficznej.
- **Poprawa obrazu** o złej jakości technicznej. Na przykład: obrazów nieostrych, poruszonych lub o niewielkim kontraście.
- **Rekonstrukcja obrazu**, który uległ częściowemu zniszczeniu. Na przykład: rekonstrukcja materiałów fotograficznych, które przez długi czas podlegały działaniom niekorzystnych warunków.

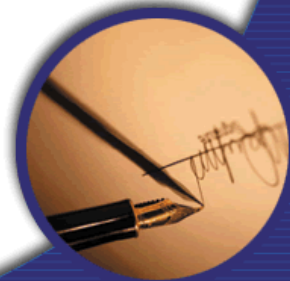


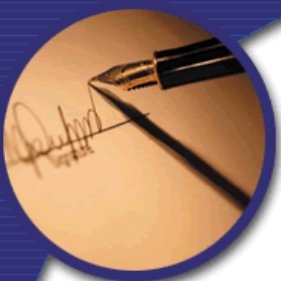


Podział

Będziemy wyróżniali filtry:

- liniowe (wykonujące operację filtracji oparciu o pewną liniową kombinację wybranych pikseli obrazu wejściowego);
- nieliniowe (wykonujące operację filtracji w oparciu o pewną nieliniową funkcję wybranych pikseli obrazu wejściowego).
- Filtry liniowe są z reguły prostsze w wykonaniu, natomiast filtry nieliniowe mają z reguły bogatsze możliwości.





Filtry liniowe

Filtry zaliczane są do liniowych, jeżeli funkcja je realizująca spełnia dwa warunki liniowości:

- jest addytywna

$$\varphi(f + g) = \varphi(f) + \varphi(g) ;$$

- oraz jednorodna

$$\varphi(\lambda f) = \lambda \varphi(f) , \lambda \in \mathfrak{R}$$

gdzie:

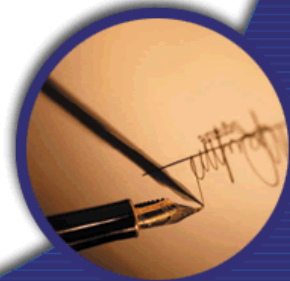
φ - funkcja realizująca filtr;
 f, g - obrazy podlegające filtracji.

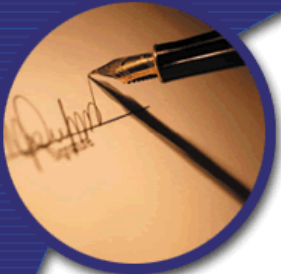
W praktyce najczęściej wykorzystywane są filtry spełniające dodatkowo warunek niezmienności względem przesunięcia:

$$\varphi(f_{\vec{h}}) = [\varphi(f)]_{\vec{h}}$$

gdzie dodatkowo:

$\vec{h}$ - wektor przesunięcia.





Konwolucja

- Inaczej zwana splotem funkcji

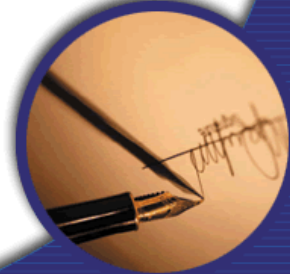
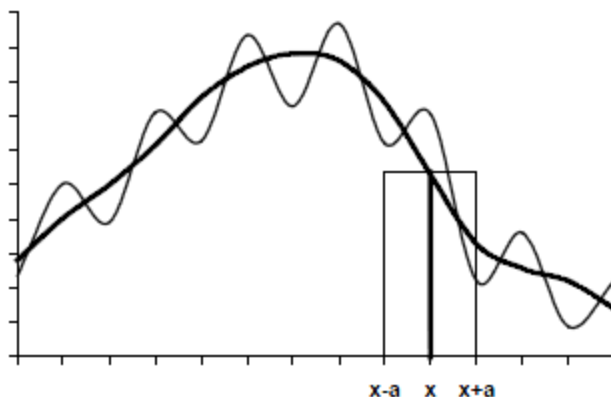
$$g(x) = (f \times h)(x) = \int_{-\infty}^{+\infty} f(x-t)h(t)dt$$

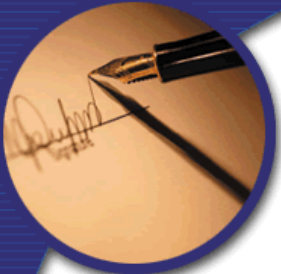
gdzie:

f, h - splatane funkcje.

- Wspomnianą wyżej funkcję realizującą tłumienie szumów na zasadzie lokalnych średnich można realizować jako następującą konwolucję:

$$g(x) = \frac{1}{2a} \int_{-a}^a f(t)dt = (f \times h)(x), \quad \text{gdzie:} \quad h(u) = \begin{cases} \frac{1}{2a}; & u \in (-a, a) \\ 0; & u \notin (-a, a) \end{cases}$$

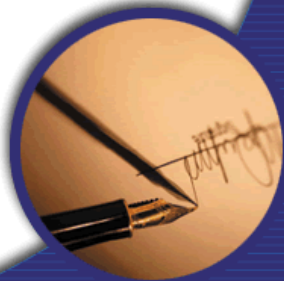


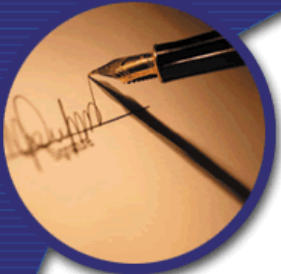


Własności

Konwolucja posiada pewne własności, które są bardzo pomocne w praktycznej realizacji filtrowania. Do własności tych należą, między innymi:

- **Łączność:** $(f \times g) \times h = f \times (g \times h) = f \times g \times h$ Własność ta pozwala na rozdzielenie filtrowania dowolnie dużą matrycą na kolejne filtrowanie przy pomocy małych matryc.
- **Rozdzielność.** Pozwala to rozdzielić filtrację dwuwymiarowego obrazu jako złożenie filtracji jednowymiarowych.

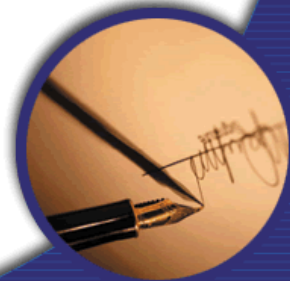


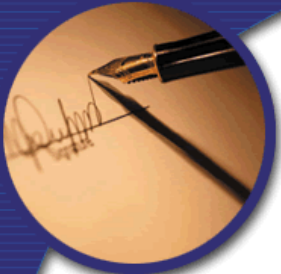


Konwolucja dyskretna

- W komputerowej analizie obrazu dziedziną funkcji $L(m,n)$ jest dwuwymiarowa i dyskretna (nieciągła), gdyż taka jest reprezentacja obrazu.
- Konwolucję dla 2-wymiarowego, dyskretnego obrazu można zapisać w następujący sposób:

$$L'(m,n) = (w \times L)(m,n) = \sum_{i,j \in K} L(m-i, n-j) w(i,j)$$

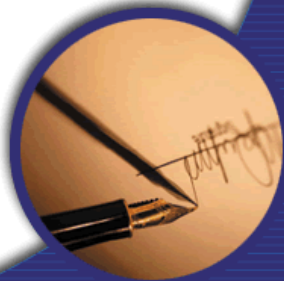


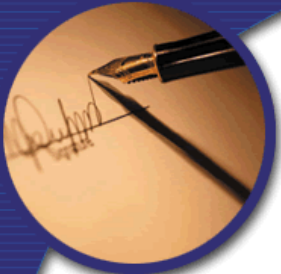


Normalizacja

- Współczynniki $w(i,j)$ wybiera się zwykle w taki sposób, by były liczbami całkowitymi.
- Powód takiego wyboru jest oczywisty - dla uzyskania wartości $L'(m,n)$ dla wszystkich punktów obrazu trzeba w typowych warunkach wykonać ogromną liczbę dodawań i mnożeń (dla obrazu o typowych rozmiarach 512x512 punktów i otoczenia K obejmującego zaledwie 9 punktów konieczne jest wykonanie aż 2.359.296 dodawań i mnożeń!)
- Dla filtrów eliminujących proste zakłócenia, dla których wszystkie współczynniki spełniają warunek $w(i,j) \geq 0$ możliwe jest zastosowanie stosunkowo prostej techniki normalizacji, danej wzorem:

$$L'(m,n) = \frac{1}{\sum_{(i,j) \in K} w(i,j)} \sum_{(i,j) \in K} L(m-i, n-j) w(i,j)$$

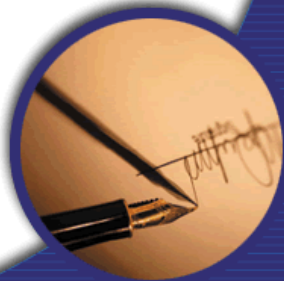


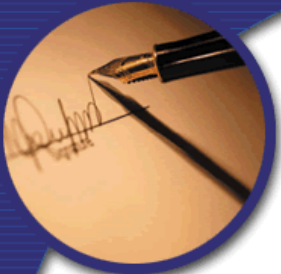


Normalizacja

- Gorzej jest w przypadku operatorów odwołujących się do współczynników $w(i,j)$ przyjmujących zarówno wartości dodatnie, jak i ujemne.
- W takich przypadkach operacja normalizacji musi odwoływać się do rzeczywistych, uzyskanych po przetworzeniu obrazu wartości $\min L'(m,n)$ oraz $\max L'(m,n)$ i opierać się na wzorze:

$$L''(m,n) = \frac{L'(m,n) - \min L'(m,n)}{\max L'(m,n) - \min L'(m,n)} 2^B$$





Okno

- Najczęściej spotykamy okna o wymiarach 3x3
następującej postaci:

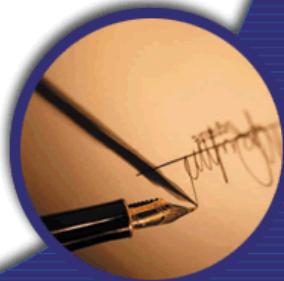
$$\begin{bmatrix} w_{11} & w_{10} & w_{1-1} \\ w_{01} & w_{00} & w_{0-1} \\ w_{-11} & w_{-10} & w_{-1-1} \end{bmatrix}$$

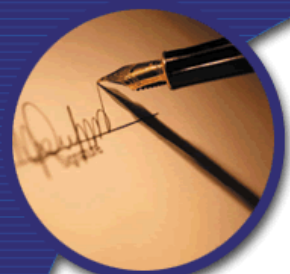
- W skrócie:

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$$

W tym przypadku funkcja realizująca proces filtracji (z użyciem konwolucji) może być zapisana w następujący sposób:

$$\begin{aligned} L'(m,n) = & w_1 L(m-1,n-1) + w_2 L(m-1,n) + w_3 L(m-1,n+1) \\ & + w_4 L(m,n-1) + w_5 L(m,n) + w_6 L(m,n+1) \\ & + w_7 L(m+1,n-1) + w_8 L(m+1,n) + w_9 L(m+1,n+1) \end{aligned}$$





Złożoność obliczeniowa

zakładając $M=N$ i $n=n$:

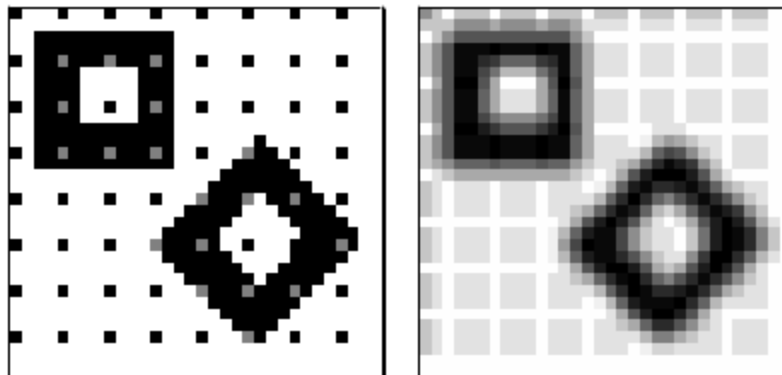
- – N^2n^2 mnożeń
- – N^2n^2 dodawań
- koszt $O(N^2 \log^2(N))$ – przy użyciu FFT (w dalszej części wykładu)



Filtracja dolnoprzepustowa

1	1	1
1	1	1
1	1	1

- Jak łatwo zauważyć, filtr uśredniający usuwa drobne zakłócenia z obrazu - znikają (a raczej ulegają znacznemu osłabieniu ale i pewnemu .rozmazaniu.) pojedyncze czarne punkty na jasnym tle lub jasne plamki na tle ciemnym, wygładzane są drobne .zawirowania. krawędzi obiektów, usuwane mogą być efekty falowania jasności zarówno w obszarze samych obiektów, jak i w obszarze tła, itp.



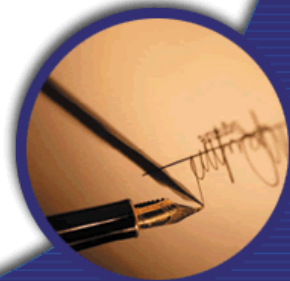


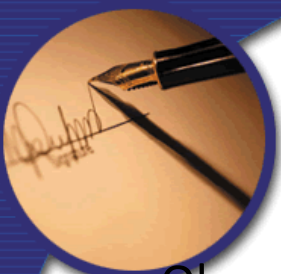
Filtracja dolnoprzepustowa

- W celu zmniejszenia negatywnych skutków filtracji konwolucyjnej stosuje się czasem filtry uśredniające wartości pikseli wewnątrz rozważanego obszaru w sposób ważony. - tak, aby pierwotna wartość piksela $L(m,n)$ w większym stopniu wpływała na wartość piksela po przetworzeniu $L'(m,n)$.

1	1	1
1	2	1
1	1	1

1	2	1
2	4	2
1	2	1





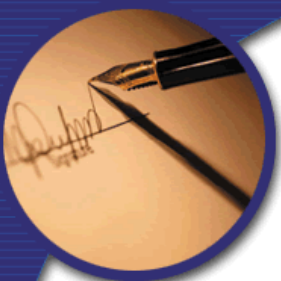
Przykłady

- Obraz oryginalny i maska „jedynkowa”



- Po filtracji z uwydatnionym punktem centralnym





Przykłady

obraz oryginalny $I(x,y)$



- czym większa maska (mniejszy wpływ piksela centralnego) tym rozmycie większe



Przykłady

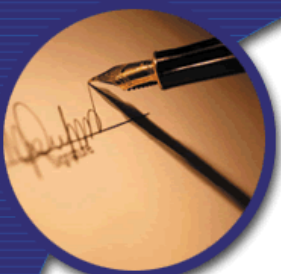
obraz oryginalny $I(x,y)$



obraz z szumem $J(x,y)=I(x,y)+N(0,\sigma)$



- szum ulega zmniejszeniu kosztem rozmycia krawędzi i utraty szczegółów



Przykłady

obraz oryginalny $I(x,y)$



obraz z szumem $J(x,y)$: losowe piksele z I ustawione na losową wartość



- szum trudniejszy do usunięcia przez ten typ filtra



Filtr Gaussa

- filtr Gaussa – współczynniki maski wyliczane są z równania rozkładu normalnego zakładając że centralny element ma współrzędne $x=0$ i $y=0$ oraz jednakową wariancję $\sigma_x=\sigma_y=\sigma$

$$\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\sigma=1 \begin{bmatrix} 0.06 & 0.1 & 0.06 \\ 0.1 & 0.16 & 0.1 \\ 0.06 & 0.1 & 0.06 \end{bmatrix} \Sigma F_{ij}=0,78$$

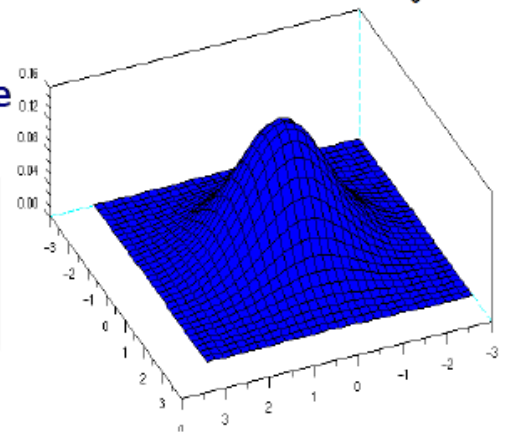
$$\begin{bmatrix} 0 & 0.01 & 0.02 & 0.01 & 0 \\ 0.01 & 0.06 & 0.1 & 0.06 & 0.01 \\ 0.02 & 0.1 & 0.16 & 0.1 & 0.02 \\ 0.01 & 0.06 & 0.1 & 0.06 & 0.01 \\ 0 & 0.01 & 0.02 & 0.01 & 0 \end{bmatrix} \Sigma F_{ij}=0,98$$

$$\sigma=1,5 \begin{bmatrix} 0 & 0 & 0.01 & 0.01 & 0.01 & 0 & 0 \\ 0 & 0.01 & 0.02 & 0.03 & 0.02 & 0.01 & 0 \\ 0.01 & 0.02 & 0.05 & 0.06 & 0.05 & 0.02 & 0.01 \\ 0.01 & 0.03 & 0.06 & 0.07 & 0.06 & 0.03 & 0.01 \\ 0.01 & 0.02 & 0.05 & 0.06 & 0.05 & 0.02 & 0.01 \\ 0 & 0.01 & 0.02 & 0.03 & 0.02 & 0.01 & 0 \\ 0 & 0 & 0.01 & 0.01 & 0.01 & 0 & 0 \end{bmatrix} \Sigma F_{ij}=0,97$$

- im dalszy punkt tym mniejszy wpływ na wynik
- im większa wariancja rozkładu tym większe rozmycie
- często stosowana wersja ze współczynnikami -potęgami 2:

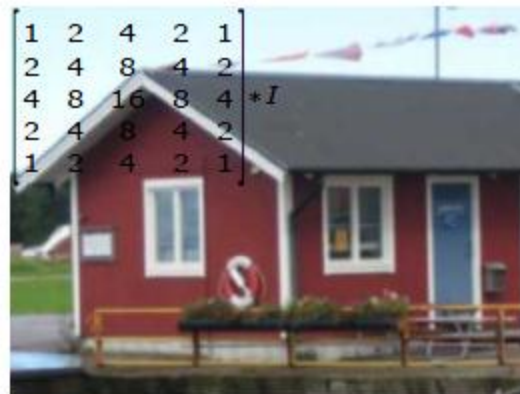
$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 2 & 1 \\ 2 & 4 & 8 & 4 & 2 \\ 4 & 8 & 16 & 8 & 4 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 1 \end{bmatrix}$$



Filtr Gaussa

obraz oryginalny $I(x,y)$

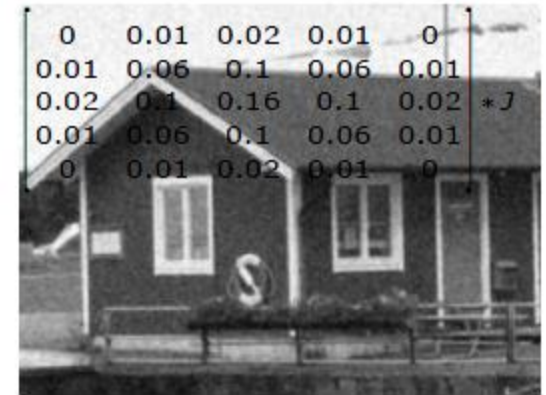
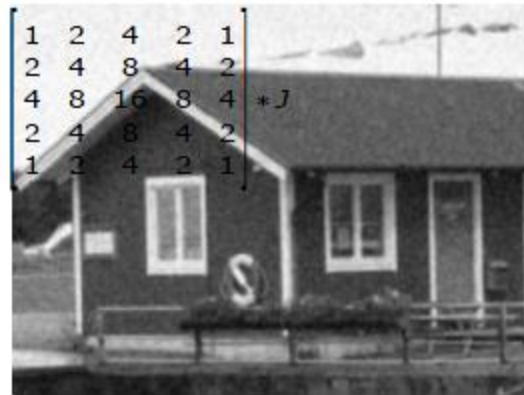


Filtr Gaussa – usuwanie szumu

obraz oryginalny $I(x,y)$



obraz z szumem $J(x,y)$





Filtr Gaussa – porównanie z maską jednolitą

obraz oryginalny $I(x,y)$



obraz z szumem $J(x,y)$



Filtr z maską jednolitą

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



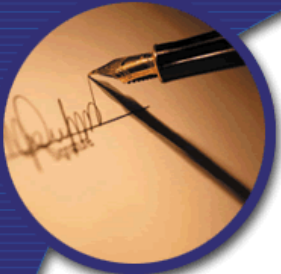
Filtr z maską Gaussa

0	0.01	0.02	0.01	0
0.01	0.06	0.1	0.06	0.01
0.02	0.1	0.16	0.1	0.02
0.01	0.06	0.1	0.06	0.01
0	0.01	0.02	0.01	0



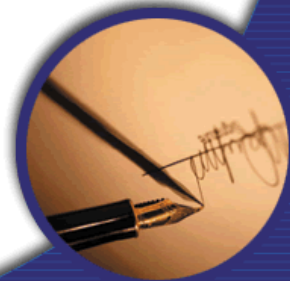
- filtr Gaussa lepiej zachowuje krawędzie i detale ale mniej skutecznie usuwa szum

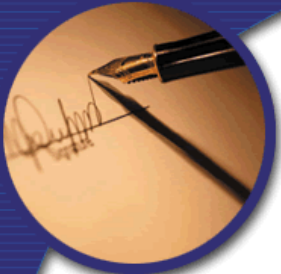




Filtry górnoprzepustowe

- Filtry tego typu służyć mogą do wydobywania z obrazu składników odpowiedzialnych za szybkie zmiany jasności - a więc konturów, krawędzi, drobnych elementów faktury, itp.
- Popularnie mówi się, że filtry górnoprzepustowe dokonują wyostrozania sygnału.





Filtry górnoprzepustowe

Krawędź jest to linia (w najprostszym przypadku prosta) oddzielająca obszary o różnej jasności L_1, L_2 . Prosty model matematyczny krawędzi ma postać skoku jednostkowego w zerze:

$$i(z) = \begin{cases} 1; z > 0 \\ \frac{1}{2}; z = 0 \\ 0; z < 0 \end{cases}$$

Zachodzi więc zależność:

$$i(z) = \int_{-\infty}^z \delta(t) dt; \text{ gdzie } \delta(t) - \text{delta Diraca}$$

Dla potrzeb analizy można przyjąć, że krawędź leży wzdłuż prostej o równaniu:

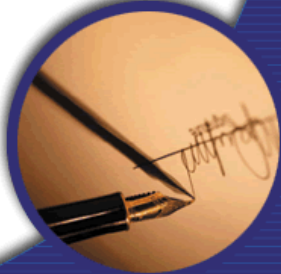
$$y = ax + b$$

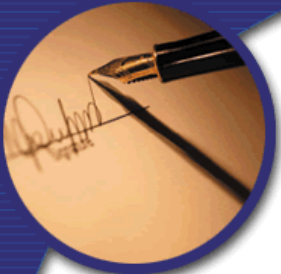
Ponadto: $a = \operatorname{tg} \alpha = \frac{\sin \alpha}{\cos \alpha}$; $b = \frac{\beta}{\cos \alpha}$; α, β - stałe.

$$y = \frac{\sin \alpha}{\cos \alpha} x + \frac{\beta}{\cos \alpha}$$

$$y \cos \alpha = x \sin \alpha + \beta$$

Postać kanoniczna:

$$x \sin \alpha - y \cos \alpha + \beta = 0$$




Filtry górnoprzepustowe

Jasność obrazu w poszczególnych punktach opisać można funkcją:

$$f(x, y) = L_1 + (L_2 - L_1)i(x \sin \alpha - y \cos \alpha + \beta)$$

Pierwsze pochodne cząstkowe mają postać:

$$\begin{aligned}\frac{\partial f}{\partial x} &= \sin \alpha (L_2 - L_1) \delta(x \sin \alpha - y \cos \alpha + \beta) \\ \frac{\partial f}{\partial y} &= -\cos \alpha (L_2 - L_1) \delta(x \sin \alpha - y \cos \alpha + \beta)\end{aligned}$$

Wektor $\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$ nazywany jest gradientem intensywności. Do wykrywania krawędzi używany jest kwadrat jego długości:

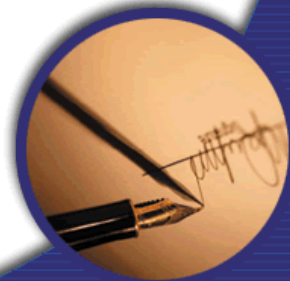
$$\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 = ((L_2 - L_1) \delta(x \sin \alpha - y \cos \alpha + \beta))^2$$

Obliczając drugie pochodne f :

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= \sin^2 \alpha (L_2 - L_1) \delta'(x \sin \alpha - y \cos \alpha + \beta) \\ \frac{\partial^2 f}{\partial x \partial y} &= -\sin \alpha \cos \alpha (L_2 - L_1) \delta'(x \sin \alpha - y \cos \alpha + \beta) \\ \frac{\partial^2 f}{\partial y^2} &= \cos^2 \alpha (L_2 - L_1) \delta'(x \sin \alpha - y \cos \alpha + \beta)\end{aligned}$$

otrzymujemy Laplasjan funkcji intensywności:

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} = (L_2 - L_1) \delta'(x \sin \alpha - y \cos \alpha + \beta)$$



Obliczanie gradientu

obraz oryginalny $I(x,y)$

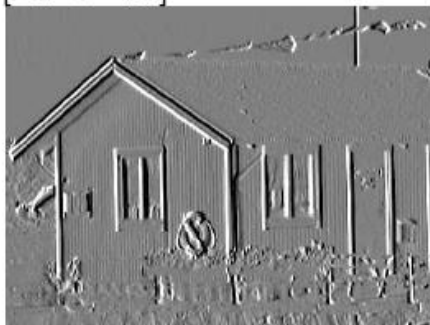


Obliczanie gradientu

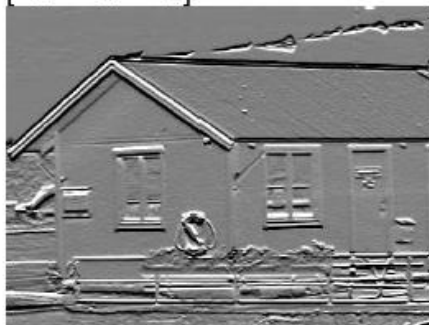
- operacja matematyczna gradientu
- gradient to pole wektorowe
- każda maska oblicza jeden gradient kierunkowy
- wartości wynikowe przesunięte o 128:
 - kolor czarny (0) – wartości ujemne
 - kolor szary (128) – zero
 - kolor biały (255) – wartości dodatnie

$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right] \approx \left[\frac{I(x+h, y) - I(x, y)}{h}, \frac{I(x, y+h) - I(x, y)}{h} \right]$$

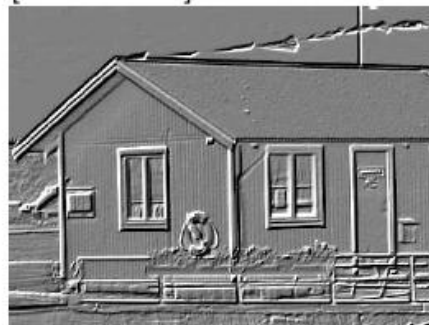
$$\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}, \nabla_x I = \left[\frac{\partial I}{\partial x}, 0 \right]$$



$$\begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, \nabla_y(I) = \left[0, \frac{\partial I}{\partial y} \right]$$



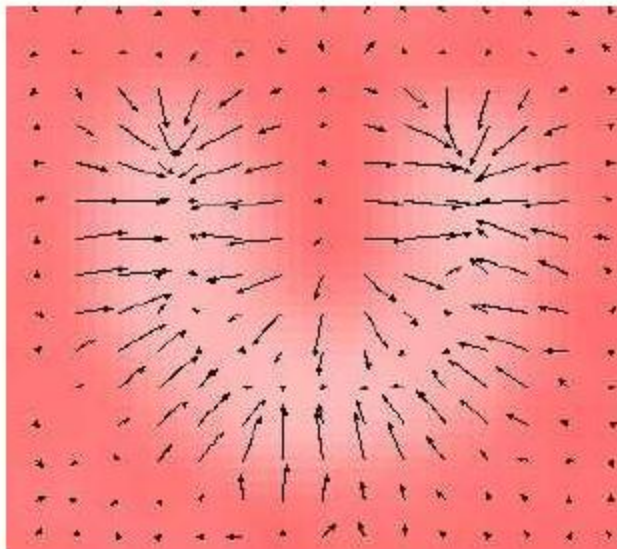
$$\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & +1 \\ 0 & +1 & +1 \end{bmatrix}, \nabla_{[1,-1]} I$$



Gradient jako pole wektorowe

$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]$$

$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x} \right)^2 + \left(\frac{\partial I}{\partial y} \right)^2}$$



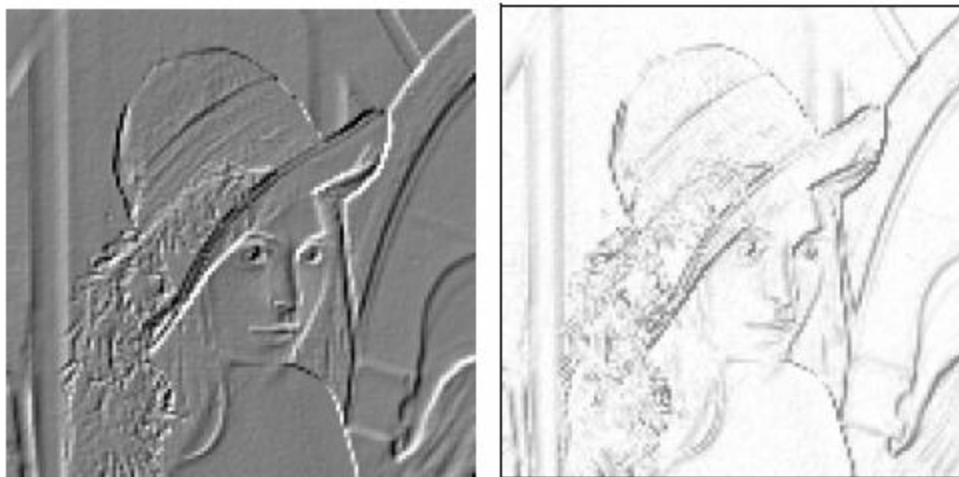
- pole wektorowe gradientu
- nałożone na obraz oryginalny
- wyliczone przez 2 konwolucje
- pokazuje kierunek wzrostu jasności

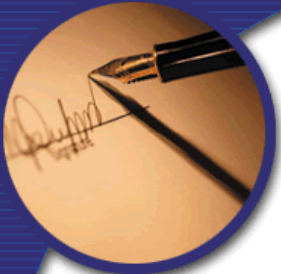


- moduł gradientu
- pole skalarne
- operacja nieliniowa !!!
- wyliczona przy pomocy 2 konwolucji

Gradient Roberts

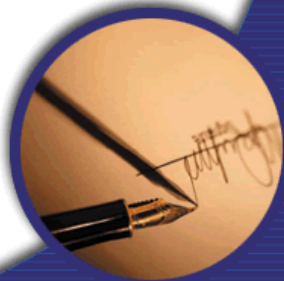
0	0	0
-1	0	0
0	1	0

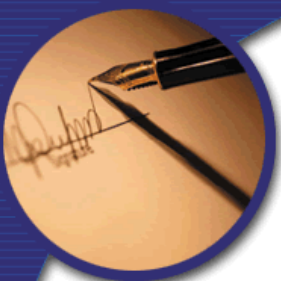




Gradient Robertsa

- w macierzach konwolucji filtrów górnoprzepustowych występują zarówno dodatnie, jak i ujemne współczynniki, co oznacza, że po wykonaniu splotu w obrazie wynikowym będą występowały zarówno piksele o wartościach dodatnich, jak i piksele o wartościach ujemnych
- w celu prawidłowego zinterpretowania wyniku konwolucji w postaci wynikowego obrazu trzeba albo dokonać skalowania, albo brać pod uwagę jedynie wartość bezwzględną odpowiednich pikseli
- przy skalowaniu obrazu wadą jest fakt, że dotychczasowe tło obrazu otrzymuje pewne .zabarwienie. (zwykle jest to szarość odpowiadająca wartości pikseli wynoszącej 128), natomiast piksele dodatnie są intensywnie ciemne, zaś piksele ujemne - pojawiają się jako jaśniejsze od tła.
- alternatywny sposób prezentacji, zakładający tworzenie obrazu wynikowego w postaci wartości bezwzględnych (modułów) wartości odpowiednich pikseli też ma swoje zalety, ale zwykle trochę .rozmywa. efekty, a ponadto ukrywa informacje o tym, czy poszczególne zmiany obliczone zostały jako dodatnie (wzrost jasności) czy jako ujemne (zmniejszenie jasności w odpowiednim kierunku)





Maska Prewitta

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



Obraz „Lena” filtrowany górnoprzepustowo z użyciem poziomej maski Prewitta.



Obraz „Lena” filtrowany górnoprzepustowo z użyciem pionowej maski Prewitta.



Maska Sobela

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



Obraz „Lena” filtrowany górnoprzepustowo z użyciem poziomej maski Sobela.

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



Obraz „Lena” filtrowany górnoprzepustowo z użyciem pionowej maski Sobela.

Maska Sobela – różne orientacje

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

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

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

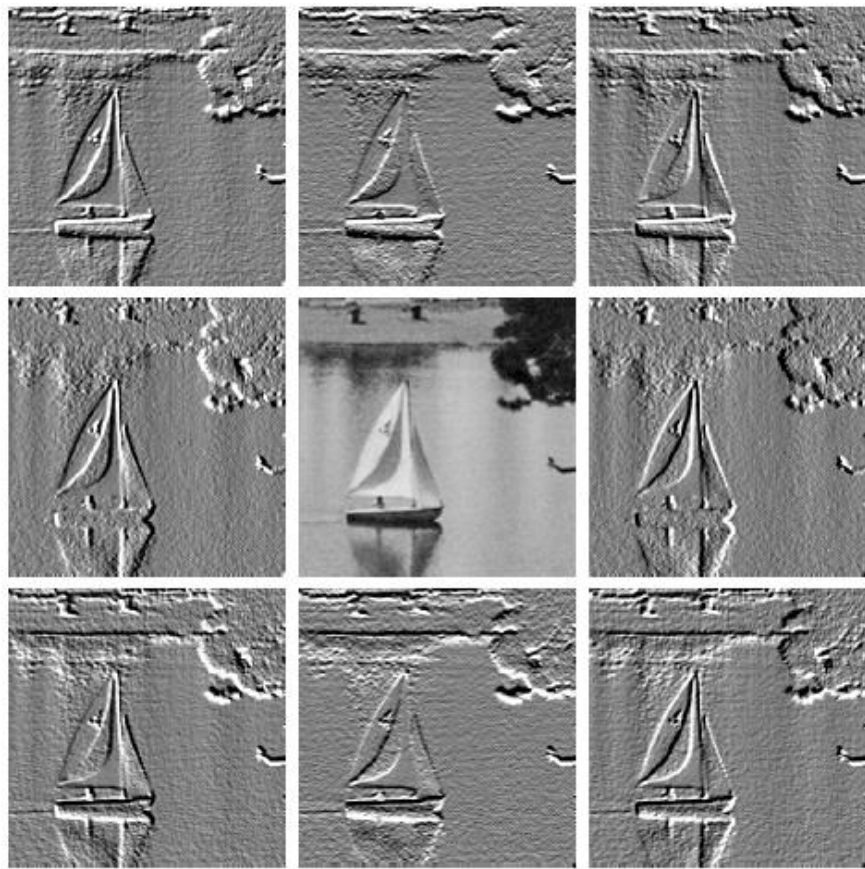
1	0	-1
2	0	-2
1	0	-1

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

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

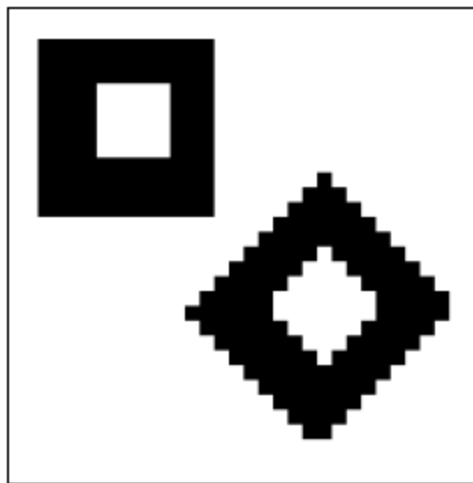
-1	-2	-1
0	0	0
1	2	1

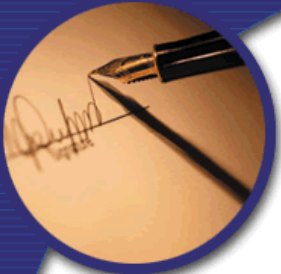
-2	-1	0
-1	0	1
0	1	2



Maski wykrywające narożniki

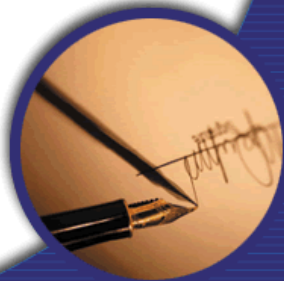
1	1	1
-1	-2	1
-1	-1	1





Laplasjan

- Laplasjan jest symetryczny względem obrotu oraz zachowuje znak krawędzi (mówi o tym, czy intensywność obrazu za krawędzią jest większa, czy mniejsza od intensywności obrazu przed krawędzią).
- Co ważne, jest to operator liniowy, a więc prosty w implementacji (np. jako konwolucja). Dlatego znajduje szerokie zastosowanie w przetwarzaniu obrazów



Obliczanie Laplasjanu

$$\Delta I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

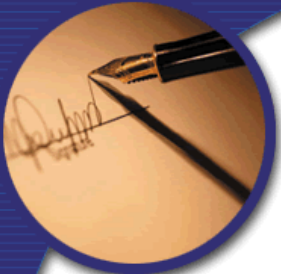
$$\Delta I \approx \frac{I(x+h, y) - 2 \cdot I(x, y) + I(x-h, y)}{h^2} + \frac{I(x, y+h) - 2 \cdot I(x, y) + I(x, y-h)}{h^2}$$

- druga pochodna w 2D
- pole skalarne
- przybliżane przez ilorazy różnicowe
- wartości wynikowe przesunięte o 128:
 - kolor czarny (0) – wartości ujemne
 - kolor szary (128) – zero
 - kolor biały (255) – wartości dodatnie
- wykorzystanie:
 - wykrywania linii
 - wykrywanie „plam” w różnej skali (ang. *blob detection*)

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

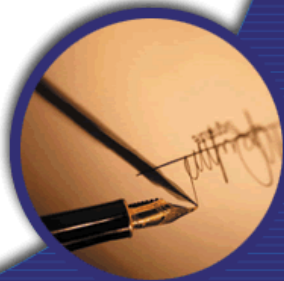
obraz oryginalny $I(x, y)$





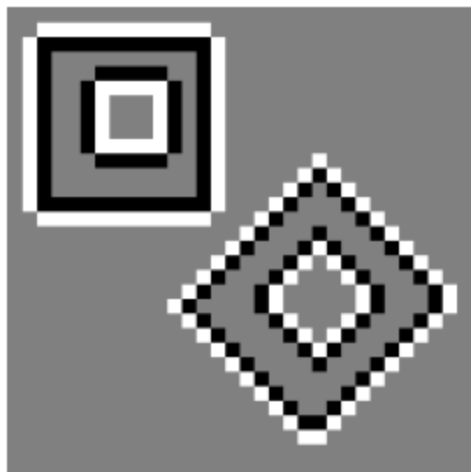
Laplasjan vs. gradient

- Wspólną cechą wszystkich wyżej wymienionych operatorów, wykorzystywanych do wyznaczania gradientów obrazu lub do wykrywania narożników (jako specjalnych form górnoprzepustowej filtracji obrazu), był fakt, że odpowiednie macierze konwolucji wykazywały określoną asymetrię.
- Następstwem tej asymetrii były kierunkowe własności realizowanych z użyciem tych macierzy transformacji obrazu - podkreślały one lub tłumiły pewne cechy obrazu - zależnie od tego, pod jakim kątem przebiegały określone krawędzie obiektów na obrazie w stosunku do kierunku asymetrii macierzy konwolucji.
- Czasem jednak zadania stawiane przed algorytmem przetwarzającym obrazy mają wyraźnie bezkierunkowy charakter - wtedy do filtracji górnoprzepustowej trzeba użyć innego typu masek



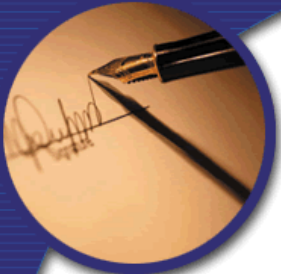
Laplasjan

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



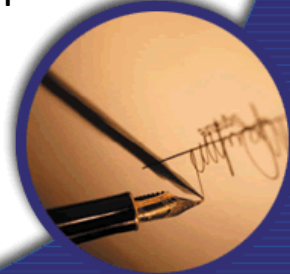
-1	-1	-1
-1	8	-1
-1	-1	-1



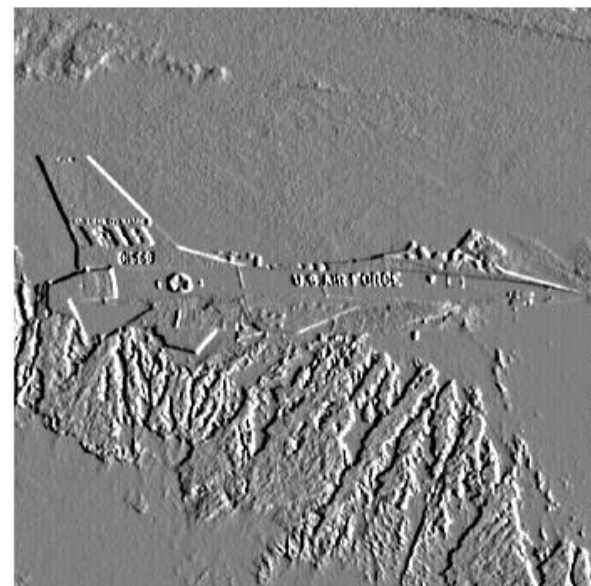
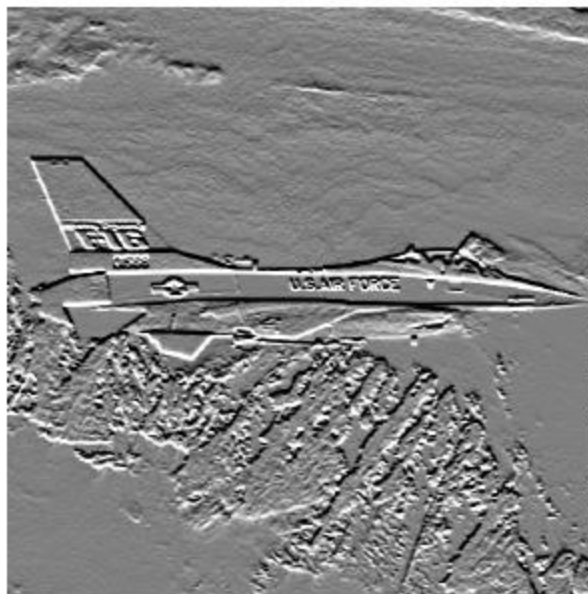


Filtracja nieliniowa – metody kombinowane

- Opisane laplasjany pozwalają wykrywać krawędzie i kontury obiektów na obrazach z dość dobrą dokładnością w większości praktycznych zastosowań, jednak wtedy, gdy chcemy użyć potem wyznaczonych krawędzi do jakichś bardziej zaawansowanych technik przetwarzania obrazów - potrzebne jest narzędzie lokalizujące kontury dokładniej i w sposób bardziej pewny niż laplasjan.
- W opisanych okolicznościach uciekamy się najczęściej do filtrów nieliniowych znanych pod nazwą filtrów kombinowanych.
- Idea tych filtrów polega na kolejnym zastosowaniu dwóch gradientów w prostopadłych do siebie kierunkach, a następnie na dokonaniu nieliniowej kombinacji wyników tych gradientów.
- Dzięki nieliniowej kombinacji rezultatów liniowych transformacji obrazu tworzy się w ten sposób obraz wynikowy o wyjątkowo dobrze podkreślonych konturach - niezależnie od kierunku ich przebiegu.

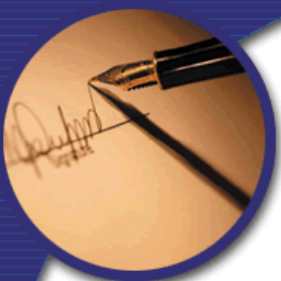


Euklidesowa kombinacja obrazów



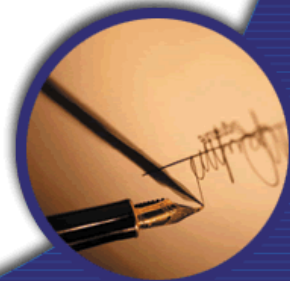
$$L'(m,n) = \sqrt{(L_1(m,n))^2 + (L_2(m,n))^2}$$



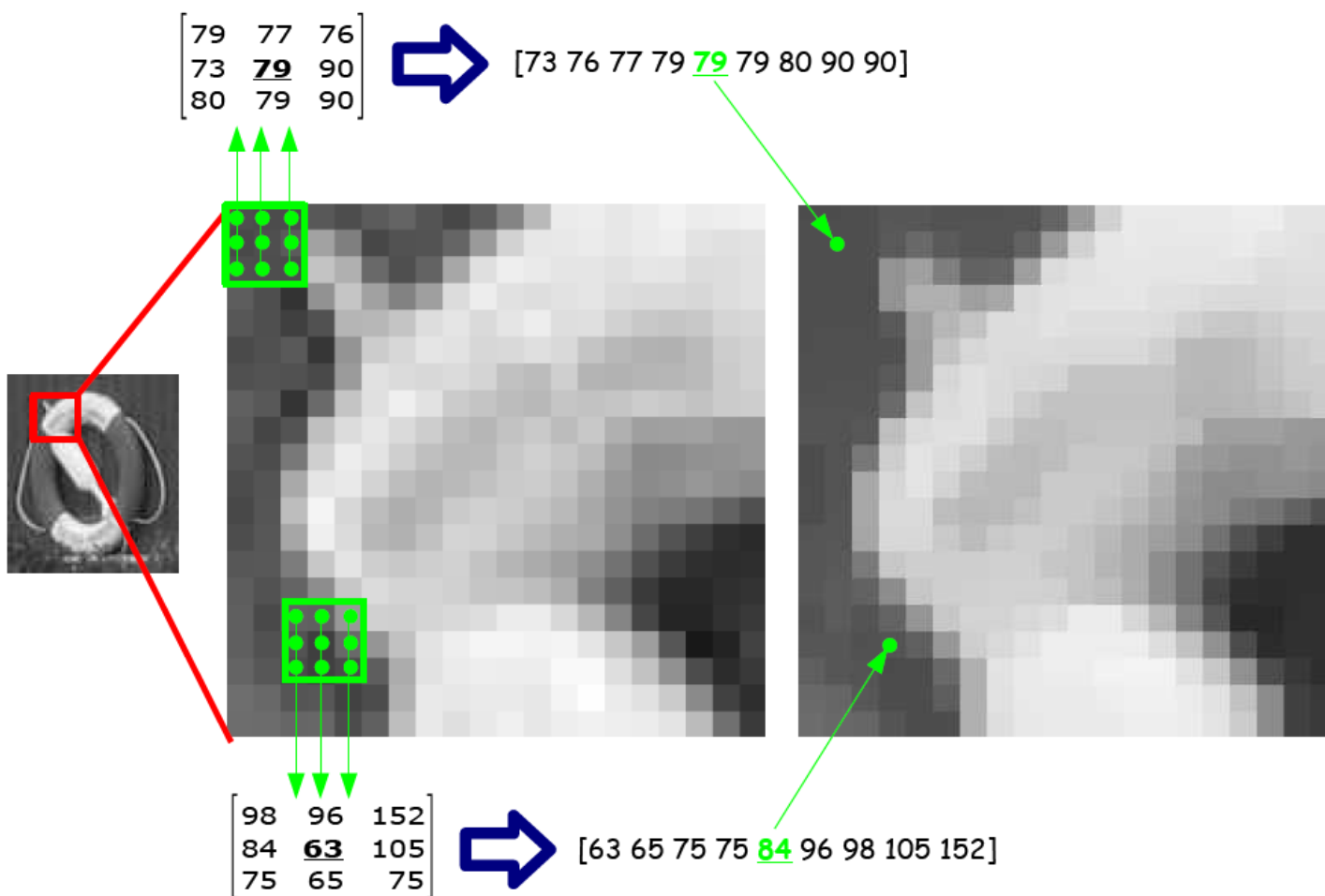


Filtry medianowe

- funkcja nieliniowa f jest medianą z argumentów:
funkcją sortującą swoje argumenty i wybierającą środkowy spośród nich
- ważna zaleta – wynik jest zawsze jednym z pikseli występujących w obrazie, co może być ważne w przypadku obrazów indeksowanych
- filtr medianowy jest często używany do usuwania drobnych zakłóceń, np szumu typu „*salt and pepper*”
- filtr ten nie powoduje rozmycia krawędzi
- jeśli sąsiedztwo brane pod uwagę wzrasta, filtrowanie może być wolne



Filtr medianowy



Działanie filtru medianowego

obraz oryginalny



filtr medianowy 3x3



filtr medianowy 5x5



- filtr usuwa detale o rozmiarze proporcjonalnym do promienia jego sąsiedztwa, zatem jest filtrem dolnoprzepustowym
- filtr nie rozmywa krawędzi
- filtr tworzy obszary o stałej jasności, bez faktury

filtr medianowy 11x11



filtr medianowy 21x21



Usuwanie szumu

obraz oryginalny



obraz z szumem „salt&pepper”



filtr medianowy 3x3



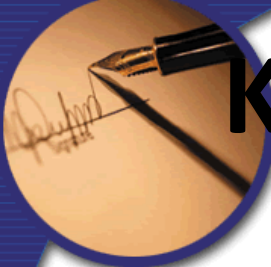
obraz z szumem białym



filtr medianowy 3x3



- filtr medianowy dobrze usuwa szum „salt&pepper”, gdzie tylko nieliczne piksele są zmienione
- gorzej radzi sobie z szumem białym gdzie wszystkie piksele są zmienione



Konserwatywne wygładzanie (conservative smoothing)

- filtr nieliniowy przeznaczony głównie do usuwania szumu impulsowego
- w minimalnym stopniu zniekształca obraz
- wyszukuje lokalne, odizolowane piksele z wartością mocno odbiegającą od otoczenia i tylko je usuwa:
 - dla każdego sąsiedztwa piksela $I(x,y)$ liczone są wartości *max* i *min* (bez brania pod uwagę tego piksela)
 - jeśli $I(x,y)$ należy do przedziału $\langle \min, \max \rangle$, jego wartość nie jest zmieniana

$$\begin{bmatrix} 79 & 77 & 76 \\ 73 & \mathbf{81} & 90 \\ 80 & 79 & 90 \end{bmatrix} \Rightarrow \begin{bmatrix} 79 & 77 & 76 \\ 73 & \mathbf{81} & 90 \\ 80 & 79 & 90 \end{bmatrix}$$

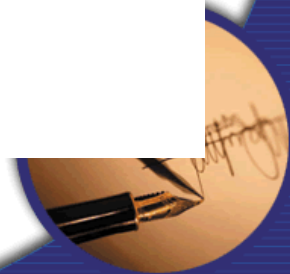
- jeśli $I(x,y) > \max$ wtedy $I(x,y) := \max$

$$\begin{bmatrix} 79 & 77 & 76 \\ 73 & \mathbf{204} & 90 \\ 80 & 79 & 90 \end{bmatrix} \Rightarrow \begin{bmatrix} 79 & 77 & 76 \\ 73 & \mathbf{90} & 90 \\ 80 & 79 & 90 \end{bmatrix}$$

- jeśli $I(x,y) < \min$ wtedy $I(x,y) := \min$

$$\begin{bmatrix} 79 & 77 & 76 \\ 73 & \mathbf{15} & 90 \\ 80 & 79 & 90 \end{bmatrix} \Rightarrow \begin{bmatrix} 79 & 77 & 76 \\ 73 & \mathbf{73} & 90 \\ 80 & 79 & 90 \end{bmatrix}$$

- filtr szybszy od medianowego – wyliczenie *min* i *max* jest szybsze niż posortowanie zakresu



Przykład



- jeśli impulsy szumu są zbyt bliskie sobie, nie zostaną usunięte
- w zamian za to detale nie są wygładzane (jak to się dzieje w filtrze medianowym)



Filtr wartości środkowej (midpoint)

- wartość piksela $I(x,y)$ jest zastępowana przez średnią z wartości maksymalnej i minimalnej z jego otoczenia:

$$J(x, y) = [\max(I(x-m : x+m, y-n : y+n)) + \min(I(x-m : x+m, y-n : y+n))] / 2$$

- stosowana głównie do białego szumu o rozkładzie normalnym lub jednorodnym



Filtr uśredniający z pomijaniem alfa wartości skrajnych

- przyjmując kwadratowe sąsiedztwo $n \times n$ (n nieparzyste) sortuje się wartości pikseli z tego sąsiedztwa:

$$[I_1, I_2, \dots, I_{n^2-1}, I_{n^2}]$$

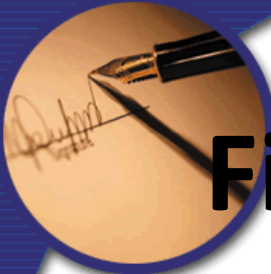
- z tego zakresu usuwa się α skrajnych (minimalnych i maksymalnych) wartości
- pozostałe wartości uśrednia się:

$$J(x, y) = \frac{1}{n^2 - 2 \cdot \alpha} \cdot \sum_{i=\alpha+1}^{n^2-\alpha} I_i$$

- dla $\alpha=0$ filtr jest filtrem uśredniającym o jednolitej masce
- dla $\alpha = (n^2-1)/2$ filtr jest filtrem medianowym

obraz z szumem białym po filtrowaniu, $\alpha=2$ obraz z szumem impulsowym po filtrowaniu, $\alpha=2$

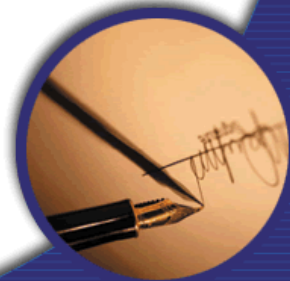




Filtr uśredniający adaptacyjny

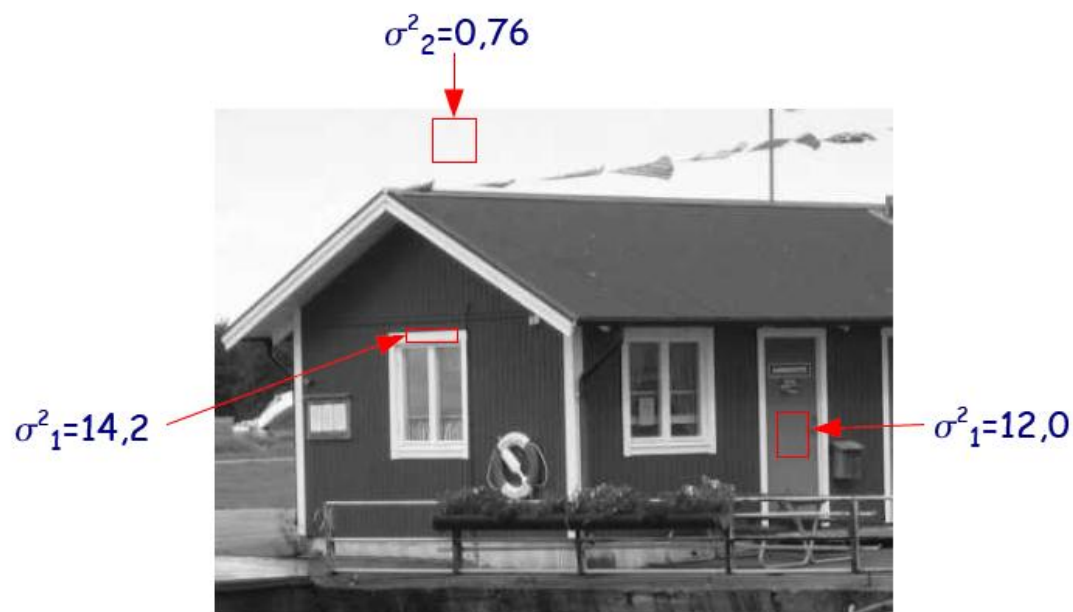
$$J(x, y) = I(x, y) - \frac{\sigma^2}{\sigma^2(x, y)} \cdot (I(x, y) - \text{mean}(x, y))$$

- σ^2 – wariancja szumu obrazu
- $\sigma^2(x, y)$ – wariancja w otoczeniu piksela $I(x, y)$
- $\text{mean}(x, y)$ – średnia intensywność w otoczeniu piksela $I(x, y)$
- dla obszarów jednolitych, bez detali, $\sigma^2 = \sigma^2(x, y)$ i $J(x, y) = \text{mean}(x, y)$ – obraz jest uśredniany
- dla obszarów ze szczegółami $\sigma^2(x, y) \gg \sigma^2$ i $J(x, y) = I(x, y)$ – obraz oryginalny się nie zmienia



Eksperymentalne wyliczanie wariacji szumu

- jako wariację szumu przyjmuje się wariację w obszarach uznanych za jednolite

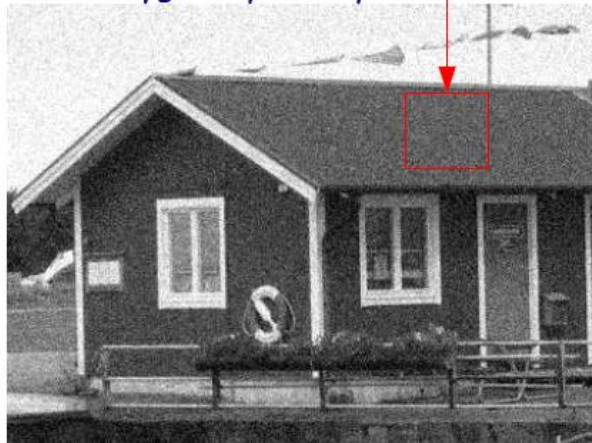


Filtr uśredniający adaptacyjny

obraz przefiltrowany z wyliczoną wariancją i sąsiedztwem 7x7

$$\sigma^2=340$$

obraz oryginalny z białym szumem



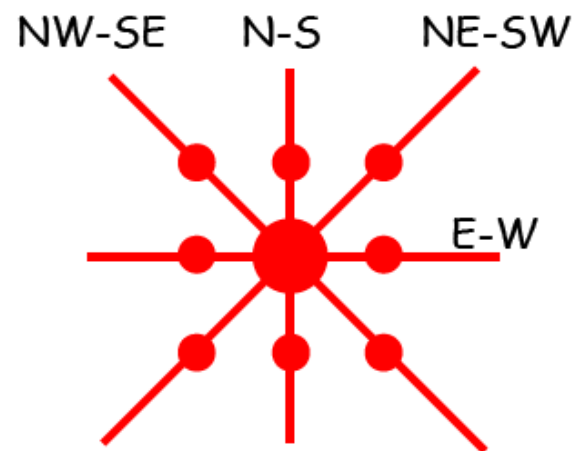
- uśrednianie odbywa się w obszarach „jednolitych”
- okolice krawędzi nie są filtrowane – nie ulegają one rozmyciu jak w uśrednianiu
- filtr kosztowny obliczeniowo





Filtr odplamiający Crimmins (Crimmins speckle removal)

- nieliniowy filtr iteracyjny **stopniowo** niwelujący lokalne odstępstwa intensywności pikseli od intensywności otoczenia
- przeznaczony do usuwania szumu impulsowego
- rodzaj filtra dolnoprzepustowego – stopniowo usuwa drobne detale
- dla każdego piksela jego intensywność jest porównywana z 2 sąsiadami na jednym z 4 kierunków
- jeżeli od nich odbiega, jest modyfikowana:
 - jest zmniejszana jeśli jest większa
 - jest zwiększana jeśli jest mniejsza
- analizowane są po kolei wszystkie kierunki
- algorytm jest złożony obliczeniowo !!!



Algorytm filtru Crimmins

pojedyncza iteracja filtrowania:

1. Rozjaśnianie pikseli (usuwanie szumu „pepper”)

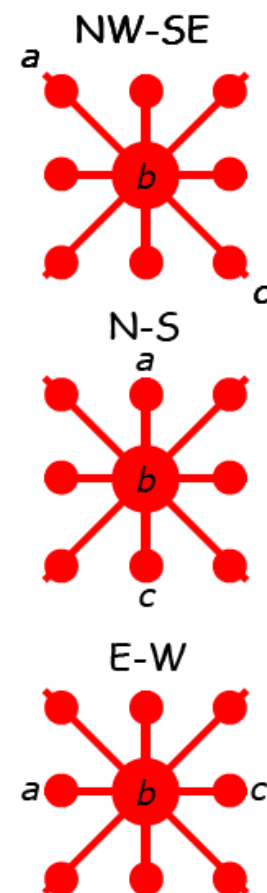
dla każdego z kierunków NW-SE, N-S, NE-SW, E-W:

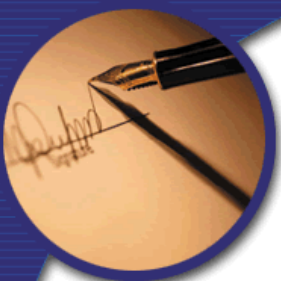
- oznacz sąsiednie piksele na kierunku jako a , b , c
- dla wszystkich trypletów (a, b, c) w obrazie:
jeśli $(a \geq b+2) \Rightarrow b+=1$;
- dla wszystkich trypletów (a, b, c) w obrazie:
jeśli $((a > b) \text{ i } (b \leq c)) \Rightarrow b+=1$;
- dla wszystkich trypletów (a, b, c) w obrazie:
jeśli $((c > b) \text{ i } (b \leq a)) \Rightarrow b+=1$;
- dla wszystkich trypletów (a, b, c) w obrazie:
jeśli $(c \geq b+2) \Rightarrow b+=1$;

2. Wygaszanie pikseli (usuwanie szumu „salt”)

dla każdego z kierunków NW-SE, N-S, NE-SW, E-W:

- oznacz sąsiednie piksele na kierunku jako a , b , c
- dla wszystkich trypletów (a, b, c) w obrazie:
jeśli $(a \leq b-2) \Rightarrow b-=1$;
- dla wszystkich trypletów (a, b, c) w obrazie:
jeśli $((a < b) \text{ i } (b \geq c)) \Rightarrow b-=1$;
- dla wszystkich trypletów (a, b, c) w obrazie:
jeśli $((c < b) \text{ i } (b \geq a)) \Rightarrow b-=1$;
- dla wszystkich trypletów (a, b, c) w obrazie:
jeśli $(c \leq b-2) \Rightarrow b-=1$;





Wyniki filtru Crimmins

obraz oryginalny z szumem impulsowym obraz po 1 iteracji



obraz po 3 iteracjach



obraz po 5 iteracjach



obraz po 10 iteracjach



- zmiany zachodzą stopniowo
- detale są usuwane
- krawędzie pozostają ostre
- szum zanika
- filtr bardzo wolny
(1 iteracja - kilka minut)



Inne filtry nieliniowe

- zwykle filtry specjalizowane wykrywające pewne rodzaje struktur:
 - krawędzie
 - zakończenia krawędzi
 - struktury kuliste
 - rogi
 - struktury liniowe
- przykład – wykrywanie naczyń krwionośnych 3D (struktur liniowych) w oparciu o wartości momentów geometrycznych wyliczonych w otoczeniu każdego punktu

obraz oryginalny, naczynia
zasłonięte przez kości



różne wersje filtrowania z użyciem momentów geometrycznych

