Technical University of Cluj - Napoca
Computer Science Department

Procesarea Imaginilor

Curs 9:

Prelucrari pe imagini multinivel (grayscale) (IV)

Filtre digitale

Filtrarea digitala a imaginilor

Filtrarea digitala - una dintre cele mai utilizate tehnici in procesarea imaginilor $\Rightarrow$ detectie muchii, eliminare zgomote, restaurare imagini

2 clase principale de filtre:

- **liniare**: iesirea este o combinatie liniara a valorilor pixelilor de intrare

  $\Rightarrow$ se aplica prin operatorul de convolutie

  ex: suma, medie aritmetica, …

- **neliniare**: iesirea este o combinatie ne-liniara a valorilor pixelilor de intrare

  $\Rightarrow$ modul de aplicare este dependent de tipul filtrului

  ex: minim, maxim, elem. median

Technical University of Cluj Napoca
Computer Science Department

Filtre digitale liniare

Obiectiv: convolutia imaginii *f(i,j)* cu functia de filtrare *h(i,j)*

In spatiul Real (coordonate imagine):

$$g(i,j) = h(i,j) \odot f(i,j)$$

In spatiul Fourier (spatiul frecventelor) $\Rightarrow$ teorema convolutiei:

$$G(k,l) = H(k,l)\,F(k,l)$$

Operatia de filtrare este controlata de forma functiei de filtrare:

h(i,j) in spatiul Real

H(k,l) in spatiul Fourier

Cele doua modalitati de aplicare sunt identice matematic, dar difera d.p.d.v. al efortului de calcul.

Technical University of Cluj Napoca
Computer Science Department

Convolutia in spatiul Fourier

$$g(i,j) = F^{-1}\{H(k,l)F(k,l)\}$$

Efortul de calcul:

2x DFT + 1x inmultire complexa (operatii cu virgula)

daca *H(k,l)* se deduce din *h(i,j)* $\Rightarrow$ 1x DFT

Costul de calcul nu depinde de tipul (dimensiunea) filtrului: *h(i,j)*

Se poate aplica doar pt. operatii de filtrare liniare

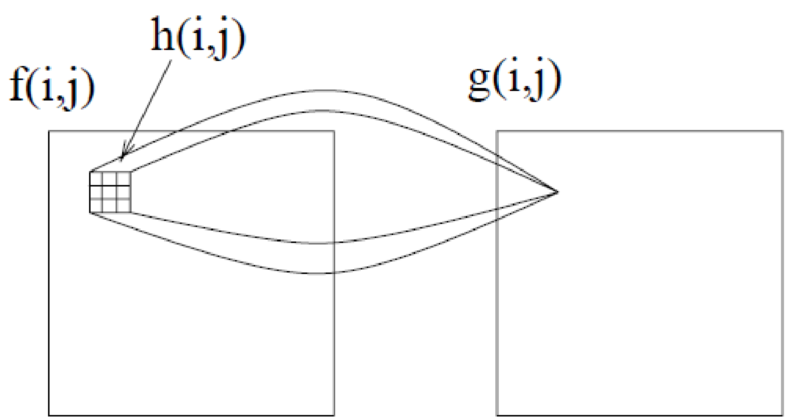
NU se poate aplica pt. operatii de filtrare ne-liniare

Technical University of Cluj Napoca
Computer Science Department

Convolutia in spatiul Real

Pentru un filtru *h(i,j)* de dimensiune MxM se defineste ca:

$$g(i,j) = \sum_{m,n=-M/2}^{M/2} h(m,n) f(i-m, j-n)$$



Schema “shift & multiply”

Operatii pe tip intreg / byte

Costul de calcul ~ M^2 (depinde de dimensiunea filtrului)

Technical University of Cluj Napoca
Computer Science Department

Convolutie in spatiul Real sau Fourier?

Costul convolutiei in spatiul Real depinde de dimensiunea Filtrului

Costul convolutiei in spatiul Fourier – fix (depinde de costul DFT si x)

In functie de tipul platformei hardware exista o limita a dimensiunii filtrului sub care care aplicare convolutiei in spatiul Real este mai aficienta decat aplicarea convolutiei in spatiul Fourier

Daca se depaseste aceasta limita pentru dimensiunea filtrului, aplicarea convolutiei in spatiul Fourier este mai eficienta (cost de calul mai scazut).

Filtrarea in spatiul Fourier

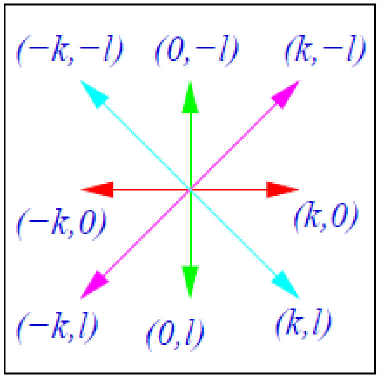
- Modifica transformata Fourier a imaginii de intrare: *F(k,l)*
- Depinde de forma filtrului: *H(k,l)*

Imaginea de intrare / iesire ⇒ **Reale** (majoritatea aplicatiilor)

Transformata Fourier complexa a imaginilor reale ⇒ proptietati de simetrie:

- Partea **Reala simetrica**

- Partea **Imaginara anti-simetrica**



Pentru ca imaginea de iesire sa fie Reala ⇒ filtrul Fourier trebuie sa respecte proprietatile de simetrie

Filtre trece jos (low-pass)

- Permit trecerea neatenuata a frecventelor spatiale joase
- Atenueaza sau blocheaza trecerea frecventelor inalte
- Folosite in reducerea zgomotului din imagini

Filtrul trece jos ideal $\Rightarrow$ blocheaza toate frecventele mai mari decat o frecventa de taiere (cut-off frequency) ω_0:

$$
H(k,l) = \begin{cases} 1 & k^2 + l^2 \leq w_0^2 \\ 0 & \text{else} \end{cases}
$$

Filtrul trece jos in spatiul Real:

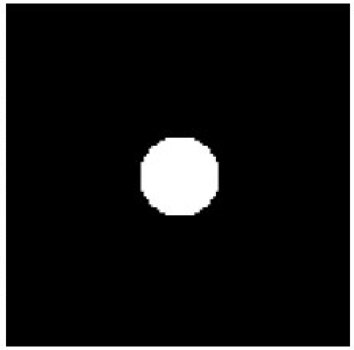
$$
h(i,j) = \frac{J_1(r/w_0)}{r/w_0}
$$

$$
r^2 = i^2 + j^2.
$$

Technical University of Cluj Napoca
Computer Science Department

Filtrul trece jos ideal - exemplu

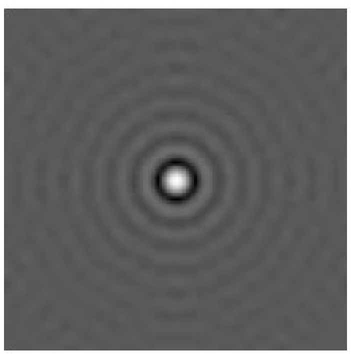
Input image



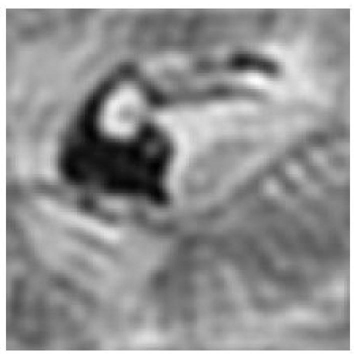
Low-pass filter

Dimensiune imagine: 128 x 128

Frecv. taiere: 15 pixeli



Real space filter



Filtered Image

Imaginea rezultata după aplicarea filtrului trece-jos ideal are aspect de undă circulară la variatii bruste ale intensitatii din imagine $\Rightarrow$ nefolositor

Filtrul trece-jos Gaussian

$$H(k,l) = \exp\left(-\frac{w}{w_0}\right)^2 \qquad w^2 = k^2 + l^2$$

Latimea filtrului ω_0 ($k^2 + l^2 = \omega_0^2$) este valoarea pt. care gaussianul H(k,l) este 1/e.

In spatiul Real se obtine tot un filtru gaussian:

$$h(i,j) = \frac{\pi}{w_0^2}\exp\left(-\pi^2 w_0^2 r^2\right) \qquad r^2 = i^2 + j^2.$$

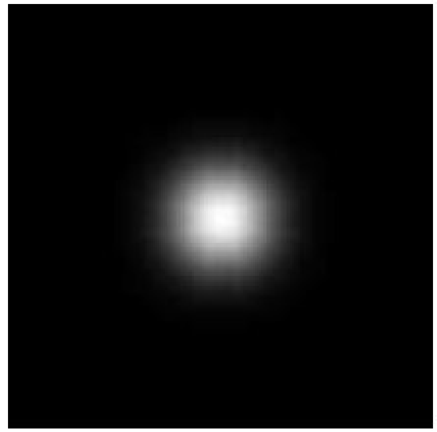
Filtrul este infinit atat in spatiul Fourier cat si in cel Real ⇒ atenueaza frecventele inalte, fara sa le inlature complet ca si filtrul ideal.

Efecte: netezirea imaginii (“smoothing”), fara efecte de unda circulara (“ringing”).

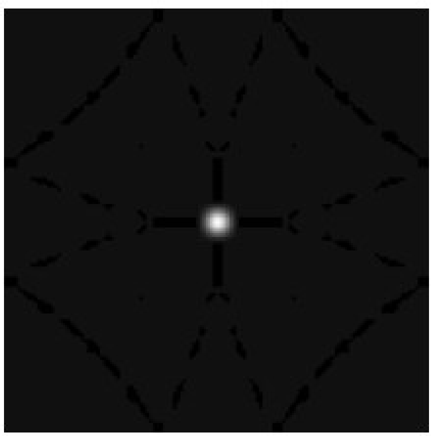
De obicei se aplica inainte de o segmentare (bazata pe regiuni sau pe muchii) pentru eliminarea zgomotului (frecv. inalte)

Filtrul trece-jos Gaussian - exemplu

(a) Input image



(b) Low-pass filter



(c) Filter in real space.

(d) Filtered Image

Dimensiune imagine:
128 x 128

ω_0 = 30 pixeli

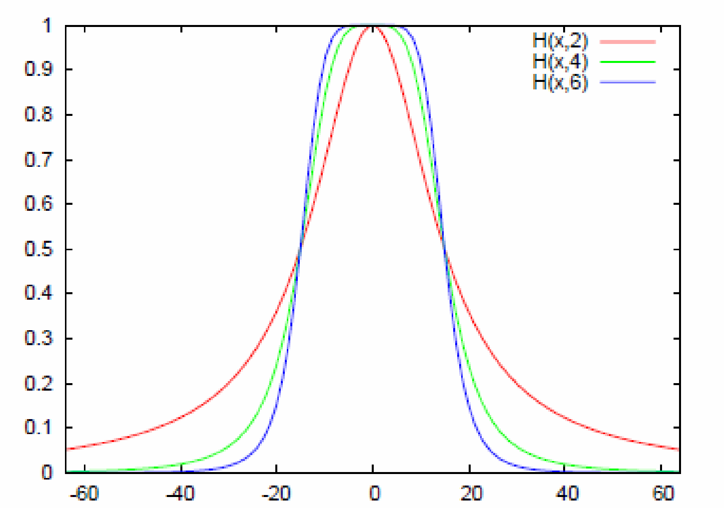
Alte fitre trece-jos

Filtrul Butterworth

$$H(k,l) = \frac{1}{1 + (\frac{w}{w_0})^n}$$

ω_0 - punctul in care H(k,l) = ½

n – ordinul filtrului



Filtrul trapezoidal

$$H(k,l) \begin{cases} = 1 & \text{for } w < w_0 \\ = \frac{(w - w_1)}{(w_0 - w_1)} & \text{for } w_0 \le w \le w_1 \\ = 0 & \text{for } w > w_1 \end{cases}$$

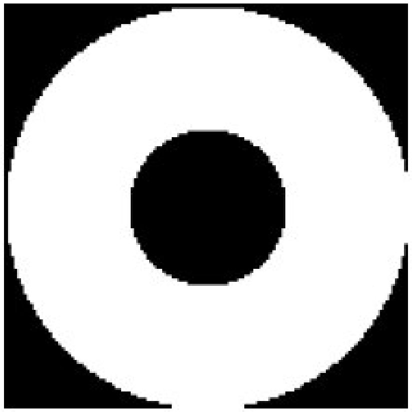
Introduce efect de unda circulara (mai puternic decat cel Gaussian sau Butherworth dar mai slab decat cel Ideal)

Technical University of Cluj Napoca
Computer Science Department

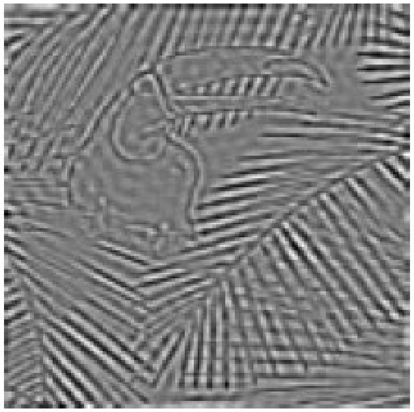
Filtre trece-sus (high-pass)

- Permit trecerea neatenuata a frecventelor spatiale inalte
- Atenueaza sau blocheaza trecerea frecventelor joase
- Folosit la accentuarea frecventelor inalte (muchii)

Filtrul trece sus ideal $\Rightarrow$ blocheaza toate frecventele mai mici decat o frecventa de taiere (cut-off frequency) ω_0:



Filter



Output Image

$\omega_0 = 25$

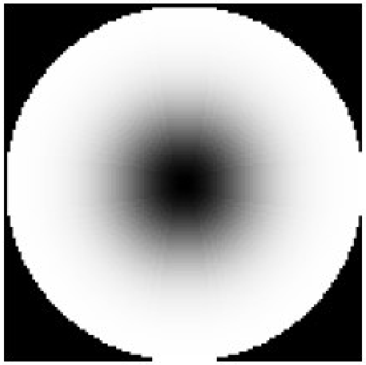
Imaginea rezultata după aplicarea filtrului trece-sus ideal are aspect de undă circulară $\Rightarrow$ nefolositor

Filtru trece-sus Gaussian

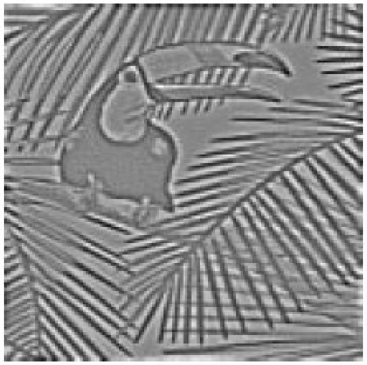
- Realizeaza o reducere gradata a frecventelor joase
- Frecventele inalte trec prin filtru nealterate

$$H(k,l) = 1 - \exp\left(-\frac{w}{w_0}\right)^2 \qquad w^2 = k^2 + l^2,$$

In spatiul Real $\Rightarrow$ filtru *h(i,j)* cu variatie gradata $\Rightarrow$ accentuarea frecventelor inalte (ex: muchii) fara fect de unda circulara (“ringing”)



Filter



Output Image

$\omega_0 = 25$

In practica, acest filtru se combina cu un Gaussian trece jos $\Rightarrow$ Difference of Gaussian (DOG)

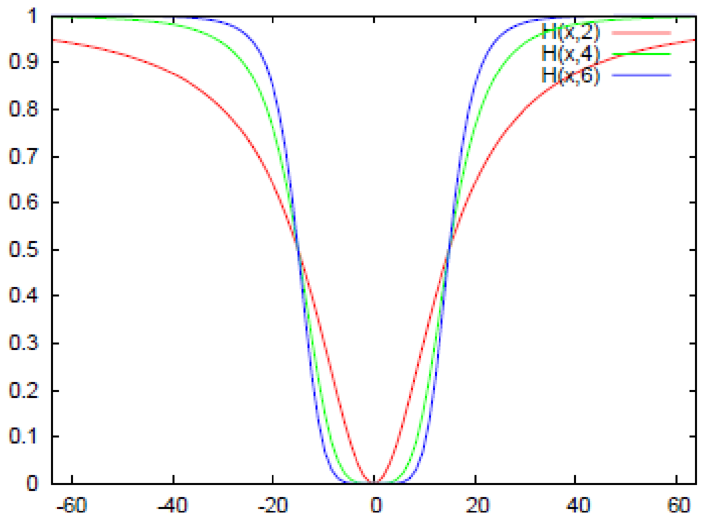
Alte filtre trece-sus

Filtrul Butterworth trece-sus

$$H(k,l) = 1 - \frac{1}{1 + \left(\frac{w}{w_0}\right)^n} = \frac{1}{1 + \left(\frac{w_0}{w}\right)^n}$$

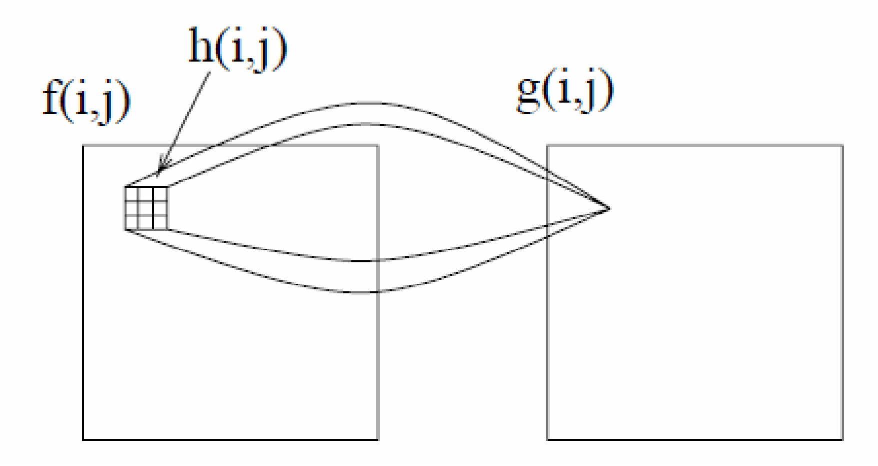
ω_0 - punctul in care H(k,l) = ½

n – ordinul filtrului



Filtrarea in spatiul Real

Filtrul este specificat in spatiul Real printr-o masca h(i,j) de dimensiune finita: 3x3, 5x5, 7x7 ….



Elementele h(i,j) sunt reale (intregi sau flotante)

Pentru masti mai mari de 7x7 folosirea filtrarii in spatiul Fourier poate fi mai rapida

Filtre de medie in spatiul Real

Inlocuiesc fiecare pixel cu media vecinilor ⇒ efect low-pass (reducere a zgomotului)

Exemple:

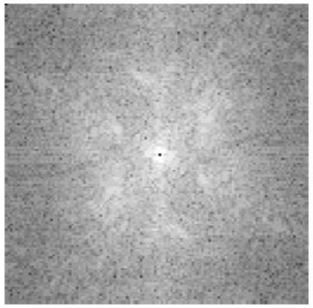
5 Pixel Average

$$\begin{matrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{matrix}$$

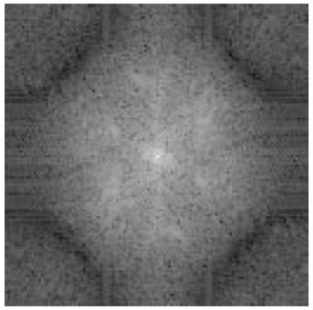
9 Pixel Average

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

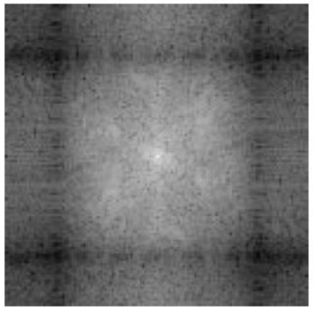

Input Image


Fourier Transform


5 point ave


Fourier Transform


9 point ave


Fourier Transform

Technical University of Cluj Napoca
Computer Science Department

Derivata in spatiul real

Derivata unei functii continue 1D:

$$\frac{d f(x)}{\mathrm{d}x} = \lim_{\delta \to 0} \frac{f(x+\delta) - f(x)}{\delta}$$

Pentru cazul discret (δ = 1):

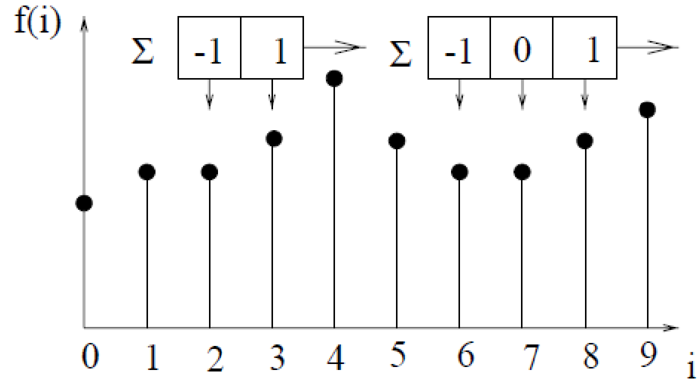
$$\frac{\mathrm{d} f(i)}{\mathrm{d}i} = f(i+1) - f(i)$$

Procesul de derivare se poate exprima printr-o operatie de convolutie:

$$\frac{\mathrm{d} f(i)}{\mathrm{d}i} = [-1 \quad 1] \odot f(i)$$

Pentru (δ = 2):

$$\frac{\mathrm{d} f(i)}{\mathrm{d}i} = [-1 \quad 0 \quad 1] \odot f(i)$$

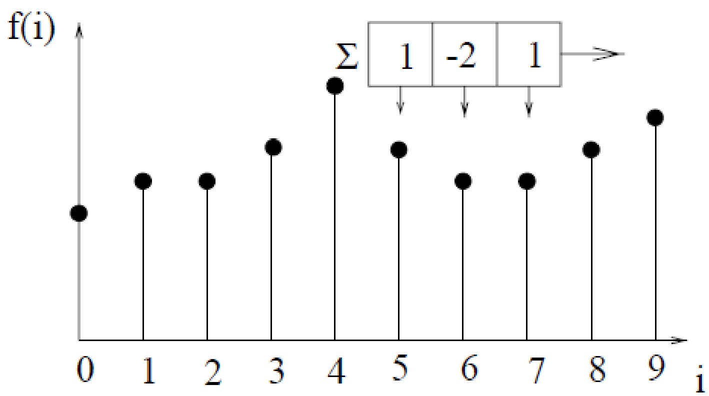


Derivata de ordin 2

Derivata de ordin 2 a unei functii discrete 1D:

$$\frac{d^2 f(i)}{di^2} = f(i+1) - 2f(i) + f(i-1)$$

Sau exprimata prin convolutie: $$\frac{d^2 f(i)}{di^2} = [1 \quad -2 \quad 1] \odot f(i)$$



Datorita faptului ca operatia de convolutie este liniara, remarcam ca:

$$[1 \quad -2 \quad 1] = [-1 \quad 1] \odot [-1 \quad 1]$$

Derivate 2D

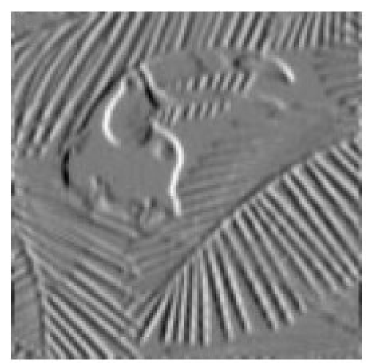
In cazul bi-dimenional (imagini reale) avem:

$$\frac{\partial f(i,j)}{\partial i} = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \odot f(i,j) \qquad \frac{\partial f(i,j)}{\partial j} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \odot f(i,j)$$

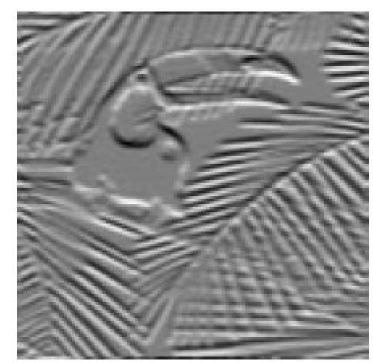
Pentru atenuarea zgomotului se practica medierea derivatelor pe 3 linii / coloane:

$$\frac{\partial f(i,j)}{\partial i} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \odot f(i,j) \qquad \frac{\partial f(i,j)}{\partial j} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \odot f(i,j)$$

Efecte: evidentierea muchiilor verticale / orizontale



x-differential



y-differential

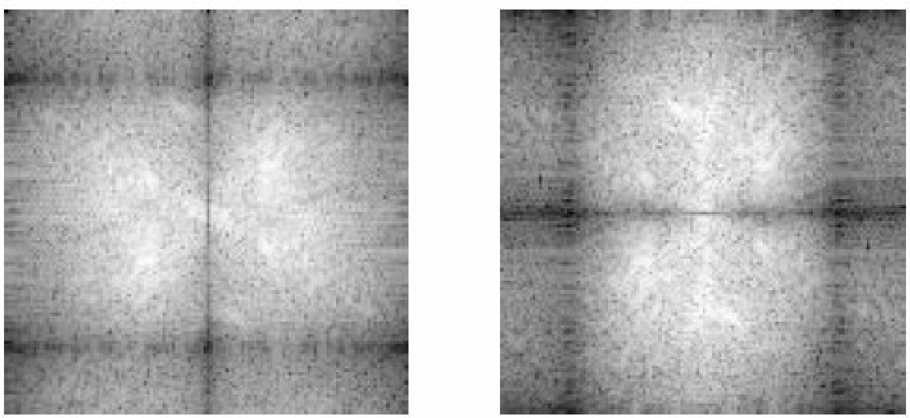
Derivata in spatiul Fourier

Proprietatile transformatei Fourier:

$$F\left\{\frac{\partial f(x,y)}{\partial x}\right\} = i2\pi u F(u,v) \qquad F\left\{\frac{\partial f(x,y)}{\partial y}\right\} = i2\pi v F(u,v)$$

$\Rightarrow$ Derivarea in spatiul Fourier $\equiv$ inmultire cu $i2\pi u$ sau $i2\pi v$

$\Rightarrow$ Efect high-pass (accentuarea frecventelor inalte)



x-differential (FT) y-differential (FT)

Pe langa liniile “negre” (zero) orizontale si verticale centrale mai apar linii de “zero” secundare datorita efectului de mediere).

Derivata de ordin 2

$$\frac{\partial^2 f(i,j)}{\partial i^2} = [1 \quad -2 \quad 1] \odot f(i,j) \qquad \frac{\partial^2 f(i,j)}{\partial j^2} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \odot f(i,j)$$

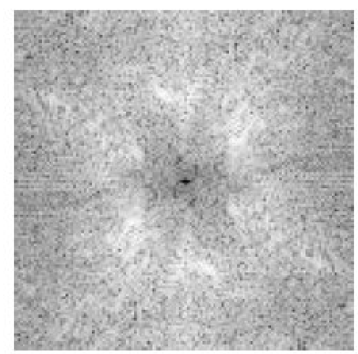
Laplacianul:

$$\nabla^2 f(i,j) = \frac{\partial^2 f(i,j)}{\partial i^2} + \frac{\partial^2 f(i,j)}{\partial j^2} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \odot f(i,j)$$

Proprietatile transformatei Fourier:

$$F\left\{\nabla^2 f(x,y)\right\} = -(2\pi w)^2 F(u,v) \qquad w^2 = u^2 + v^2,$$

Efecte $\Rightarrow$ accentuarea muchiilor din imagine in toate directiile



Technical University of Cluj Napoca
Computer Science Department

Variatii ale Laplacian-ului

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$ $\Rightarrow$ Laplacian mai putin sensibil la zgomot

Accentuarea muchiilor: scaderea din imagine a Laplacian-ului

$$f(i,j) - \nabla^2 f(i,j) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \odot f(i,j)$$

Input image

Edge Enhanced

Utilizarea filtrelor liniare

Filtre low-pass: netezirea imaginilor, diminuarea efectelor zgomtelor (se folosesc de obicei inaintea aplicarii detectorilor de puncte de muchie)

Filte high-pass (filtre derivative): accentuarea frecventelor inalte (muchii).

Filtrele pot fi combinate (ex: formarea de filtre trece banda - band-pass).

Datorita caracterului liniar, combinarea se poate face prin inmultire in spatiu Fourier si convolutie in spatiul Real.

Filtre neliniare in spatiul Real

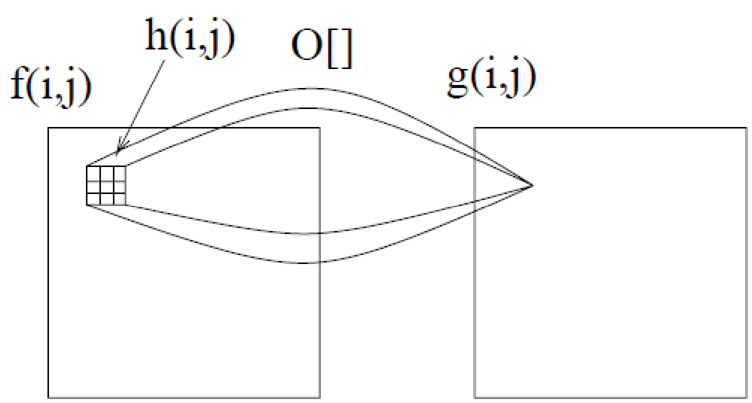
$$g(i,j) = O_{m,n \in w}[h(m,n) f(i-m, j-n)]$$

Domeniul de definite al mastii *h(m,n)* dat de *w.*

Operatia de filtrare este definita de masca *h(i,j)* si operatorul *O[]*.

La majoritatea filtrelor neliniare avem: $h(i,j) = 1 \quad i,j \in w$

operatia de filtrare fiind controlata de operatorul *O[]*



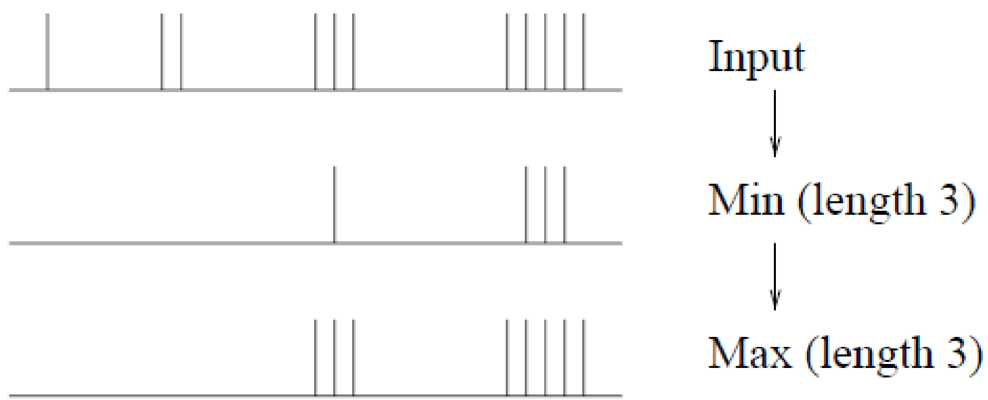
Filtre “shrink & expand”

Shrink = ingustare; Expand = largire

O[] = Min[] $\Rightarrow$ shrink: obiecte luminoase (alb = obiect) sunt reduse in dimensiune cu un ordin de marime egal cu dimensiunea filtrului

O[] = Max[] $\Rightarrow$ expand: obiecte luminoase sunt extinse in dimensiune cu un ordin de marime egal cu dimensiunea filtrului

$\Rightarrow$ Cele doua filtre sunt folosite in pereche pe imagini binare pentru eliminarea regiunilor mici / izolate



Obs: aplicarea filtrelor nu este comutativa:

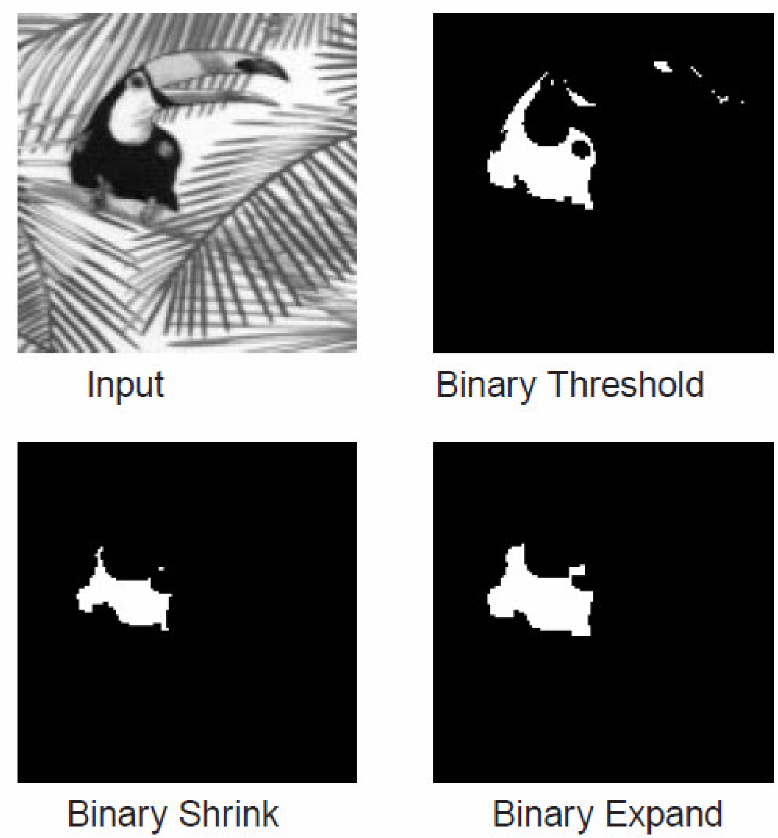
$$E[S[f(i,j)]] \neq S[E[f(i,j)]]$$

Filtre shrink/expand 2D

Exemplu: fereastra 1D de dimensiune 3 (din exemplul anterior) ⇒ fereastra 2D de dimensiune 3x3

Efect: eliminarea obiectelor (obiect = pixel alb) mici/izolate dintr-o imagine binara

Exemplu filtrare shrink&expand cu un filtru de dimensiune 4x4:



Filtrul Threshold Averge

Compara fiecare pixel cu media vecinilor (exclusiv valoarea lui) si netezeste (inlocuieste pixelul curent) doar daca diferenta este semnificativa.

Pentru fiecare pixel se construieste multimea:

$$A = \sum_{m,n=-M/2}^{M/2} h(m,n) f(i-m, j-n)$$

Ex. - pt. un filtru 3x3: $h(i,j) = \begin{bmatrix} k & k & k \\ k & 0 & k \\ k & k & k \end{bmatrix}$ $\quad k = 1/(M^2 - 1) = 0.125$

$$g(i,j) \begin{array}{ll} = A & |A - f(i,j)| > T \\ f(i,j) & \text{else} \end{array}$$

Stabilirea T: “trial and error” (~ fractiune din (Max(f) - Min(f))

Filtrul median

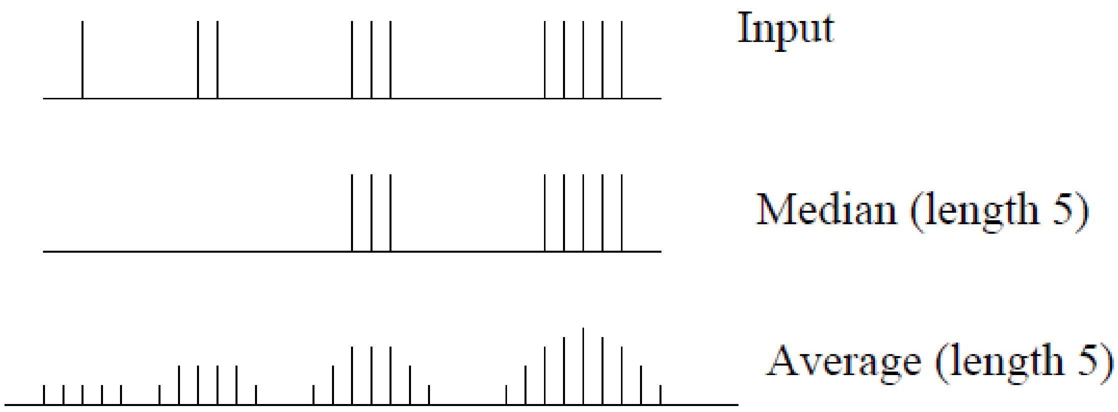
$$O[] = \text{Median}[]$$

Median := elementul de mijloc dintr-o multime sortata

Exemplu:

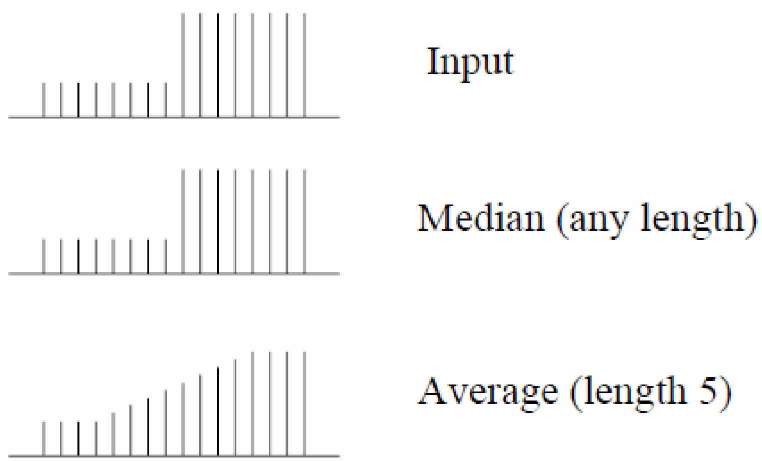
$$f(i) = 61, 10, 9, 11, 9 \qquad \text{Median}[f(i)] = 10$$

In cazul 1D filtrul median elimina toate trasaturile/obiectele cu dimensiune mai mica decat M/2+1, pastrand toate celelalte trasaturi



Proprietati (filtrul median)

Netezire fara alterearea punctelor de muchie (low-pass selectiv:



In cazul 2D elimina toate trasaturile (obiectele) cu dimensiunea ("aria") mai mica de $M^2/2-1$, pastrandu-le pe toate celelalte

3×3 Median

5×5 Median