

Procesarea Imagineilor Digitale

Curs - Reducerea zgomotului - Filtrare trece-jos

Universitatea "Transilvania" din Brașov

Filtrare

Filtrare - păstrarea anumitor elemente/trăsaturi și păstrarea altora

Din filtrarea semnalelor cu transformata Fourier:

- ➊ Filtrare trece-jos - *low-pass filtering* - păstrarea frecvențelor joase și eliminarea /reducerea celor înalte

Filtrare

Filtrare - păstrarea anumitor elemente/trăsaturi și păstrarea altora

Din filtrarea semnalelor cu transformata Fourier:

- ① Filtrare trece-jos - *low-pass filtering* - păstrarea frecvențelor joase și eliminarea /reducerea celor înalte

- ② Filtrare trece-sus - *high-pass filtering* - păstrarea frecvențelor înalte și eliminarea /reducerea celor joase

Filtrarea trece-jos pentru reducerea zgomotului

- **Zgomot:** - Perturbații în conținutul datelor imagistice.
- **Surse:**
 - Sistemele de achiziție: tip dispozitiv, timp de expunere etc..
 - Rețele de transmisie: interferențe cu alte semnale.

Zgomot - Filtrarea trece-jos pentru reducerea zgomotului

Tipuri de zgomot

- **aleatoriu:** - variații aleatorii ale intensității/colorii .
- **cu pattern fix:** - imperfecțiuni ale senzorilor - **hot pixels, dead pixels.**
- **bandă:** - depinde de structura hard a dispozitivului imagistic și se produce în timpul citirii datelor din senzori digitali.
- **periodic:** - cauzat mai ales de interferențele electrice și electromagnetice.

Filtrarea zgromotului - Exemplu



Filtrarea zgromotului periodic - Exemplu



Filtrarea ca efect special - Exemplu



Filtrarea liniară în domeniul spațial - modelare matematică

- **Filtrarea spațială liniară:** pentru fiecare pixel $f(x, y)$ se calculează o sumă ponderată a pixelilor vecini cu ponderile date de o mască de filtrare h .

Filtrarea liniară în domeniul spațial - modelare matematică

- **Filtrarea spațială liniară:** pentru fiecare pixel $f(x, y)$ se calculează o sumă ponderată a pixelilor vecini cu ponderile date de o mască de filtrare h .
- **Mască de filtrare** - *kernel, window, mask* -

Filtrarea liniară în domeniul spațial - modelare matematică

- **Filtrarea spațială liniară:** pentru fiecare pixel $f(x, y)$ se calculează o sumă ponderată a pixelilor vecini cu ponderile date de o mască de filtrare h .
- **Mască de filtrare** - *kernel, window, mask* -
 - definește o vecinătate pe care se realizează operația

Filtrarea liniară în domeniul spațial - modelare matematică

- **Filtrarea spațială liniară:** pentru fiecare pixel $f(x, y)$ se calculează o sumă ponderată a pixelilor vecini cu ponderile date de o mască de filtrare h .
- **Mască de filtrare** - *kernel, window, mask* -
 - definește o vecinătate pe care se realizează operația
 - de obicei de formă pătrată de dimensiune impară

Filtrarea liniară în domeniul spațial - modelare matematică

- **Filtrarea spațială liniară:** pentru fiecare pixel $f(x, y)$ se calculează o sumă ponderată a pixelilor vecini cu ponderile date de o mască de filtrare h .
- **Mască de filtrare** - *kernel, window, mask* -
 - definește o vecinătate pe care se realizează operația
 - de obicei de formă pătrată de dimensiune impară
 - coeficienții măștii definesc tipul de filtrare (low-pass / high-pass)

Filtrarea liniară în domeniul spațial - modelare matematică

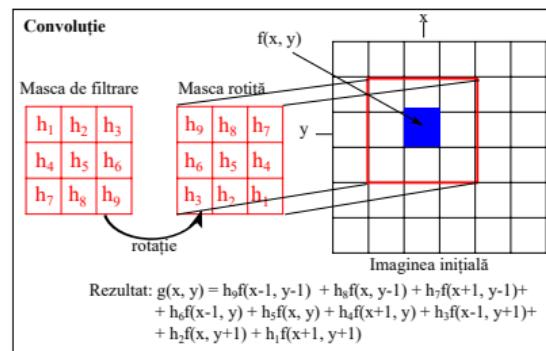
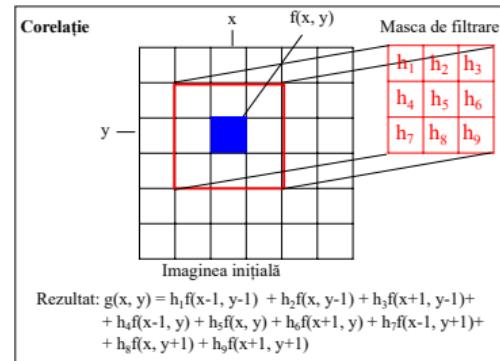
Filtrul: $h = (h(i,j), i = \overline{-k, k}, j = \overline{-k, k})$

- Corelație:

$$g(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k h(i, j) f(x + i, y + j)$$

- Convoluție

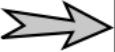
$$g(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k h(i, j) f(x - i, y - j)$$



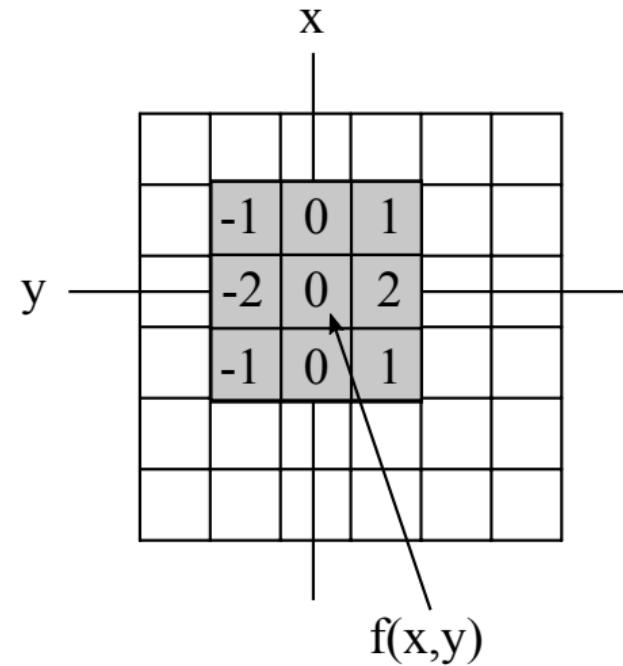
Convoluție - Exemplu

h

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

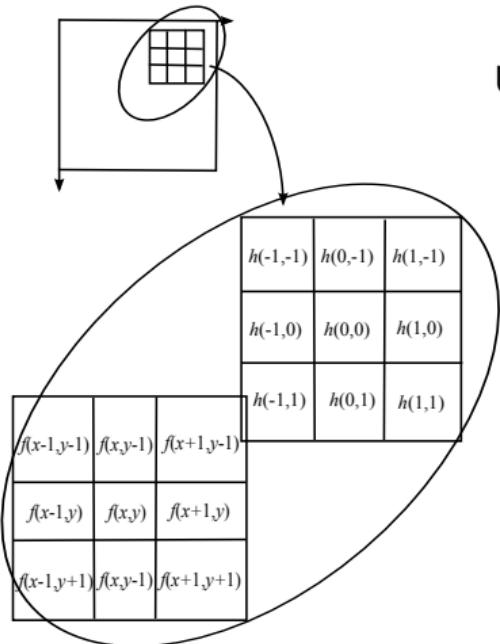


-1	0	1
-2	0	2
-1	0	1



$$g(x,y) = f(x+1,y-1) - f(x-1,y-1) + 2*f(x+1,y) - 2*f(x-1,y) + f(x+1,y+1) - f(x-1,y+1)$$

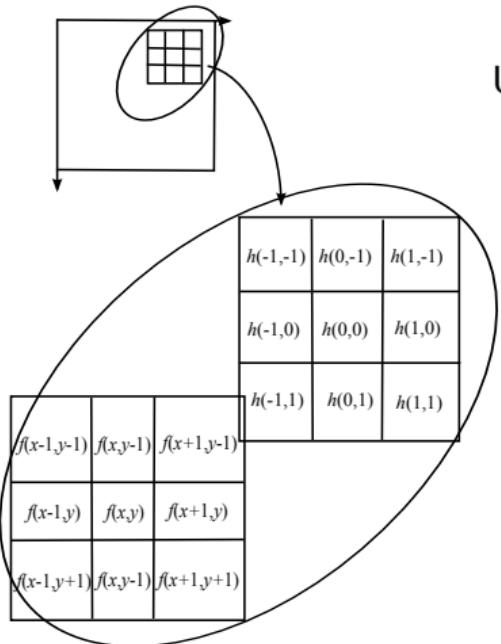
Filtrarea liniară a imaginii



Un filtru liniar:

- Se parurge imaginea inițială pixel cu pixel (ignorând o ramă de dimensiune k)

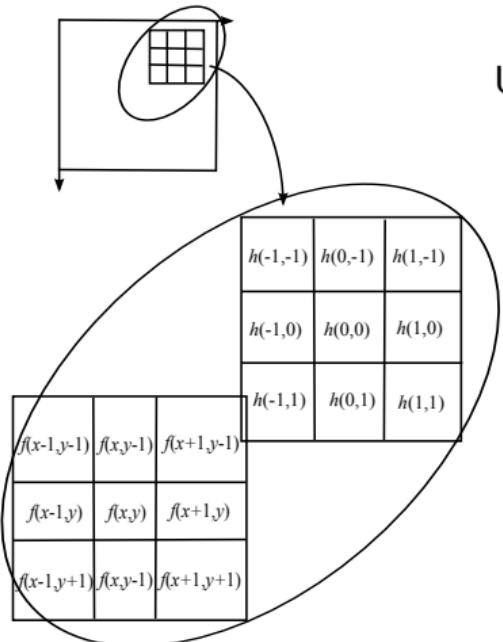
Filtrarea liniară a imaginii



Un filtru liniar:

- Se parcurge imaginea inițială pixel cu pixel (ignorând o ramă de dimensiune k)
- Se plasează masca pe imagine, centrată pe pixelul curent (x, y)

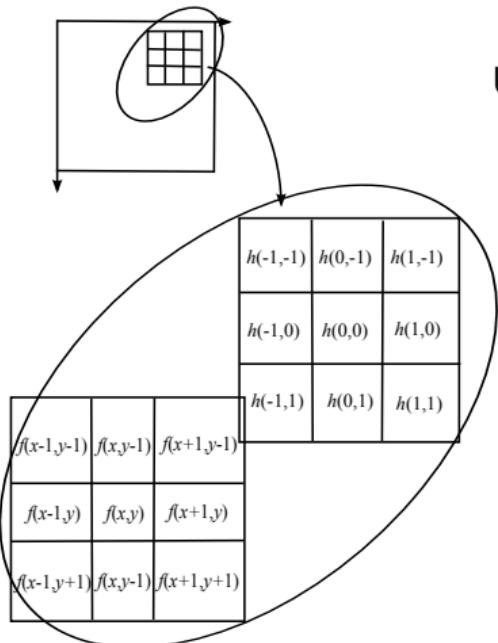
Filtrarea liniară a imaginii



Un filtru liniar:

- Se parcurge imaginea inițială pixel cu pixel (ignorând o ramă de dimensiune k)
- Se plasează masca pe imagine, centrată pe pixelul curent (x, y)
- Se calculează media ponderată a pixelilor din vecinătatea delimitată de mască, luând ca ponderi coeficienții din mască

Filtrarea liniară a imaginii



Un filtru liniar:

- Se parcurge imaginea inițială pixel cu pixel (ignorând o ramă de dimensiune k)
- Se plasează masca pe imagine, centrată pe pixelul curent (x, y)
- Se calculează media ponderată a pixelilor din vecinătatea delimitată de mască, luând ca ponderi coeficienții din mască
- Se plasează în imaginea rezultat pe poziția (x, y) valoare obținută.

Filtrarea în domeniul spațial - Exemplu

1	2	2	4	3	2
1	3	5	5	7	6
2	1	4	5	4	6
3	3	2	1	2	2
3	2	2	1	1	2
2	1	1	3	3	2

Imaginea inițială

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

Masca de filtrare

Rezultat

Filtrarea în domeniul spațial - Exemplu

-1	0	1	2	4	3	2
-2	0	2	5	5	7	6
-1	0	1	4	5	4	6
2	1	1	4	5	4	6
3	3	2	1	2	2	
3	2	2	1	1	2	
2	1	1	3	3	2	

$$\begin{aligned} g(1,1) = & -1*1 + 0*2 + 1*2 + (-2)*1 \\ & + 0*3 + 2*5 + (-1)*2 + 0*1 + 1*4 = \\ & = 11 \end{aligned}$$

Rezultat

11

Filtrarea în domeniul spațial - Exemplu

-1	0	1	2	4	3	2
-2	0	2	1	4	3	2
1	3	5	5	7	6	
-1	0	1	4	5	4	6
2	1	4	5	4	6	
3	3	2	1	2	2	
3	2	2	1	1	2	
2	1	1	3	3	2	

$$g(1,1) = -1*1 + 0*2 + 1*2 + (-2)*1 + 0*3 + 2*5 + (-1)*2 + 0*1 + 1*4 = 11$$

$$g(2,1) = -1*2 + 0*2 + 1*4 + (-2)*3 + 0*5 + 2*5 + (-1)*1 + 0*4 + 1*5 = 10$$

1	-1	0	1	2	4	3	2
1	-2	0	2	1	4	3	6
1	3	5	5	7	6		
2	-1	0	1	4	5	4	6
3	3	2	1	2	2		
3	2	2	1	1	2		
2	1	1	3	3	2		

Rezultat

11 10

Filtrarea în domeniul spațial - Exemplu

-1	0	1	2	4	3	2
-2	0	2	5	5	7	6
-1	1	3	5	4	6	
2	0	1	4	5	4	6
3	3	2	1	2	2	
3	2	2	1	1	2	
2	1	1	3	3	2	

$$g(1,1) = -1*1 + 0*2 + 1*2 + (-2)*1 + 0*3 + 2*5 + (-1)*2 + 0*1 + 1*4 = \\ = 11$$

1	-1	0	2	1	4	3	2
1	-2	0	2	5	5	7	6
2	-1	0	1	5	4	6	
3	3	2	1	2	2		
3	2	2	1	1	2		
2	1	1	3	3	2		

$$g(2,1) = -1*2 + 0*2 + 1*4 + (-2)*3 + 0*5 + 2*5 + (-1)*1 + 0*4 + 1*5 = \\ = 10$$

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

$$g(3,1) = -1*2 + 0*4 + 1*3 + (-2)*5 + 0*5 + 2*7 + (-1)*4 + 0*5 + 1*4 = \\ = 5$$

Rezultat

11	10	5				

Filtrarea în domeniul spațial - Exemplu

-1	0	1	2	4	3	2
-2	0	2	5	5	7	6
-1	0	1	4	5	4	6
2	1	4	5	4	6	
3	3	2	1	2	2	
3	2	2	1	1	2	
2	1	1	3	3	2	

$$\begin{aligned} g(1,1) = & -1*1 + 0*2+1*2+(-2)*1 \\ & +0*3+2*5+(-1)*2+0*1+1*4= \\ & = 11 \end{aligned}$$

1	-1	0	1	4	3	2
1	-2	0	2	5	5	7
2	-1	0	1	4	5	6
3	3	2	1	2	2	
3	2	2	1	1	2	
2	1	1	3	3	2	

$$\begin{aligned} g(2,1) = & -1*2 + 0*2+1*4+(-2)*3 \\ & +0*5+2*5+(-1)*1+0*4+1*5= \\ & = 10 \end{aligned}$$

1	2	-1	0	1	3	2
1	3	-2	0	2	5	6
2	1	4	-1	5	4	6
3	3	2	1	2	2	
3	2	2	1	1	2	
2	1	1	3	3	2	

$$\begin{aligned} g(3,1) = & -1*2 + 0*4+1*3+(-2)*5 \\ & +0*5+2*7+(-1)*4+0*5+1*4= \\ & = 5 \end{aligned}$$

1	2	2	-1	0	1	2
1	3	5	-2	0	2	6
2	1	4	-1	5	4	6
3	3	2	1	2	2	
3	2	2	1	1	2	
2	1	1	3	3	2	

$$\begin{aligned} g(4,1) = & -1*4 + 0*3+1*2+(-2)*5 \\ & +0*7+2*6+(-1)*5+0*4+1*6= \\ & = 1 \end{aligned}$$

Rezultat

11	10	5	1			

Filtrarea în domeniul spațial - Exemplu

-1	0	1	2	4	3	2
-2	0	2		4	3	2
1	3	5	5	7	6	
-1	0	1	4	5	4	6
2	1	4		5	4	6
3	3	2	1	2	2	
3	2	2	1	1	2	
2	1	1	3	3	2	

$$g(1,1) = -1*1 + 0*2+1*2+(-2)*1 +0*3+2*5+(-1)*2+0*1+1*4= \\ = 11$$

1	-1	0	2	1	4	3
1	-2	0	2	1	4	3
1	3	5	5	7	6	
2	-1	0	4	1	5	4
2	1	4		5	4	6
3	3	2	1	2	2	
3	2	2	1	1	2	
2	1	1	3	3	2	

$$g(2,1) = -1*2 + 0*2+1*4+(-2)*3 +0*5+2*5+(-1)*1+0*4+1*5= \\ = 10$$

1	2	-1	0	1	4	3
1	3	5	5	7	6	
2	1	4	-1	0	1	6
2	1	4	5	4	1	6
3	3	2	1	2	2	
3	2	2	1	1	2	
2	1	1	3	3	2	

$$g(3,1) = -1*2 + 0*4+1*3+(-2)*5 +0*5+2*7+(-1)*4+0*5+1*4= \\ = 5$$

1	2	2	-1	0	3	1	2
1	3	5	-2	0	2		
2	1	4	-1	0	1	6	
2	1	4	5	4	1	6	
3	3	2	1	2	2		
3	2	2	1	1	2		
2	1	1	3	3	2		

$$g(4,1) = -1*4 + 0*3+1*2+(-2)*5 +0*7+2*6+(-1)*5+0*4+1*6= \\ = 1$$

1	2	2	4	3	2
-1	0	1	5	5	7
-2	0	1	2	4	5
-1	0	1	2	1	2
3	2	2	1	1	2
2	1	1	3	3	2

$$g(1,2) = -1*1 + 0*3+1*5+(-2)*2 +0*7+2*6+(-1)*5+0*4+1*6= \\ = 7$$

11	10	5	1			
7						

Rezultat

Filtrarea în domeniul spațial - Exemplu

-1	0	2	1	2	4	3	2
-2	0	2			5	7	6
-1	0	3	5		5	7	6
2	1	4	5	4	6		
3	3	2	1	2	2		
3	2	2	1	1	2		
2	1	1	3	3	2		

$$g(1,1) = -1*1 + 0*2 + 1*2 + (-2)*1 + 0*3 + 2*5 + (-1)*2 + 0*1 + 1*4 = 11$$

1	2	2	-1	0	3	1	2
1	3	5	-2	0	2	7	6
2	1	4	-1	0	1	5	6
3	3	2	1	2	2		
3	2	2	1	1	2		
2	1	1	3	3	2		

$$g(4,1) = -1*4 + 0*3 + 1*2 + (-2)*5 + 0*7 + 2*6 + (-1)*5 + 0*4 + 1*6 = 1$$

1	-1	0	2	1	4	3	2
1	3	5	-2	0	2	5	7
2	1	4	-1	0	1	5	6
3	3	2	1	2	2		
3	2	2	1	1	2		
2	1	1	3	3	2		

$$g(2,1) = -1*2 + 0*2 + 1*4 + (-2)*3 + 0*5 + 2*5 + (-1)*1 + 0*4 + 1*5 = 10$$

1	2	2	4	3	2		
-1	0	1	5	5	7	6	
-2	0	2	4	5	4	6	
-1	0	1	2	1	2	2	
3	2	2	1	1	2		
2	1	1	3	3	2		

$$g(1,2) = -1*1 + 0*3 + 1*5 + (-2)*2 + 0*7 + 2*6 + (-1)*5 + 0*4 + 1*6 = 7$$

1	2	-1	0	1	3	2	
1	3	-2	0	2	5	7	6
2	1	-1	0	1	5	4	6
3	3	2	1	2	2		
3	2	2	1	1	2		
2	1	1	3	3	2		

$$g(3,1) = -1*2 + 0*4 + 1*3 + (-2)*5 + 0*5 + 2*7 + (-1)*4 + 0*5 + 1*4 = 5$$

1	2	2	4	3	2		
1	3	0	5	1	5	7	6
2	-2	0	4	2	5	4	6
3	3	0	2	1	1	2	2
3	2	2	1	1	2		
2	1	1	3	3	2		

$$g(2,2) = -1*3 + 0*4 + 1*5 + (-2)*1 + 0*4 + 2*5 + (-1)*3 + 0*2 + 1*1 = 8$$

Rezultat

11	10	5	1	
7	8			

etc

Ce fac cu marginile imaginii?

Filtrare în domeniul spațial - Tratarea marginilor



a)



b)



c)

d)

(a) ignorarea pixelilor de pe margine prin lăsarea lor neschimbați (măști relativ mici).

Filtrare în domeniul spațial - Tratarea marginilor



a)



c)



d)

- (a) ignorarea pixelilor de pe margine prin lăsarea lor neschimbați (măști relativ mici).
- (b) bordarea imaginii cu 0 - introduce erori (fig. b).

Filtrare în domeniul spațial - Tratarea marginilor



a)



b)



c)



d)

- (a) ignorarea pixelilor de pe margine prin lăsarea lor neschimbați (măști relativ mici).
- (b) bordarea imaginii cu 0 - introduce erori (fig. b).
- (c) repetarea prin periodicitate a valorilor pe margine (fig. c).

Filtrare în domeniul spațial - Tratarea marginilor



a)

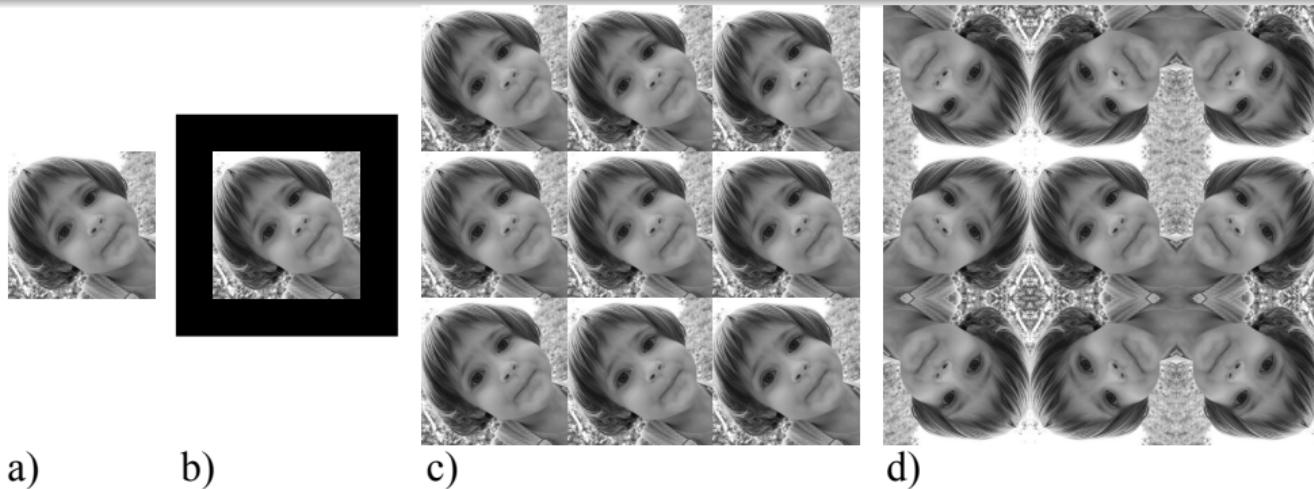
b)

c)

d)

- (a) ignorarea pixelilor de pe margine prin lăsarea lor neschimbați (măști relativ mici).
- (b) bordarea imaginii cu 0 - introduce erori (fig. b).
- (c) repetarea prin periodicitate a valorilor pe margine (fig. c).
- (d) repetarea prin oglindire a imaginii pe margini (fig. d).

Filtrare în domeniul spațial - Tratarea marginilor



- (a) ignorarea pixelilor de pe margine prin lăsarea lor neschimbați (măști relativ mici).
- (b) bordarea imaginii cu 0 - introduce erori (fig. b).
- (c) repetarea prin periodicitate a valorilor pe margine (fig. c).
- (d) repetarea prin oglindire a imaginii pe margini (fig. d).
- (e) bordarea prin replicare (*replicate padding*) - se bordează cu copii ale pixelilor de pe marginea imaginii

Filtrare liniară în domeniul spațial

Corelație versus conoluție

- operația de conoluție este comutativă și asociativă - operația de corelație NU

Filtrare liniară în domeniul spațial

Corelație versus conoluție

- operația de conoluție este comutativă și asociativă - operația de corelație NU
- comutativitatea și asociativitatea au o semnificație importantă deoarece:

$$w_1 * (w_2 * f) = (w_1 * w_2) * f$$

⇒ în loc să filtrez f succesiv cu 2 filtre, construiesc filtrul $w_1 * w_2$ și realizez o singură filtrare ⇒ crește eficiența

Filtrare liniară în domeniul spațial

Corelație versus conoluție

- operația de conoluție este comutativă și asociativă - operația de corelație NU
- comutativitatea și asociativitate au o semnificație importantă deoarece:

$$w_1 * (w_2 * f) = (w_1 * w_2) * f$$

⇒ în loc să filtrez f succesiv cu 2 filtre, construiesc filtrul $w_1 * w_2$ și realizez o singură filtrare ⇒ crește eficiența

- dacă filtrul este simetric (orizontal și vertical), atunci corelația și conoluția au același rezultat

Filtrare liniară în domeniul spațial

Corelație versus conoluție

- operația de conoluție este comutativă și asociativă - operația de corelație NU
- comutativitatea și asociativitatea au o semnificație importantă deoarece:

$$w_1 * (w_2 * f) = (w_1 * w_2) * f$$

⇒ în loc să filtrez f succesiv cu 2 filtre, construiesc filtrul $w_1 * w_2$ și realizez o singură filtrare ⇒ crește eficiența

- dacă filtrul este simetric (orizontal și vertical), atunci corelația și conoluția au același rezultat
- deseori în literatură se vorbește de conoluție, dar se realizează de fapt corelație.

Filtre separabile

Separabilitate

- O funcție 2D $G(x, y)$ se numește separabilă, dacă poate fi exprimată ca $G(x, y) = G_1(x)G_2(y)$

Filtre separabile

Separabilitate

- O functie 2D $G(x, y)$ se numește separabilă, dacă poate fi exprimată ca $G(x, y) = G_1(x)G_2(y)$
- Un filtru (*kernel*) w de dim $m \times n$ este separabil, dacă poate fi scris:

$$w = v_1 v_2^T$$

unde v_1 de dimensiuni $m \times 1$ și v_2 de dimensiuni $n \times 1$

Filtre separabile

Separabilitate

- O funcție 2D $G(x, y)$ se numește separabilă, dacă poate fi exprimată ca $G(x, y) = G_1(x)G_2(y)$
- Un filtru (*kernel*) w de dim $m \times n$ este separabil, dacă poate fi scris:

$$w = v_1 v_2^T$$

unde v_1 de dimensiuni $m \times 1$ și v_2 de dimensiuni $n \times 1$

- Produsul unui vector coloană cu un vector linie dă același rezultat ca o convoluție între cei doi vectori.

$$v_1 v_2^T = v_1 * v_2^T$$

Filtre separabile

Separabilitate - Avantaj - presupunem $w = v_1 v_2^T$, v_1 de dim $m \times 1$, v_2 de dim $n \times 1$

$$w * f = (v_1 * v_2^T) * f = v_1 * (v_2^T * f) = v_2^T * (v_1 * f)$$

Care este avantajul?

Filtre separabile

Separabilitate - Avantaj - presupunem $w = v_1 v_2^T$, v_1 de dim $m \times 1$, v_2 de dim $n \times 1$

$$w * f = (v_1 * v_2^T) * f = v_1 * (v_2^T * f) = v_2^T * (v_1 * f)$$

Care este avantajul?

Reducerea complexității de calcul!

Filtrarea trece-jos

- **Denumire:** - din reprezentarea în domeniul frecvență.
- **Reprezentă:** - eliminarea/atenuarea frecvențelor înalte și păstrarea frecvențelor joase.
- **Scop:** - reducerea zgomotului.
- **Efect:** - netezire (blurr), atenuarea contururilor.
- **Cerință:** - elementele măștii de filtrare trebuie normalizeaza cu suma coeficienților, astfel încât să satisfacă

$$\sum_{i=-k}^k \sum_{j=-k}^k h(i,j) = 1$$

Filtre trece-jos - Tipuri

- ① Filtre de mediere: filtrul medie aritmetică, filtrul binomial, filtrul Gaussian.
- ② Filtre bazate pe statistici de ordine: filtrul median, filtrul min, filtrul max, filtrul *mid-point*.

Filtrul medie aritmetică - *Box Filter / Average Filter*

$$\frac{1}{9} \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

$$\frac{1}{10} \begin{matrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{matrix}$$

- **Elementele măştii:** $h(i,j) = 1, i = -k, k, j = -k, k$.

- **Coeficientul măştii:** $c = 1/(2 * k + 1)^2$ astfel încât

$$\sum_{i=-k}^k \sum_{j=-k}^k c \cdot h(i,j) = 1$$

- **Variantă:** - creșterea ponderii pixelului central.

- **Separabilitate:** este separabil

Filtrul medie aritmetică - Exemplu

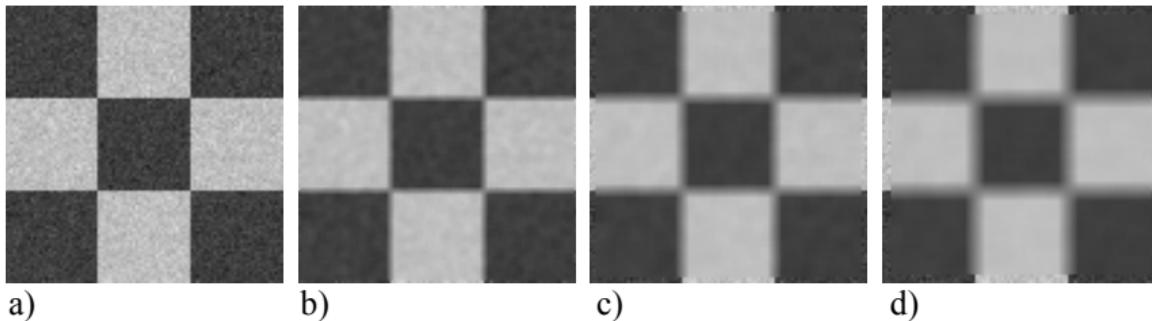
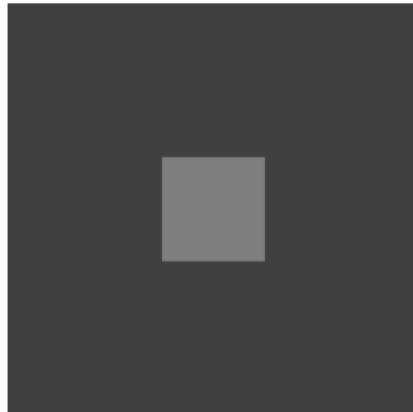


Figure: a) Figură de test sintetică alterată cu zgomot gaussian cu $\sigma = 15$; Imagine filtrată cu un filtru medie aritmetică b) 3×3 ; c) 5×5 ; d) 7×7 .

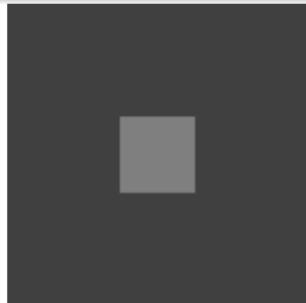
Exercițiu

Se consideră imaginea de dimensiuni 64×64 , în care pătratul are dimensiunea 16×16 și tonul de gri 127. Fundalul are tonul de gri 64:

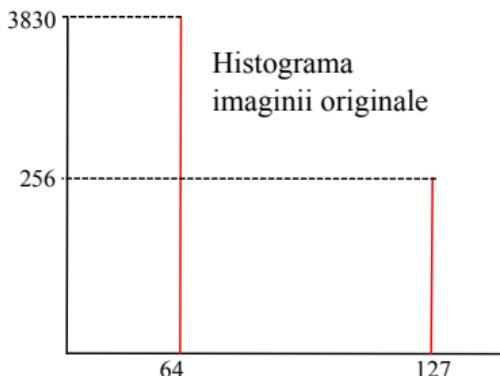


- ① Cum arată histograma?
- ② Cum arată imaginea după aplicarea unui filtru *mean* 3×3 ?
- ③ Cum arată histograma imaginii după filtrare?

Exercițiu - Rezolvare



Imagine originală



Fragment din imaginea originală

64	64	64	64	64	64	64	64	64	64
64	64	64	64	64	64	64	64	64	64
64	64	64	64	64	64	64	64	64	64
64	64	64	127	127	127	127	127	127	127
64	64	64	127	127	127	127	127	127	127
64	64	64	127	127	127	127	127	127	127
64	64	64	127	127	127	127	127	127	127
64	64	64	127	127	127	127	127	127	127
64	64	64	127	127	127	127	127	127	127
64	64	64	127	127	127	127	127	127	127

Fragment corespunzător din imaginea filtrată

64	64	64	64	64	64	64	64	64	64
64	64	64	64	64	64	64	64	64	64
64	64	64	71	78	85	85	85	85	85
64	64	78	92	106	106	106	106	106	106
64	64	85	106	127	127	127	127	127	127
64	64	85	106	127	127	127	127	127	127
64	64	85	106	127	127	127	127	127	127
64	64	85	106	127	127	127	127	127	127
64	64	85	106	127	127	127	127	127	127
64	64	85	106	127	127	127	127	127	127

Filtrul binomial - *Binomial Filter*

- **Definire:** - masca $(n + 1) \times (n + 1)$ se obține din coeficienții binomului de ordin n , $(x + y)^n$.
- **Construcție:** - $cC^T C$, C - vectorul coeficienților binomului de ordin n , c - coeficientul de scalare.

Exemplu: Filtrul binomial 3×3 , $C = [1 \ 2 \ 1]$

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

Separabilitate: este separabil (din construcție).

Filtrul binomial - Exemplu

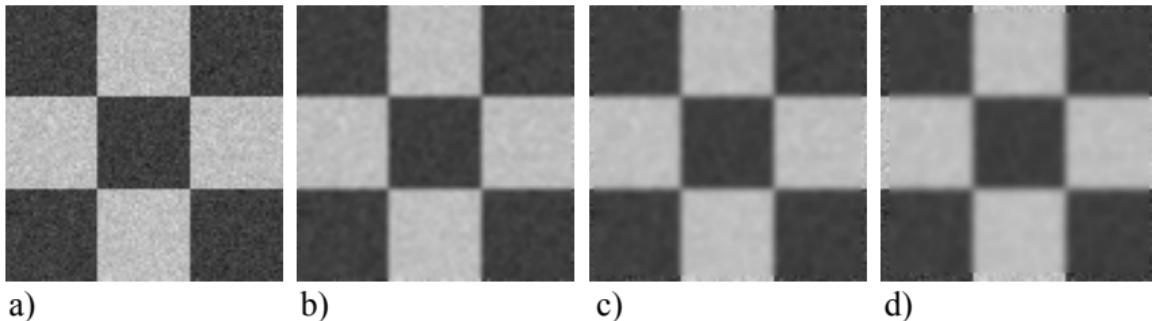
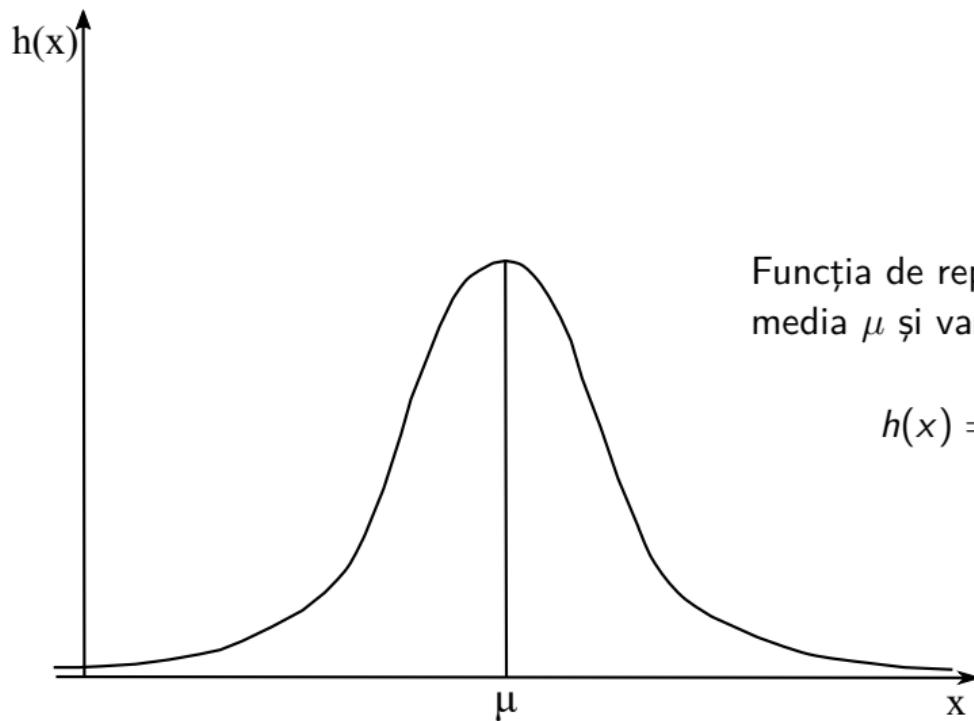


Figure: a) Figură de test sintetică alterată cu zgomot gaussian cu $\sigma = 15$; Imagine filtrată cu un filtru binomial b) 3×3 ; c) 5×5 ; d) 7×7 .

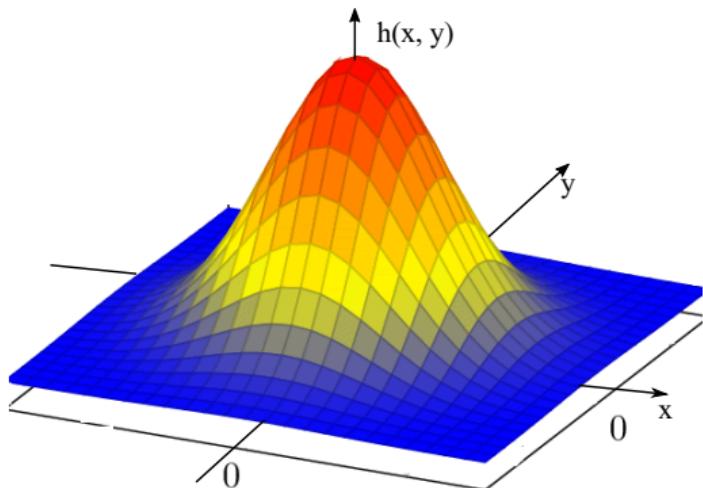
Filtrul Gaussian - *Gaussian Filter*



Funcția de repartiție gaussiana cu media μ și varianța σ

$$h(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Filtrul Gaussian - *Gaussian Filter*



În cazul bi-dimensional, funcția gaussiană de repartiție pentru două variabile aleatoare necorelate x și y având mediile μ_x , μ_y și varianțele σ_x , respectiv σ_y este dată prin:

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

Filtrul Gaussian - *Gaussian Filter*

Elementele măştii se construiesc, coniserând masca centrată în $(0, 0)$ și $\mu_x = \mu_y = 0$:

1. pentru $\sigma_x \neq \sigma_y$:

$$h(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2}\right)}, x, y = -k, k$$

2. pentru $\sigma_x = \sigma_y = \sigma$:

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

Coeficientul măştii c - trebuie ales astfel încât

$$\sum_{x=-k}^k \sum_{y=-k}^k c \cdot h(x, y) = 1$$

$\frac{1}{6.16}$

0.018	0.082	0.135	0.082	0.018
0.082	0.367	0.606	0.367	0.082
0.135	0.606	1	0.606	0.135
0.082	0.367	0.606	0.367	0.082
0.018	0.082	0.135	0.082	0.018

Filtrul Gaussian - *Gaussian Filter*

$\frac{1}{6.16}$

0.018	0.082	0.135	0.082	0.018
0.082	0.367	0.606	0.367	0.082
0.135	0.606	1	0.606	0.135
0.082	0.367	0.606	0.367	0.082
0.018	0.082	0.135	0.082	0.018

Elementele măştii se calculează:

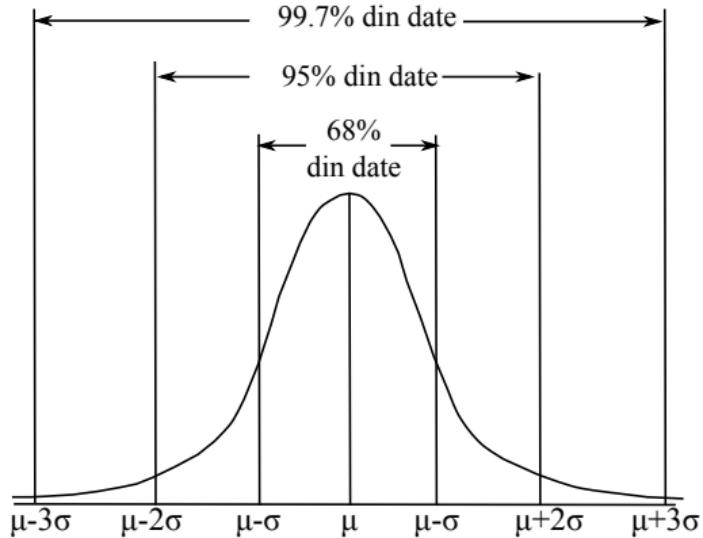
$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}, x, y = -k, k$$

Coeficientul măştii c - trebuie ales astfel încât

$$\sum_{x=-k}^k \sum_{y=-k}^k c \cdot h(x, y) = 1$$

Separabilitate: este separabil

Repartiția gaussiană (normală)



$$h(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Filtrul Gaussian - Exemplu

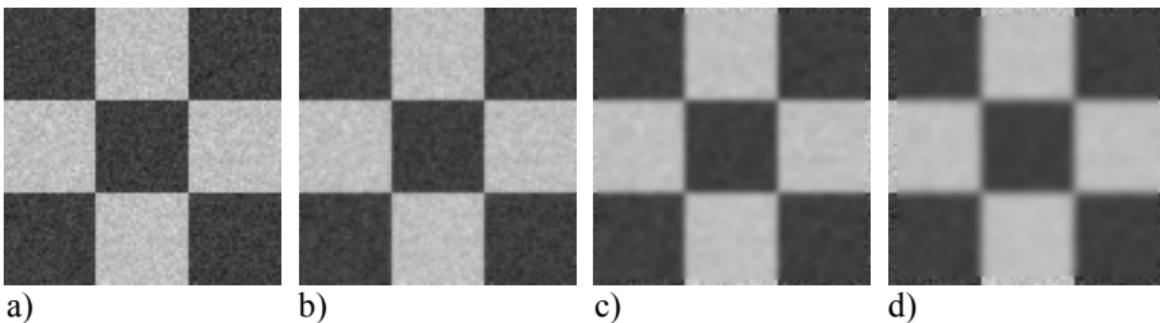


Figure: a) Figură de test sintetică alterată cu zgomot gaussian cu $\sigma = 15$; Imagine filtrată cu un filtru gaussian cu b) $\sigma = 0.5$ rezultă dim=3 × 3; c) $\sigma = 1.0$ rezultă dim=5 × 5; d) $\sigma = 1.5$ rezultă dim=7 × 7.

Filtrul Gaussian - OpenCV

```
public static void GaussianBlur(Mat src, Mat dst, Size ksize, double sigmaX, double sigmaY, int borderType)
```

- param 1: src - imaginea sursă
- param 2: dst - imaginea rezultat, cu aceeași dimensiune și număr de canale ca imaginea sursă
- param 3: ksize - dimensiunea filtrului gaussian (Ex: Size(3,3)). Dacă size = 0, atunci se calculează pe baza varianțelor.
- param 4: sigmaX, varianța în direcția x.
- param 5: sigmaY, varianța în direcția y; dacă sigmaY = 0, atunci se va considera egală cu sigmaX, dacă ambele sunt 0, se vor deduce din dimensiunile filtrului.
- param 6: borderType - flag pentru tratarea marginilor. Valoarea implicită este BORDER_DEFAULT (vezi documentație).

Filtre bazate pe statistici de ordine

- consideră pixelii din masca de filtrare sortați în ordine crescătoare a valorilor de gri;
- înlocuiesc pixelul central cu una dintre valorile pixelilor sortați - ex. medianul.

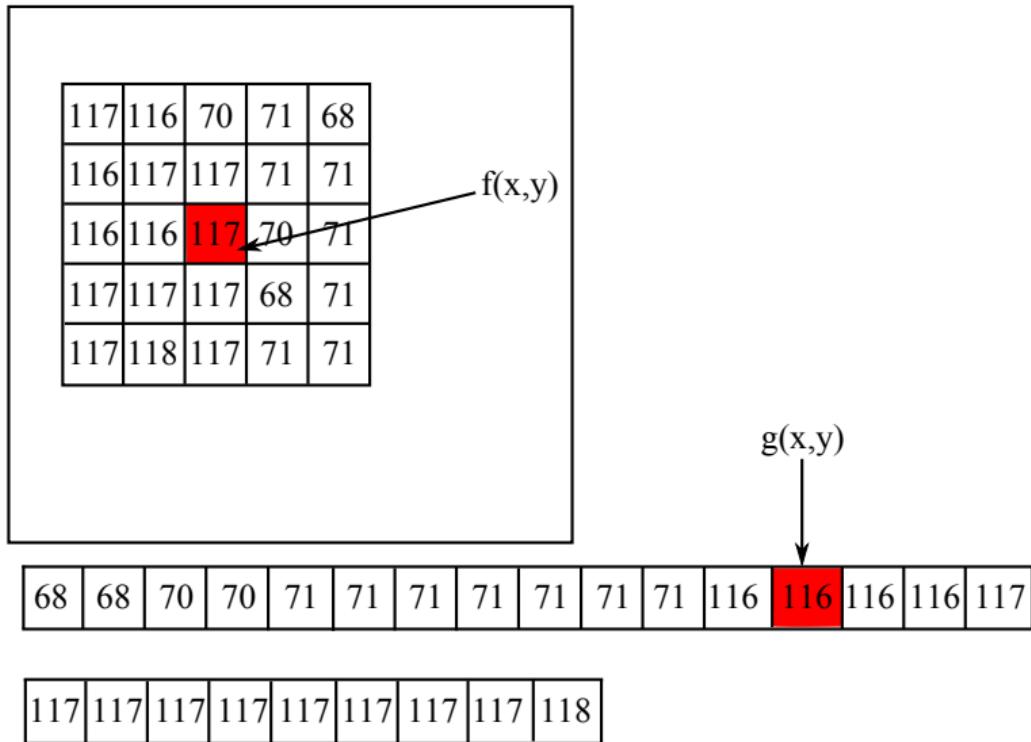
Filtrul median - *Median Filter*

Se consideră vecinătatea S_{xy} dată de o mască de filtrare.

Imaginea filtrată $g(x, y)$:

$$g(x, y) = \underset{S_{xy}}{\operatorname{median}}(f(x, y))$$

Filtrul median - Exemplu



Filtrul median - Exemplu

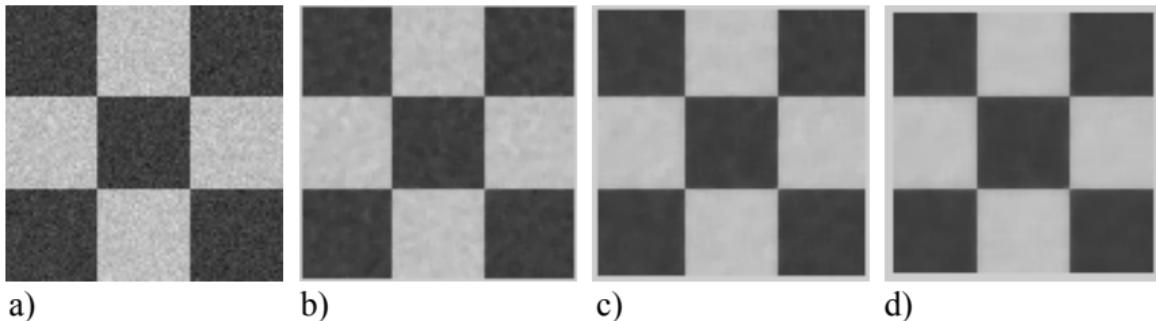
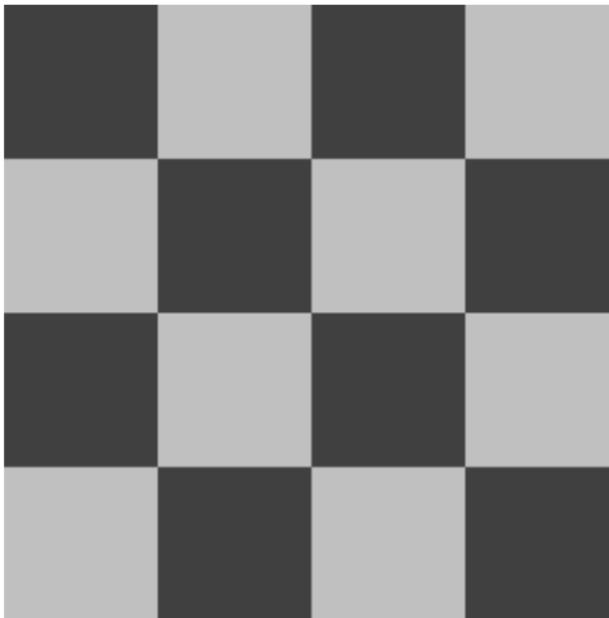


Figure: a) Figură de test sintetică alterată cu zgomot gaussian cu $\sigma = 15$; Imagine filtrată cu un filtru median b) 3×3 ; c) 5×5 ; d) 7×7 .

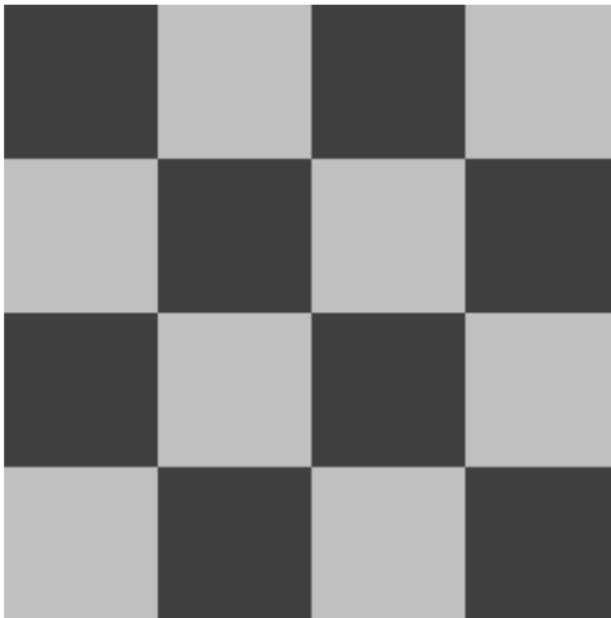
Filtrul median

Întrebare: Ce efect are un filtrul median 3×3 asupra imaginii sintetice de mai jos?



Filtrul median

Întrebare: Ce efect are un filtrul median 3×3 asupra imaginii sintetice de mai jos?



Răspuns: Imaginea nu se modifică!

Filtre trece-jos - Comparație

Ce avantaje - dezavantaje au diferitele tipuri de filtre?

Ce proprietăți au filtrele discutate?

Filtre trece-jos - Comparație

• Filtrele de mediere

- produc valori de gri noi.
- produc nivelare accentuată.
- sunt ușor de implementat

• Filtrul median

- păstrează contururile
- are complexitate mai mare decâtfiltrele de mediere

Unsharp mask utilizând filtre trece-jos

- **Operatorul:**

$$f_{sharp}(x, y) = f(x, y) - f_{low-pass}(x, y)$$

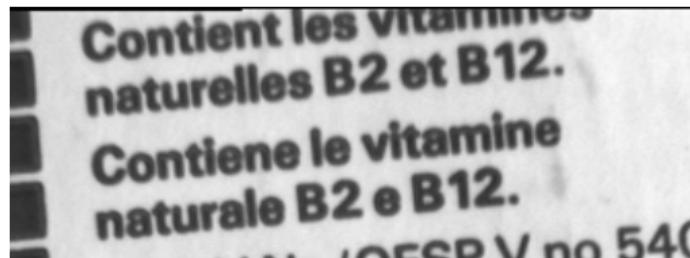
- **Unsharp Mask:**

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

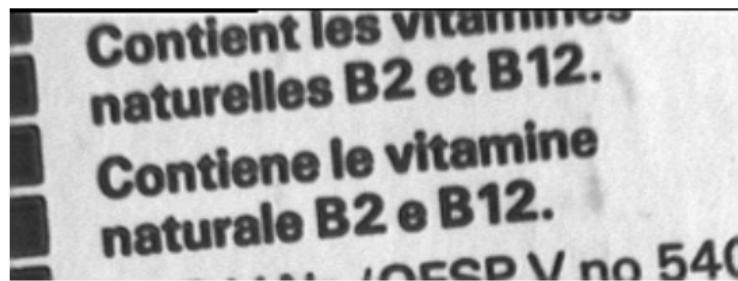
- **High-boost Filtering:**

$$g(x, y) = A \cdot f(x, y) + f_{sharp}(x, y), A \geq 1$$

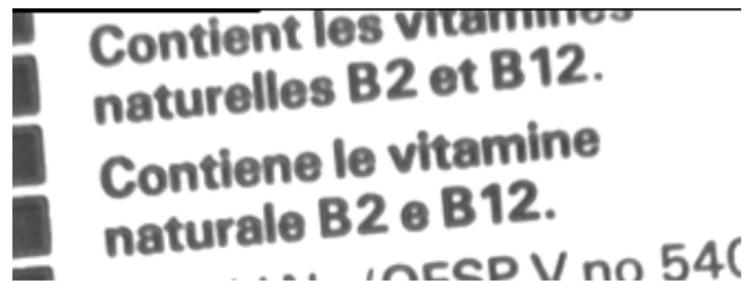
Unsharp mask - Exemple



a)



b)



c)

Figure: a) Imagine de test originală; b) *Unsharp mask* utilizând un filtru gaussian 5×5 ; c) *High-boost filtering* utilizând un filtru gaussian 3×3 cu $A = 1.5$.

Filtrarea trece-jos pentru imagini color

Netezirea imaginii prin filtrare trece-jos În cazul filtrelor liniare se realizează pe fiecare componentă înaparte:

$$c'(x, y) = \begin{pmatrix} h * R(x, y) \\ h * G(x, y) \\ h * B(x, y) \end{pmatrix}$$

unde $c'(x, y)$ - culoarea imaginii filtrate, $*$ - operația de conoluție, h - filtrul de conoluție.

Observație - prin filtrare se introduc valori noi pe cele 3 componente.

Filtrarea trece-jos color - Exemple



Original noisy image



Mean filter 3 x 3



Gauss Filter 3 x 3



Median Filter 3 x 3

Filtrul Median pentru imagini color

Filtrul Median pe cele trei componente de culoare

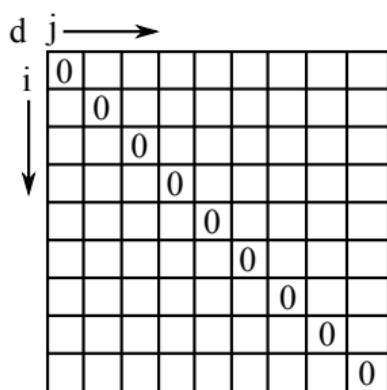
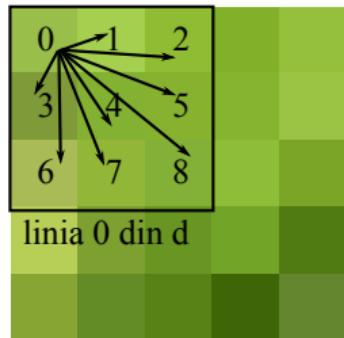
- Realizează filtrarea cu filtrul median clasic pe fiecare dintre cele trei componente r, g, b.
- Nu creează valori noi pentru r, g și b, dar creează combinații noi de triplete ⇒ culori noi, inexistente în imaginea originală.
- Pot apărea modificări cromatice în special în vecinătatea contururilor.

Filtrul Median pentru imagini color

Filtrul Median Vectorial

- Consideră pixelii dintr-o fereastră $(2k + 1) \times (2k + 1)$ centrată în (x, y) .
- Selectează ca valoare rezultat, culoarea acelui pixel pentru care suma distanțelor între această culoare și celelalte culori prezente în imagine este minimă.

Filtrul Median pentru imagini color



Filtrul Median Vectorial - Exemplu pentru o mască 3×3

- Consideră pixelul (x, y) din imaginea f pe care se centreză masca 3×3 .
- Consideră vectorul

$$v = \{f(x-1, y-1), f(x, y-1), \dots, f(x, y+1), f(x+1, y+1)\}$$

- Construiește matricea simetrică $d_{9 \times 9}$

$$d(i, j) = \|v(i) - v(j)\| = \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2}$$

unde $v(i) = (r_i, g_i, b_i)$, $v(j) = (r_j, g_j, b_j)$.

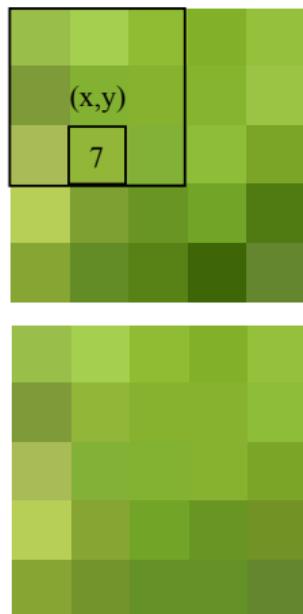
Filtrul Median pentru imagini color

Filtrul Median Vectorial - Exemplu pentru masca 3×3

- $D(i) = \sum_{j=0}^8 d(i, j)$ - suma distanțelor de la pixelul $v(i)$ la ceilalți pixeli din mască.
- $v(k) = \text{acel pixel pentru care se obține } \min\{D(i)|i = \overline{0,8}\}$
- Culoarea pixelului (x, y) în imaginea rezultat va fi culoarea lui $v(k)$.

Filtrul Median pentru imagini color

Filtrul Median Vectorial - Exemplu pentru masca 3×3



d	0	1	2	3	4	5	6	7	8	Sum on lines
0	0	20,857	24,779	47,916	35,057	34,366	20,833	20,542	31,843	236,193
1	20,857	0	39,306	68,607	53,198	51,981	22,113	37,802	50,646	344,51
2	24,779	39,306	0	38,34	17,059	14,457	43,841	7,348	17,493	202,623
3	47,916	68,607	38,34	0	24,104	26,248	61,757	34,669	23,367	325,008
4	35,057	53,198	17,059	24,104	0	3,742	53,972	17,234	5	209,366
5	34,367	51,981	14,457	26,249	3,742	0	53,169	15,264	7,681	206,91
6	20,832	22,113	43,841	61,757	53,972	53,169	0	38,807	50,675	345,166
7	20,543	37,802	7,348	34,67	17,234	15,264	38,807	0	16,186	187,854
8	31,843	50,646	17,493	23,367	5	7,681	50,675	16,186	0	202,891

Minimum

Filtrul Median Vectorial - Exemplu



Original noisy image



Median Filter on 3 channels
 5×5



Vector median filter
 5×5

Rezumat

- Zgomot - cauze -tipuri
- Filtrarea în domeniul spațial - filtre trece -jos
- Modelare matematică - convoluție / corelație
- Tratarea marginilor
- Filtre separabile
- Filtre de mediere
 - Filtrul medie aritmetică
 - Filtrul binomial
 - Filtrul gaussian
- Filtre bazate pe statistici de ordine - Filtrul median
- *Unsharp Mask*
- Filtrarea imaginilor color -Vector median filter