Filtrarea imaginilor

1 Introducere

Filtrarea unei imagini presupune modificarea fiecărui pixel în funcție de vecinătatea sa prin calcularea unei medii ponderate a acestor vecini. Ponderile folosite sunt date de o funcție numită filtru. Există două tipuri mari de filtrări: filtrarea trece-jos - **low-pass filtering**, pentru reducerea zgomotului din imagini și filtrarea trece-sus - **high-pass filtering**, marginilor (discontinuități ale culorii).
În acest capitol se va discuta noțiunea generală de filtrare. Apoi va fi discutată problematica filtrării trece-jos, precum și a filtrării trece sus. Capitolul este organizat în două secțiuni mari, fiecare dedicată unui anumit tip de filtre, discutate pe parcursul a trei cursuri.

2 Competențe conferite

La sfârșitul acestui capitol, studenții vor fi capabili să înțeleagă:

- Ce este operația de filtrare și de unde vine denumirea.
- Cum se realizează operația de convoluție a unei imagini cu o mască de filtrare.
- Care sunt caracteristicile unui filtru trece-jos și care cele ale unui filtru trece-sus.
- Care sunt principalele filtre trece jos și care sunt proprietățile lor.
- Cum funcționează filtrarea bilaterală.
- Care sunt principalele filtre trece-sus și care sunt proprietățile lor
- Cum se face detectarea de margini
- Cum se realizează filtrarea pentru imagini color.

3 Durata medie de studiu individual

Parcurgerea de către studenți a acestei unități de învățare se face în 3-4 ore pentru partea de filtrare în general și filtrare trece-jos în particular și 5-6 ore pentru partea de filtrare trece-sus cu aplicații.

4 Filtrarea în domeniul spațial - modelare matematică

Denumirea de *filtrare* vine din reprezentarea în domeniul frecvență prin transformata Fourier și se referă la eliminarea / reducerea anumitor frecvențe și păstrarea altora. Vom reveni asupra acestui aspect în capitolul despre transformata Fourier.

În domeniul spațial filtrarea se realizează printr-o funcție, care pentru fiecare pixel (x, y) din imaginea inițială calculează o nouă valoare pe baza vecinilor acestui pixel. Dacă funcția este liniară, vorbim despre filtrare liniară.

Filtrarea liniară a unei imagini f cu un filtru (mască, kernel, fereastră) h se realizează printr-o operație care, pentru fiecare pixel (x, y) din f calculează o valoare dată de suma ponderată a vecinilor lui $f(x, y)$, delimitați de masca h, unde ponderile sumei sunt valorile din această mască.

În acest context putem distinge între două operații:

1. Corelația dată prin formula:

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

2. Convoluția dată prin formula:

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

O mască de convoluție/corelație este de obicei de dimensiuni $m \times n$ impare, pentru a avea un pixel central, și se presupune că pixelul central are coordonatele $(0, 0)$. Adesea $m = n = 2k + 1$.

În figura 1 sunt reprezentate o operație de corelație și o operație de convoluție. Practic, pentru o mască h convoluția se obține prin rotirea cu 180^0 a măștii h și aplicarea apoi a corelației, prin centrarea măștii rotite asupra fiecărui pixel în parte (x,y) și calcularea valorii rezultat $g(x, y)$ ca suma ponderată a pixelilor aflați sub mască și având ponderile date de elementele măștii.

În cazul filtrelor trece-jos măștile sunt în general simetrice atât pe verticală cât și pe orizontală. De aceea prin rotirea măștii nu se produc schimbări, convoluția fiind echivalentă cu corelația!

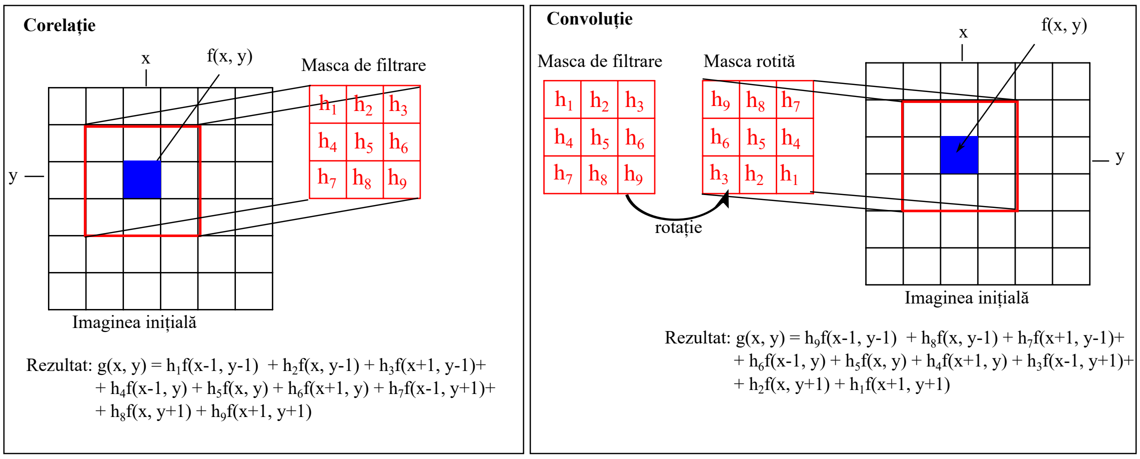


Figura 1: Exemplificarea operațiilor de convoluție și de corelație. Rezultatul $g(x, y)$ reprezintă valoarea care va fi plasată pe poziția (x, y) în imaginea rezultat. Masca este colorată cu roșu, iar pixelul din imaginea sursă f supra căruia se centrează masca este $f(x, y)$

Important: În anumite documentații se vorbește despre convoluție, dar se realizează corelația cu măștile deja rotite. Acest lucru este relevant în cazul filtrelor care nu sunt simetrice după toate direcțiile, cum sunt de exemplu filtrele Sobel.

O reprezentare grafică a operațiilor de corelație și de convoluției este prezentată în fig. 1.

Corelația versus convoluția. Totuși de ce diferențiem între cele două noțiuni și nu utilizăm direct corelația cu măști corespunzătoare, fără a mai lua în discuție convoluția și a ne complica cu rotirea suplimentară? Din cauza proprietăților convoluției. Spre deosebire de corelație, convoluția este atât comutativă cât și asociativă [2]. Acest lucru are implicații semnificative:

1. Considerând două filtre h_1 și h_2 cu care dorim să filtrăm succesiv imaginea f, această operație poate fi exprimată prin

$$h_1 * (h_2 * f) = (h_1 * h_2) * f$$

   și deci, în loc să realizăm două filtrări succesive ale întregii imagini, construim un filtru nou $h = h_1 * h_2$, cu care filtrăm imaginea f. Acest lucru duce la reducerea complexității, în mod semnificativ, pe măsură ce dimensiunea măștilor este mai mare.

2. Aplicarea filtrării poate fi de asemenea eficientizată prin construirea de filtre separabile, așa cum este descris în continuare.

4.1 Separabilitate

O funcție bidimensională $G(x, y)$ se numește separabilă, dacă poate fi scrisă ca produs de două funcții de o variabilă

$$G(x, y) = G_1(x)G_2(x)$$

Pentru un filtru $h(i,j)$, care reprezintă o matrice de coeficienți de dimensiuni $m \times n$ separabilitatea se exprimă prin faptul că această matrice h poate fi scrisă ca produs de un vector coloană cu un vector linie:

$$h = h_1 h_2^T$$

unde h_1 are dimensiunile $m \times 1$ și h_2 are dimensiunile $n \times 1$. Pentru măști pătratice $m = n$. Un exemplu este masca separabilă este

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

În plus, produsul dintre un vector linie și un vector coloană este echivalent cu convoluția dintre cei doi vectori [2]. Astfel

$$h = h_1 h_2^T = h_1 * h_2^T$$

.

Acest lucru, combinat cu proprietățile de comutativitate și asociativitate ale operației de convoluție, are următoarea consecință importantă:

$$h * f = (h_1 * h_2^T) * f = h_1 * (h_2^T * f) = h_2^T * (h_1 * f)$$

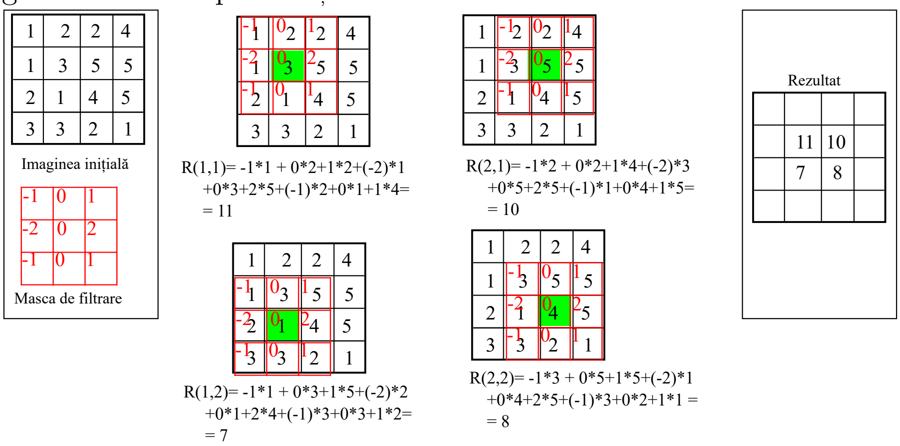
Dacă dimensiunile lui h sunt $m \times n$ și dimensiunile imaginii sunt $M \times N$, atunci operația de convoluție a imaginii f cu filtrul h are complexitatea $m \times n \times M \times N$. Dacă în schimb realizăm două convoluții succesive:

- $g_1 = h_2^T * f$ - obținem o imagine de dimensiune $M \times N$ cu o complexitate o operației de convoluție $n \times M \times N$
- $g_2 = h_1 * g_1$ -obținem imaginea rezultat de dimensiune $M \times N$ cu o complexitate o operației de convoluție $m \times M \times N$

Complexitatea rezultată pentru ambele operații este $MN(m+n)$, care pentru m și n relativ mari este semnificativ mai mică decât convoluția cu masca h.

Astfel, pentru măști separabile de dimensiuni mari, filtrarea succesivă cu măști de tip vector linie respectiv coloană este preferabilă filtrării cu masca originală.

Exemplu: de convoluție cu un filtru Sobel. Masca de filtrare este deja rotită în exemplu. Sunt calculate valorile convoluției pentru 4 pixeli ai imaginii, pentru coordonatele (x, y): $(1,1)$, $(2,1)$, $(1,2)$ și $(2,2)$. Este vorba de coordonatele ecran, în care prima componentă x reprezintă *coloana*. iar a doua componentă y reprezintă linia. Pixelul procesat este cel colorat cu verde. Coeficienții măștii (cu roșu) sunt suprapuși peste pixelii din imaginea inițială, pentru a ilustra mai bine corespondența. În imaginea rezultat au fost plasate valorile obținute prin convoluție. Se observă că pixelii de pe margine nu au fost procesați.



Algoritmul general de filtrare a unei imagini de dimensiuni $M \times N$ cu un filtru h de dimensiuni $(2k+1) \times (2k+1)$ (deja rotit) cu ignorarea pixelilor de pe margine este prezentat mai jos. Observăm faptul, că pe măsură ce crește dimensiunea filtrului, crește și numărul de pixeli din imagine care se pierd (sunt ignorați).

Algoritm: Filtering

Intrare: Imaginea inițială Img de dimensiuni $M \times N$, masca h de dimensiuni $(2k+1) \times (2k+1)$

Iesire: Imaginea filtrară Rez

```
pentru y = k, M − k executa
    pentru x = k, N − k executa
        sum ← 0
        pentru i = −k, k executa
            pentru j = −k, k executa
                sum ← sum + h[y + i][x + j] ∗ Img[y + i][x + j]
            sfarsit_for
        sfarsit_for
        Rez(y, x) ← sum
    sfarsit_for
sfarsit_for
```

a) b) c) d)

Figura 2: a) Imaginea originală. b) Imaginea bordată cu 0. c) Imaginea prelungită prin periodicitate. d) Imaginea prelungită prin simetrie.

4.2 Tratarea marginilor

Probleme apar atunci când se procesează pixeli de pe marginea imaginii, deci când $x < k$ sau $y < k$ sau $x > M - 1 - k$ sau $y > N - 1 - k$. Problema tratării marginilor imaginii poate fi rezolvate în mai multe moduri:

(1) Ignorarea pixelilor de pe margine prin lăsarea lor neschimbați, atunci când masca are dimensiuni relativ mici și când zona de interes se află într-o regiune relativ centrală a imaginii.

(2) Bordarea imaginii cu 0. În general această metodă nu se folosește deoarece introduce erori semnificative pe margine.

(3) Bordarea prin periodicitate în toate cele 4 direcții ale imaginii. Această metodă poate introduce erori semnificative, atunci când marginea dreaptă diferă semnificativ de cea stângă, respectiv marginea de sus de cea de jos a imaginii.

(4) Bordarea imaginii prin simetrie. Este o metodă ceva mai elaborată decât prelungirea prin periodicitate, dar este preferată, atunci când tratarea marginilor este esențială pentru algoritm, de exemplu atunci când se lucrează cu diferite scale ale imaginii sau cu filtre de dimensiuni mari.

(5) Bordarea imaginii prin replicare, presupune adăugarea de linii, coloane pe margine, care reprezintă replici ale pixelilor de pe marginile imaginii. Această metodă poate fi o alternativă la (4), dacă dimensiunea măștii este relativ mică.

Exemple pentru tratarea marginilor în cazurile (2) - (4) sunt prezentate în fig. 2.

5 Filtre trece-jos - *Low-pass Filters*

Filtrele trece-jos sunt un caz particular de filtre. Acestea sunt folosite pentru reducerea zgomotului în imagini digitale și reprezintă adesea un pas de preprocesare în algoritmi mai complecși.

5.1 Zgomot

Perturbațiile în conținutul datelor imagistice, care se produc în diferite stadii ale procesului de achiziție sau de transmitere a imaginilor, se numesc zgomot. Aceste perturbații ale semnalului pot produce inconveniențe vizuale dar pot, de asemenea, să influențeze procesarea și analiza imaginii. Zgomotul este vizibil în special în regiunile considerate uniforme.

Procesul de reducere a zgomotului într-un semnal digital presupune reducerea erorilor introduse la achiziția și/sau transmisia datelor imagistice cu alterarea minimă a informației semnalului original. De obicei, reducerea zgomotului reprezintă un pas de preprocesare, înaintea efectuării unor sarcini mai complexe cum ar fi segmentarea.

Sursele principale de zgomot se întâlnesc în sistemele de achiziție și transmitere a imaginilor. Zgomotul poate fi produs de către senzorii dispozitivului imagistic și de interferența cu alte semnale.

În general orice dispozitiv electronic produce o anumită cantitate de zgomot. Această cantitate poate să varieze în funcție de tipul de dispozitiv, timpul de expunere pentru achiziția imaginii și de alți factori.

Zgomotul produs de camerele digitale poate fi:

- **zgomot aleator**, caracterizat prin variații aleatorii ale intensității/culorii valorii reale a pixelului imaginii. Acest tip de zgomot poate fi modelat printr-o variabilă aleatoare;
- **zgomot cu pattern fix**, cauzat mai ales de către imperfecțiuni ale senzorilor, de exemplu pixeli fierbinți sau morți - *hot or dead*. Distribuția lui spațială este aproape aceeași pentru imaginile obținute în același condiții. Acest tip de zgomot este amplificat prin timpii de expunere lungi și este influențat de temperatura echipamentului electronic.;
- **zgomot bandă**, care este corelat cu structura hard a dispozitivului imagistic. Zgomotul bandă se produce în timpul citirii datelor din senzorii digitali;
- **zgomot periodic**; cauzat mai ales de interferențele electrice și electromagnetice.

Modelarea zgomotului

Zgomotul este caracterizat în domeniul spațial prin parametri ca medie, varianță și probabilitate și în domeniul frecvență prin spectrul frecvențelor obținut prin transformata Fourier. În domeniul spațial se pot defini două tipuri de zgomot:

- zgomot independent de coordonatele spațiale, cum ar fi zgomotul aleator;
- zgomot dependent de coordonatele spațiale, cum ar fi zgomotul periodic sau zgomotul cu pattern fix.

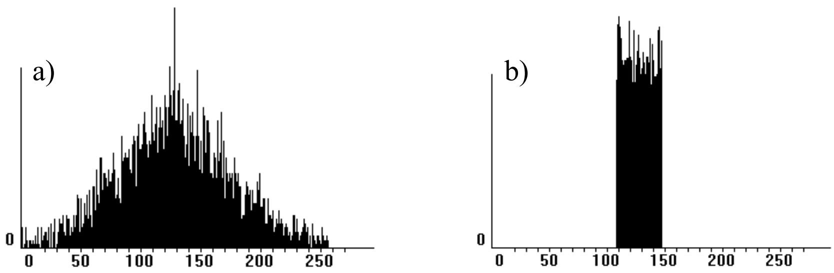


Figura 3: a) Zgomot Gaussian; b) Zgomot uniform

Zgomotul aleator poate fi modelat printr-o variabilă aleatoare cu o probabilitate dată. Câteva exemple sunt zgomotul Gaussian, zgomotul Rayleigh, zgomotul exponențial, zgomotul uniform. În figura 3 sunt prezentate distribuțiile zgomotului Gaussian și a zgomotului uniform.

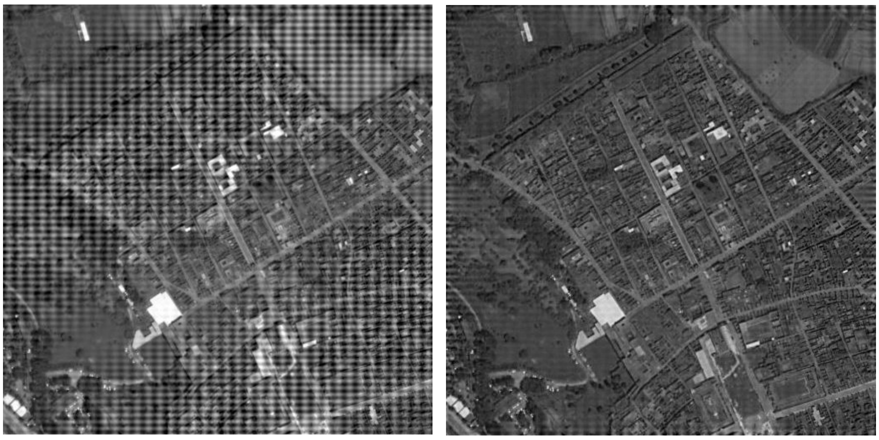
Zgomotul periodic dintr-o imagine poate fi reprezentat printr-o funcție sinus, printr-o combinație liniară de funcții sinus sau prin structuri repetitive, ca de exemplu pattern-urile orizontale, verticale sau diagonale. Modele pentru zgomotul periodic pot fi extrase din spectrul Fourier.

Problema reducerii zgomotului poate fi rezolvată prin determinarea celei mai bune metode de filtrare a zgomotului, păstrând în același timp informația esențială nemodificată. Dacă distribuția zgomotului este cunoscută sau poate fi estimată pentru o arie uniformă a imaginii, problema se reduce la selectarea filtrului adecvat. Din nefericire, imaginile conțin diferite combinații de de tipuri de zgomot. În general nu există informații privind caracteristicile zgomotului.

O măsură a conținutului relativ de zgomot într-o imagine este dată de raportul semnal / zgomot - **SNR** - ***signal to noise ratio***, care reprezintă raportul dintre cantitatea de semnal corect și cantitatea de zgomot. Definiția SNR este dată de raportul dintre valoarea medie a semnalului și deviația standard a fundalului. În cazul scenelor cu contrast ridicat sau cu fundal inexistent aceasta nu este o definiție convenabilă. De aceea, o definiție mai bună a SNR este dată de raportul dintre valoarea medie a semnalului, μ și deviația lui standard, σ:

$$SNR = \frac{\mu}{\sigma}$$

Exemplu Imagine cu zgomot periodic și rezultatul filtrării în spațiul frecvență (Fourier).



Eliminarea zgomotului în domeniul spațial se efectuează prin intermediul filtrării trece-jos. Denumirea de *filtru trece-jos* - *low-pass filter* - provine din filtrarea în domeniul frecvență și se referă la proprietatea filtrelor de a lăsa să "treacă" nemodificate, sau cu modificări minore, componentele de frecvență joasă și a bloca / reduce componentele cu frecvență înaltă. Datorită faptului că variațiile bruște în imagine, care reprezintă contururi și/sau zgomot, se regăsesc în componentele de frecvență înaltă, reducerea acestor componente are ca efect atenuarea tranzițiilor semnificative ale nivelurilor de gri. Cu alte cuvine se obține un efect de netezire - *smoothing* - al contururilor dar și de reducere a zgomotului. În general, cu cât reducerea zgomotului este mai pronunțată, cu atât și efectul de netezire este mai mare. Aceste efecte depind însă de caracteristicile filtrului.

Vom considera în continuare măști pătratice de dimensiuni $(2k+1) \times (2k+1)$.

Pentru filtrele trece-jos o condiție necesară pentru ca media nivelurilor de gri în imagine, respectiv luminozitatea imaginii, să nu se modifice este:

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

unde $h(i,j)$ reprezintă coeficienții măștii de filtrare. De fapt coeficienții din mască trebuie normați cu suma lor.

5.2 Filtre de mediere

Filtrele de mediere, precum reiese și din denumire, realizează o medie ponderată a valorilor de gri dintr-o anumită regiune. Ponderile sunt date de coeficienții filtrului și sunt construite pe baza anumitor modele matematice, de exemplu: media aritmetică, repartiția gaussiană, etc. În continuare vor fi prezentate cele mai frecvent utilizate astfel de filtre împreună cu proprietățile lor.

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \qquad \frac{1}{10}\begin{bmatrix}1 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 1\end{bmatrix}$$

a) b)

Figura 4: a) Filtrul medie aritmetică pentru $k = 3$; b) variantă a filtrului de medie aritmetică în care pixelul central are o pondere mai mare.

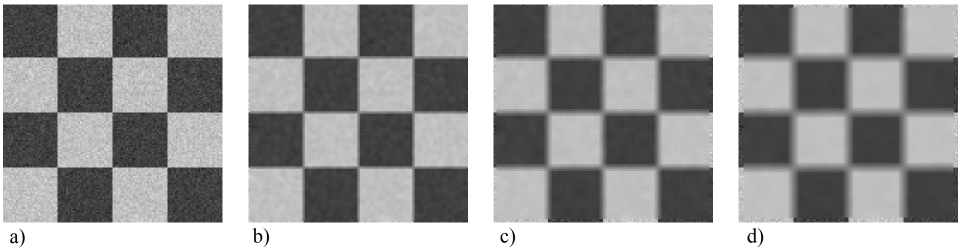


Figura 5: a) Imagine de test sintetică alterată cu zgomot gaussian cu $\sigma = 15$; b) Imaginea filtrată cu un filtru medie aritmetică 3×3; c) Imaginea filtrată cu un filtru medie aritmetică 5×5; d) Imaginea filtrată cu un filtru medie aritmetică 7×7.

5.2.1 Filtrul medie aritmetică - *Average Filter / Box Filter*

Filtrul medie aritmetică înlocuiește valoarea fiecărui pixel cu valoarea medie a pixelilor din regiunea definită de masca de filtrare. Elementele măștii de convoluție de dimensiune $(2k+1) \times (2k+1)$ sunt $h(i,j) = 1$. Coeficientul măștii este $c = 1/(2k+1)^2$, astfel încât

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

Un exemplu de filtru medie aritmetică este dat în figura 4 a).

O variantă a filtrului medie aritmetică este aceea în care pixelului central i se acordă o pondere mai mare, el contribuind într-o măsură mai mare la valoarea rezultată în imaginea filtrată. De exemplu filtrul din fig 4 b).

În figura 5 sunt prezentate exemple de reducere a zgomotului într-o imagine de test utilizând filtre medie aritmetică de diferite dimensiuni.

Așa cum s-a văzut mai sus, la discuția legată de separabilitate, un filtru medie aritmetică este separabil.

Se observă faptul că, pe măsură ce crește dimensiunea filtrului, reducerea zgomotului este mai pronunțată, dar și efectul de netezire, apărut din cauza atenuării contururilor este mai pronunțat.

Pentru dimensiuni mai mari ale filtrului se poate utiliza calculul rapid al mediei unei vecinătăți folosind imaginea integrală, așa cum a fost descris în capitolul de *thresholding.*

5.2.2 Filtrul Gaussian

Unul dintre cele mai frecvent utilizate filtre de netezire este *filtrul Gaussian*. El se numește astfel, deoarece coeficienții săi se obțin pe baza repartiției Gaussiene. Spunem că o variabilă aleatoare X cu media μ și abaterea medie pătratică σ^2 are o distribuție Gaussiană (este normal distribuită), dacă are funcția de densitate de probabilitate dată prin:

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

Această funcție este reprezentată în figura 6 a). Putem observa faptul că 95% dintre valori sunt cuprinse în intervalul $[\mu - 2\sigma, \mu + 2\sigma]$ și peste 99% dintre valori sunt cuprinse în intervalul $[\mu - 3\sigma, \mu + 3\sigma]$.

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:

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

Această funcție este reprezentată grafic în figura 6 b).

Elementele măștii se construiesc, considerând masca centrată în $(0, 0)$ și $\mu_x = \mu_y = 0$:
1. pentru $\sigma_x \neq \sigma_y$:

$$h(i, j) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\left(\frac{i^2}{2\sigma_x^2} + \frac{j^2}{2\sigma_y^2}\right)}, i, j = \overline{-k, k} \tag{3}$$

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

$$h(i, j) = \frac{1}{2\pi\sigma^2} e^{-\frac{(i^2+j^2)}{2\sigma^2}}, i, j = \overline{-k, k} \tag{4}$$

Coeficientul măștii c - trebuie ales astfel încât

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

Astfel fiecare linie, respectiv coloană, este o aproximare a unei funcții Gaussiene.

Dimensiunea măștii Gaussiene

Dimensiunea măștii se determină în funcție de parametrul σ, astfel încât să cuprindă toate valorile semnificative. Din figura 6 reiese faptul că dimensiunea măștii de filtrare ar

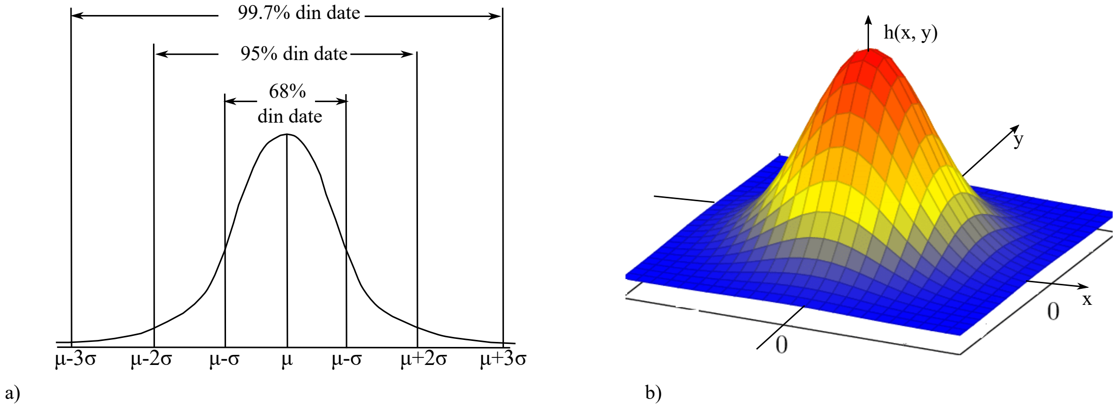


Figura 6: Funcția Gaussiană pentru media μ și abaterea medie pătratică σ^2.

trebui să fie între 4σ și 6σ, în funcție de cât de exactă se dorește aproximarea. În general este suficientă dimensiunea de aproximativ 4σ, considerând cea mai mică valoare impară $\geq 4\sigma$, ceea ce corespunde unei măști de 3×3 pentru $\sigma = 0.5$, unei măști de 5×5 pentru $\sigma = 1$ și de 7×7 pentru $\sigma = 1.5$.

Observație: Coeficienții pot fi calculați și pe baza formulei:

$$G_\sigma(i,j) = ce^{-\frac{i^2+j^2}{2\sigma^2}},$$

valoarea $1/(2\pi\sigma^2)$ intrând în calculul coeficientului de normare. În acest caz elementul nenormat de pe poziția $(0,0)$ din mască este 1.

Exemple: de măști pentru $\sigma = 0.5$ și pentru $\sigma = 1$, unde $c =$ este coeficientul de normare și este egal cu 1/(suma valorilor din mască).

$$c\begin{bmatrix} 0.018 & 0.135 & 0.018 \\ 0.135 & 1.0 & 0.135 \\ 0.018 & 0.135 & 0.018 \end{bmatrix} \qquad c\begin{bmatrix} 0.018 & 0.082 & 0.135 & 0.082 & 0.018 \\ 0.082 & 0.367 & 0.606 & 0.367 & 0.082 \\ 0.135 & 0.606 & 1.0 & 0.606 & 0.135 \\ 0.082 & 0.367 & 0.606 & 0.367 & 0.082 \\ 0.018 & 0.082 & 0.135 & 0.082 & 0.018 \end{bmatrix}$$

În figura 7 sunt prezentate exemple de reducere a zgomotului într-o imagine de test utilizând filtre Gaussiene de diferite dimensiuni.

Filtrul Gaussian este utilizat în diferite aplicații, de exemplu pentru detectarea de contur cu operatorul Marr-Hildreth sau în extragerea punctelor cheie cu algoritmul SURF.

Observații: O problemă în cazul filtrării cu un filtru Gaussian este complexitatea operației

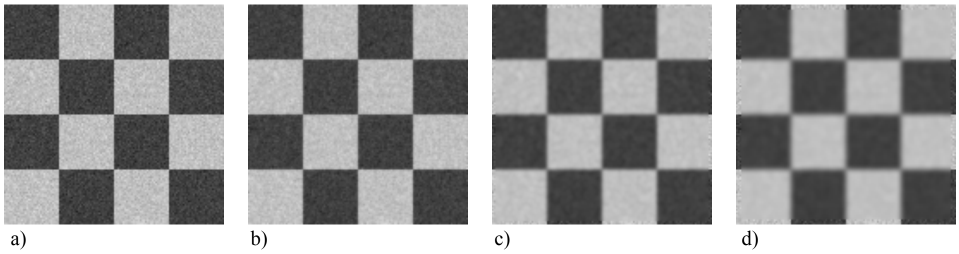


Figura 7: a) Imagine de test sintetică alterată cu zgomot gaussian cu $\sigma = 15$; b) Imaginea filtrată cu un filtru gauss 3×3 ($\sigma = 0.5$); c) Imaginea filtrată cu un filtru gauss 5×5 ($\sigma = 1$); d) Imaginea filtrată cu un filtru gauss 7×7 ($\sigma = 1.5$).

de convoluție atunci când masca este de dimensiuni mai mari. Se poate obține o creștere a vitezei de execuție în diferite moduri:

- În loc să se aplice o mască de filtrare mai mare se poate aplica repetitiv o mască mai mică. Se poate demonstra că aplicarea de două ori succesiv a de unei măști de filtrare Gaussiene cu abaterea medie σ este echivalentă cu aplicarea unei măști Gaussiene cu abaterea medie $\sigma\sqrt{2}$. De fapt convoluția a două filtre Gaussiene cu abaterile pătratice medii σ_1^2 și σ_2^2 are ca rezultat un filtru Gaussian cu abaterea medie pătratică $\sigma^2 = \sigma_1^2 + \sigma_2^2$.
- Filtrarea Gaussiană se poate aproxima prin filtrări succesive cu un filtru de medie aritmetică.
- Filtrul Gaussian este separabil: se poate efectua întâi o filtrare cu un filtru unidimensional Gaussian în direcția liniilor, iar apoi rezultatul se poate filtra cu același filtru, dar orientat la 90^0, în direcția coloanelor.

To Do:

- Implementați unul dintre filtrele de mediere pentru imagini în tonuri de gri și pentru imagini color în *framework*-ul de laborator.
- Implementați unul dintre filtrele de mediere separabil.
- Căutați în documentația online funcțiile de filtrare din OpenCv pentru filtrele medie aritmetică și Gaussian. Studiați care este semnificația parametrilor acestor funcții.

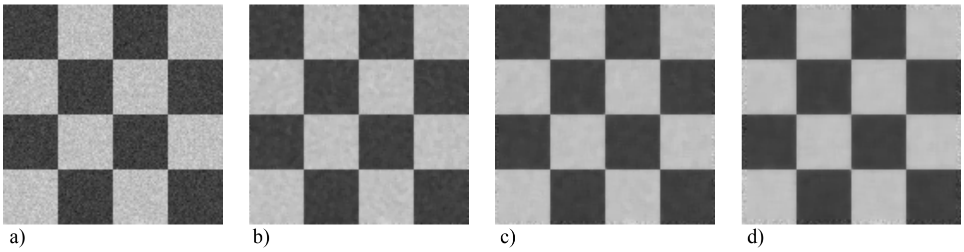
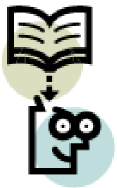


Figura 8: a) Imagine de test sintetică alterată cu zgomot gaussian cu $\sigma = 15$; b) Imaginea filtrată cu un filtru median 3×3; c) Imaginea filtrată cu un filtru median 5×5; d) Imaginea filtrată cu un filtru median 7×7.



Să ne reamintim că

- filtrarea cu un filtru trece-jos de mediere realizează pentru fiecare pixel (x, y) o medie ponderată a vecinătății acestui pixel, unde ponderile sunt coeficienții din mască. Valoare pixelului (x, y) în imaginea rezultat va fi această medie ponderată.
- filtrarea trece jos duce la o netezire a contururilor - un efect de "blurare" a imaginii. Cu cât masca de filtrare este mai mare, cu atât acest efect este mai puternic.
- suma coeficienților unui filtru trece-jos de mediere trebuie să fie 1, pentru a nu modifica luminozitatea medie din imagine.

5.3 Filtre bazate pe statistici de ordine

În cazul filtrelor bazate pe statistici de ordine, masca filtrului nu conține coeficienți, ci doar definește o vecinătate în jurul pixelului central. Pixelii din vecinătatea respectivă se consideră în ordinea valorilor de gri și se selectează un pixel rezultat în funcție de un anume criteriu.

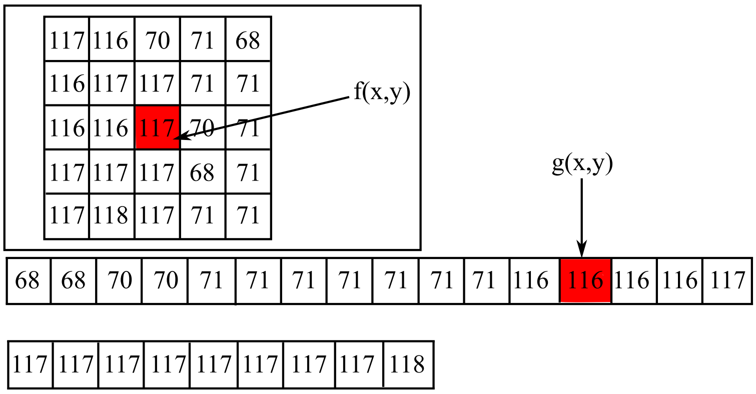
5.3.1 Filtrul median

Cel mai frecvent utilizat filtru bazat pe statistici de ordine este filtrul median, definit în modul următor:

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

unde S_{xy} este vecinătatea definită de masca centrată în pixelul (x, y), f este imaginea originală și $g(x, y)$ este valoarea rezultată pentru pixelul (x, y) după filtrare. Median înseamnă valoarea din mijlocul șirului valorilor luate în ordine crescătoare ale pixelilor din vecinătate.

Exemplu numeric de aplicare a filtrului median este prezentat în figură. Rezultatul $g(x, y)$ este obținut prin aplicarea unui filtru median de dimensiune 5×5 asupra unui pixel din imaginea $f(x, y)$



În figura 8 este prezentat un exemplu de filtrare a zgomotului gaussian cu filtrul median. Se observă faptul că, filtrul median alterează mai puțin contururile decât filtrele de mediere.

5.3.2 Filtrele min / max

Filtrele *min* și *max* se definesc prin

$$g(x, y) = \min_{S_{xy}}(f)$$

și respectiv

$$g(x, y) = \max_{S_{xy}}(f)$$

În cazul filtrului min are loc o creștere a suprafețelor/obiectelor închise la culoare, iar în cazul celui max, o creștere a suprafețelor/obiectelor deschise la culoare.

5.3.3 Filtrul *mid-point*

Filtrul *mid-point* combină ideea de statistică de ordine cu cea de mediere a valorilor. Formula de calcul este dată prin:

$$g(x, y) = \frac{1}{2}\left(\min_{S_{xy}}(f) + \max_{S_{xy}}(f)\right),$$

unde f reprezintă imaginea inițială, S_{xy} vecinătatea pixelului (x, y) definită de către mască și $g(x, y)$ valoarea corespunzătoare a pixelului în imaginea rezultat.

Un exemplu de aplicare al fitrului *mid-point* asupra imaginii de test sintetice reprezentând o tablă de șah este prezentat în figura 9.

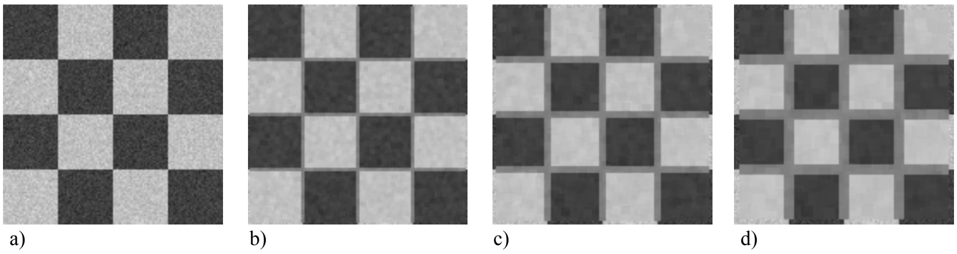
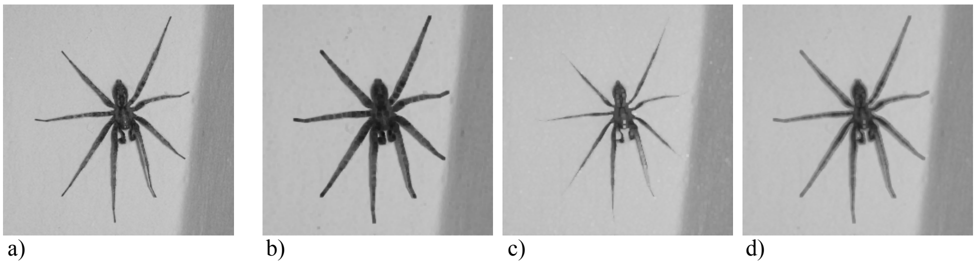


Figura 9: a) Imagine de test sintetică alterată cu zgomot gaussian cu $\sigma = 15$; b) Imaginea filtrată cu un filtru *mid-point* 3×3; c) Imaginea filtrată cu un filtru *mid-point* 5×5; d) Imaginea filtrată cu un filtru *mid-point* 7×7.

Exemplu comparativ de filtrare cu filtrele min (b), max (c) și mid-point (d) a unei imagini în tonuri de gri (a).



5.4 Comparație între filtrele de mediere și filtrul median

- Filtrele medie aritmetică și Gaussian produc valori care nu sunt prezente în imaginea originală. Ele netezesc contururile, ceea ce, în cele mai multe aplicații, reprezintă un efect nedorit. Acest efect nu are loc în cazul filtrelor bazate pe statistici de ordine precum filtrul median.

- Filtrul median produce numai valori de gri care apar în imaginea originală. Acest filtru păstrează contururile nealterate, reducând în același timp în mod semnificativ zgomotul.

- Filtrul medie aritmetică se caracterizează printr-un timp de execuție redus datorită algoritmului său simplu, precum și posibilității reducerii complexității utilizând imaginea integrală, în timp ce filtrul median este de o complexitate mai mare, datorată sortării valorilor pixelilor. Avantajul său însă constă în păstrarea nealterată a contururilor. Există însă în prezent algoritmi de calcul rapid sau paralelizat pentru filtrul median, astfel încât complexitatea nu mai este un impediment.

În figura 10 poate fi observată modificarea contururilor produsă de filtrele medie aritmetică, binomial, gaussian și median.

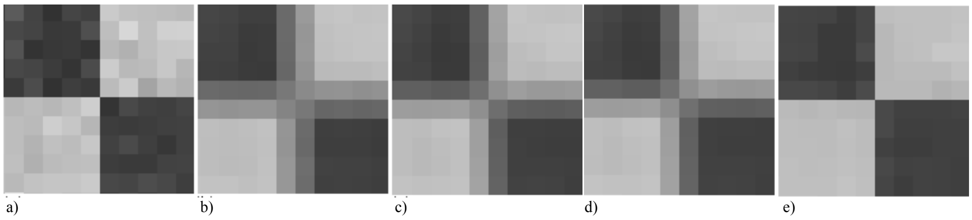


Figura 10: a) Secțiune mărită a imaginii de test alterată cu zgomot Gaussian ($\sigma = 10$). Secțiune mărită a imaginii de test după aplicarea unui filtru 3×3 b) medie aritmetică; c) binomial; d) Gaussian; e) median.

To Do:

- Implementați filtrul median pentru imagini în tonuri de gri și pentru imagini color (pe fiecare canal înparte) în *framework*-ul de laborator.
- Căutați în documentația online funcțiile de filtrare din OpenCv pentru filtrul median.

5.5 Operatorul *Unsharp-mask*

Pentru a crește claritatea imaginii pot fi puse în evidență contururile obiectelor și regiunilor folosind operatorul *unsharp-mask*. Acesta poate fi construit atât pe baza filtrelor trece-jos, așa cum va fi descris în această secțiune, cât și pe baza filtrelor trece-sus. Metoda prezentată în [2] constă în următoarele etape:

(1) Se netezește imaginea cu un filtru trece-jos

(2) Se calculează diferența pixel cu pixel între imaginea originală și cea netezită.

(3) Rezultatul de la punctul (2) se adaugă la imaginea originală.

Observații: Prin scăderea din imaginea originală a imaginii netezite, în zonele uniforme rezultatul acestei scăderi va fi 0 sau aproape 0, iar în regiunile de contur vor apărea valori negative sau, respectiv, pozitive în funcție de situația, dacă acel contur este la o trecere de la o regiune închisă la una deschisă sau invers. Aceste valori, adunate apoi la imaginea originală vor determina modificări în zonele de contur care vor duce la o ușoară creștere a contrastului în acele regiuni. Astfel de crează impresia de creștere a clarității imaginii.

Algoritmul poate fi sintetizat prin formulele următoare:

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

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

5.6 Filtrarea trece-jos a imaginilor color

În cazul imaginilor color în spațiul de culoare RGB, filtrarea liniară se realizează pe fiecare componentă în parte.

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

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

Un caz aparte este filtrul median pe color. Există posibilitatea de a realiza și această filtrare pe fiecare componentă de culoare separat, dar se pierde unul dintre avantajele filtrului median, și anume acela al neintroducerii de valori noi, inexistente în imagine. Datorită faptului că se face sortarea valorilor separat pe fiecare culoare, pe fiecare componentă este posibil să se selecteze elemente aflate pe poziții diferite în imaginea originală, producând astfel valori noi. O alternativă la acestă abordare este ce a filtrului median vectorial, în care se iau în considerare vectorii de culoare.

5.6.1 Filtrul median vectorial

Se consideră o mască de filtrare de dimensiuni $(2k+1) \times (2k+1)$. Pixelul central din această mască se va înlocui în imaginea rezultat cu acel pixel, pentru care suma distanțelor dintre culoarea sa și cea a celorlalți pixeli din vecinătate este minimă. Această idee se bazează pe generalizarea noțiunii de *median* al unei secvențe de date. Considerând valori scalare (ca în cazul valorilor de gri), se poate arăta ușor faptul că, medianul unei secvențe $P = \{p_1, p_2, \ldots, p_n\}$ este valoarea $p_m \in P$, pentru care

$$\sum_{i=1}^{n} |p_m - p_i| \leq \sum_{i=1}^{n} |p_j - p_i|, \forall j \in \{1, 2, \ldots, n\}.$$

Se poate generaliza această definiție pentru secvențe de valori vectoriale (cum sunt de exemplu culorile RGB) $P = \{p_1, p_2, \ldots, p_n\}$, medianul p_m fiind acea valoare care satisface relația:

$$\sum_{i=1}^{n} \|p_m - p_i\|_L \leq \sum_{i=1}^{n} \|p_j - p_i\|_L, \forall j \in \{1, 2, \ldots, n\}, p_i \in \mathbb{R}^K$$

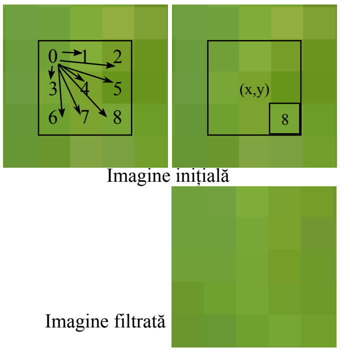
Numim

$$D_L(p, P) = \sum_{p_i \in P} \|p - p_i\|_L$$

"distanța agregată" (*aggregate distance*) a vectorului de referință p relativ la toate elementele lui P. Pot fi folosite ca distanțe între vectori:

$$L_1 : \|p - q\|_1 = \sum_{k=1}^{K} |p_k - q_k|$$

$$L_2 : \|p - q\|_2 = \sqrt{\sum_{k=1}^{K} (p_k - q_k)^2}$$



d	0	1	2	3	4	5	6	7	8	Sum on lines
0	0	23.686	16.432	9.274	15.133	32.939	8.307	14.457	9.695	129.923
1	23.686	0	27.331	14.731	20.976	49.779	26.646	15.166	15.246	193.561
2	16.432	27.331	0	16.673	6.557	23.087	11.358	12.207	15.033	128.678
3	9.274	14.731	16.673	0	11.874	37.376	12.450	8.062	2.00	112.440
4	15.133	20.976	6.557	11.874	0	29.086	11.747	6.164	10.050	111.589
5	32.939	49.779	23.087	37.376	29.086	0	25.961	34.986	35.902	269.116
6	8.307	26.646	11.358	12.450	11.747	25.961	0	14.353	11.445	122.267
7	14.457	15.166	12.207	8.062	6.164	34.986	14.353	0	6.708	112.103
8	9.695	15.246	15.033	2.00	10.050	35.902	11.445	6.708	0	106.079

Minimum

Figura 11: Secțiune dintr-o imagine color, asupra căreia se aplică filtrul median vectorial. Tabelul distanțelor între culorile pixelilor din vecinătatea 3×3 a pixelului central.

$$L_\infty : \|p - q\|_\infty = \max |p_k - q_k|, k \in \overline{1, K}$$

Algoritmul de calcul pentru filtrul median vectorial cu o mască 3×3, considerând distanța Euclidiană $L_2 = \|\|_2$, este următorul. Pentru fiecare pixel (x, y)

- Se consideră vecinătatea sa

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

- Se construiește matricea simetrică $d_{9\times 9}$

$$d(i,j) = \|v(i) - v(j)\|_2 = \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2}$$

  unde $v(i) = (r_i, g_i, b_i)$, $v(j) = (r_j, g_j, b_j)$, $v(i)$ - al i -lea pixel din vectorul v. (fig. 11)

- Pentru fiecare $i \in \{0, 1, \ldots, 8\}$ se calculează suma distanțelor $d(i,j)$ de la $v(i)$ la ceilalți pixeli:

$$D(i) = \sum_{j=0}^{8} d(i,j),$$

  adică suma pe liniile matricii d

- Culoarea pixelului (x, y) în imaginea rezultat, va fi culoarea acelui pixel $v(k)$ pentru care se obține $\min\{D(i) | i = \overline{0, 8}\}$

În exemplul din figura 11 este prezentată o vecinătate a unui pixel color (x, y), împreună cu matricea d a distanțelor între pixelii vecinătății. Matricea are o coloană suplimentară, care conține sumele valorilor de pe liniile matricii d, adică valorile $D(i), i = \overline{0, 8}$. Se observă faptul că suma minimă se obține pentru pixelul 7. Culoarea pixelului central se va înlocui cu culoarea pixelului aflat pe poziția 7 din vecinătatea selectată.

Exemplu comparativ de filtrare cu un filtru Gaussian aplicat pe cele 3 canale de culoare și cu un filtru median vectorial a unei imagini color alterate de zgomot.

Imagine originală

Imagine filtrată cu un filtru Gauss aplicat pe fiecare dintre cele 3 canale

Imagine filtrată cu un filtru median vectorial de dimensiune 5 x 5

Să ne reamintim

- Filtrarea trece-jos a unei imagini color poate fi făcută pe pe fiecare canal în parte sau considerând vectorul (r, g, b)
- Filtrul median aplicat pe fiecare canal de culoare separat poate produce artefacte pe contururi.
- Filtrul median vectorial selectează ca valoare rezultată pentru pixelul curent (x, y), acel pixel din vecinătatea lui (x, y) care se află la cea mai mică distanță față de toți pixelii din vecinătate, unde ca distanță poate fi considerată distanța euclidiană între vectorii de culoare corespunzători.

6 Filtrarea trece-jos adaptivă

În secțiunile anterioare s-a discutat importanța filtrelor trece-jos pentru reducerea zgomotului din imagini. Etapa de reducere a zgomotului poate reprezenta o etapă de preprocesare importantă pentru anumiți algoritmi de segmentare sau recunoaștere. Așa cum reiese și din numele acestor filtre, *trece-jos*, ele conduc la o atenuare sau suprimare a frecvențelor înalte. Aceste frecvențe, așa cum se va vedea la capitolul legat de transformata Fourier, conțin informația modificărilor bruște a culorii într-o vecinătate. Astfel ele sunt caracteristice nu doar zgomotului, ci și contururilor. Cum s-a putut observa din exemple, filtrele trece-jos produc o alterare a contururilor, care este cu atât mai semnificativă, cu cât dimensiunea filtrului este mai mare. Cea mai bună performanță în acest sens o are filtrul median. Cu toate acestea, atunci când în imagine sunt prezente linii relativ subțiri, așa cum este de exemplu în cazul literelor și cifrelor, filtrul median poate duce la distrugerea semnificativă acestora. O preocupare în domeniu a reprezentat-o astfel, elaborarea unor filtre, care să reducă zgomotul, dar să păstreze contururile cât mai puțin alterate. În această secțiune sunt descrise două tipuri filtre trece-jos adaptive care urmăresc să netezească mai puternic regiunile considerate relativ uniforme, păstrând cât mai bine zonele de contur.

6.1 Filtre de tip Kuwahara

Filtrele de tip *Kuwahara* împart vecinătatea pixelului curent (x, y) în regiuni care se suprapun parțial R_1, R_2, $\ldots R_k$, căutând acea regiune care este cea mai uniformă. Ideea principală este aceea de calcula mediile μ_i și variantele σ_i, $i = \overline{1,k}$ ale acestor regiuni:

$$\mu_i = \frac{1}{|R_i|} \sum_{(x,y)\in R_i} f(x, y)$$

$$\sigma_i = \frac{1}{|R_i|} \sum_{(x,y)\in R_i} (f(x, y) - \mu_i)^2$$

f este imaginea originală și $|R_i|$ = numărul de pixeli din regiunea R_i. Apoi se înlocuiește pixelul central cu media regiunii celei mai uniforme (cu varianța cea mai mică). În figura 12 (1) sunt prezentate vecinătățile originale propuse de Kuwahara pentru un pixel. Problema acestor vecinătăți este că nu iau în considerare o vecinătate centrată pe pixelul curent, ceea ce produce o translatare în imagine. Pentru îmbunătățirea rezultatelor a fost propus un set de vecinătăți care iau în considerare și regiunea din jurul pixelului central, precum în figura 12 (2).

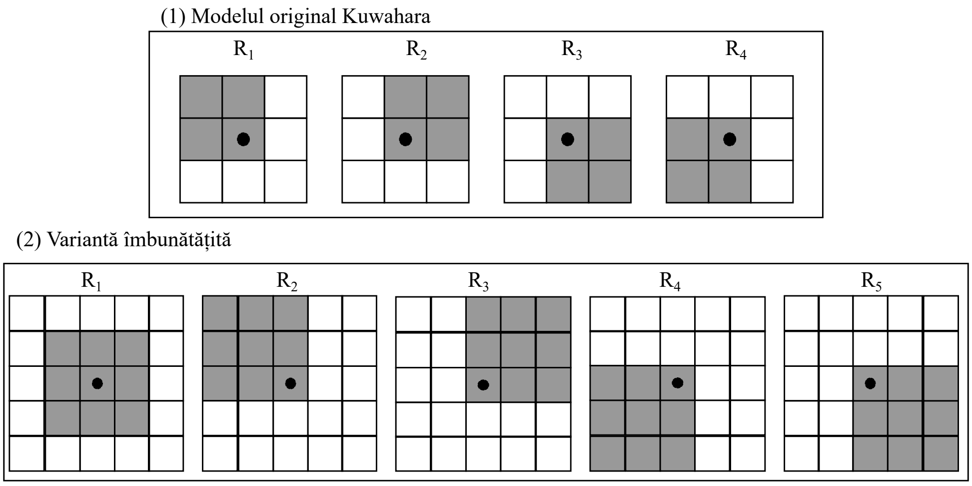


Figura 12: 1) Vecinătățile unui pixel din metoda Kuwahara originală; 2) Vecinătăți mai mari, incluzând o regiune centrată în pixelul curent.

Observații: Filtrele de tip Kuwahara

- sunt de fapt filtre de medie aritmetică - au dezavantajele acestora
- tind să introducă artefacte în zonele uniforme, dacă există zgomot
- pot fi îmbunătățite dacă se introduce drept condiție să se înlocuiască valoarea pixelului considerat prin media calculată, doar dacă varianța într-una dintre regiuni este semnificativ mai mică decât în regiunea centrală R_1.

6.2 Filtrare bilaterală

În cazul filtrelor bilaterale, pentru a realiza o netezire cât mai pronunțată a regiunilor relativ uniforme, dar a păstra regiunile de contur nealterate, ideea a fost aceea de a combina filtrarea liniară bazată pe filtrele de mediere cu filtrarea de tip rang (median). Acest lucru conduce în final la calcularea unor măști de filtrare individuale pentru fiecare pixel, ponderile din mască fiind calculate pe baza similitudinii dintre pixelul central și pixelii din vecinătatea acestuia.

Modelare matematică

În cazul unui filtru de mediere h ponderile $h(i, j)$ ale filtrului depind doar de poziția spațială (i, j) relativ la centrul măștii.

În cazul filtrelor de tip rang h, ponderile $h(i, j)$ se adaptează în funcție de similitudinea pixelului central (x, y) din imaginea f considerată cu vecinul $(x + i, y + j)$. Astfel pixelul rezultat $g(x, y)$ se calculează pe baza formulei:

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

unde h este de fapt funcția după care se calculează coeficientul din mască, de pe poziția (i, j).

În cazul unui filtru de tip rang ponderea prin care un pixel din imagine contribuie în modificarea pixelului central NU depinde de poziționarea acestuia, ci doar de similitudinea cu pixelul central. Evident, că nu are sens considerarea unui filtru global (peste toată imaginea), deoarece pixeli aflați la distanță prea mare de pixelul central nu ar trebui sa contribuie la modificarea acestuia. Din acest motiv filtrele bilaterale țin cont atât de poziție, prin componenta unui filtru de mediere h_d, cât și de similitudine, prin componenta unui filtru de tip rang h_r. Astfel rezultatul $g(x, y)$ al filtrării unei imagini $f(x, y)$ cu un filtru bilateral bazat pe h_d și h_r este dat prin [1]:

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

unde filtrului h_d, de dimensiune $(2k + 1) \times (2k + 1)$, definește vecinătatea considerată pentru pixelul curent (x, y). Astfel valoarea $g(x, y)$ se calculează ca o medie ponderată vecinilor similari cu $f(x, y)$. Cu cât un vecin este mai diferit, cu atât contribuie mai puțin la valoarea rezultat.

Filtrare bilaterală Gaussiană

Un exemplu frecvent utilizat este acela în care pentru ambele filtre se folosesc repartiții Gaussiene. Formulele pot fi considerate:

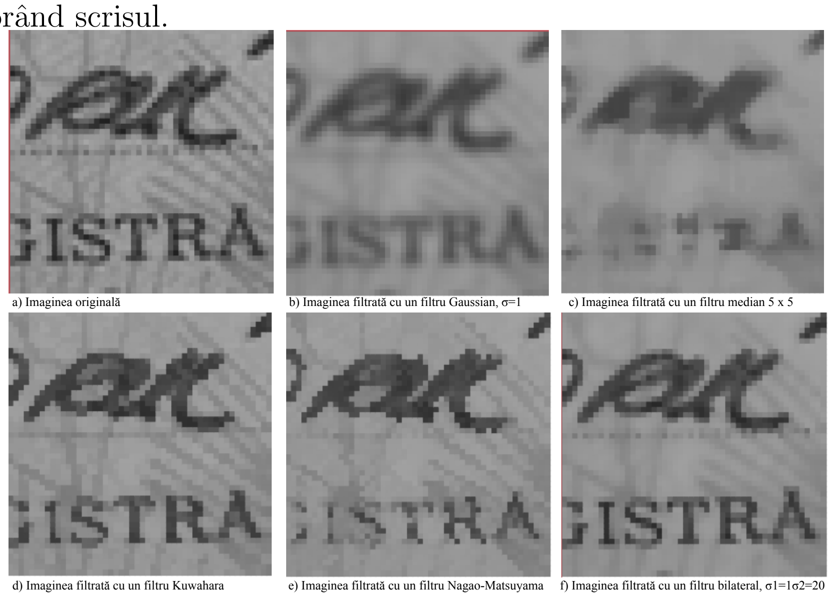
$$h_d^{G,\sigma_d}(i,j) = e^{-\frac{(i^2+j^2)}{2\sigma_d^2}}$$

$$h_r^{G,\sigma_r}(x) = e^{-\frac{x^2}{2\sigma_r^2}}$$

Pe baza acestora rezultatul imaginii filtrate este dat de:

$$g(x,y) = \frac{\sum_{i=-k}^{k}\sum_{j=-k}^{k} f(x+i,y+j)h_d^{G,\sigma_d}(i,j)h_r^{G,\sigma_r}(f(x+i,y+j)-f(x,y))}{\sum_{i=-k}^{k}\sum_{j=-k}^{k} h_d^{G,\sigma_d}(i,j)h_r^{G,\sigma_r}(f(x+i,y+j)-f(x,y))}$$

Exemplu: comparativ de filtrare cu diferite filtre trece jos precum și cu filtre Kuwahara și cu un filtru bilateral. Se observă calitatea superioară a filtrării cu un filtru bilateral. Datorită faptului că e vorba de un text alcătuit din linii relativ subțiri, filtrul median produce rezultate proaste, deteriorând scrisul.



a) Imaginea originală
b) Imaginea filtrată cu un filtru Gaussian, σ=1
c) Imaginea filtrată cu un filtru median 5 x 5
d) Imaginea filtrată cu un filtru Kuwahara
e) Imaginea filtrată cu un filtru Nagao-Matsuyama
f) Imaginea filtrată cu un filtru bilateral, σ1=1σ2=20

To Do:

- Implementați unul dintre algoritmii de filtrare adaptivă pentru imagini color în tonuri de gri sau color în *framework*-ul de laborator.
- Căutați în documentația online funcțiile de binarizare din OpenCv pentru filtrarea bilaterală.
- Comparați rezultatele filtrării unei imagini cu filtre neadaptive versus filtrul bilateral.

Să ne reamintim

- Un filtru adaptiv calculează o mască nouă pentru fiecare pixel (x, y) în funcție de proprietățile statistice locale ale vecinătății pixelului.
- Filtrele adaptive au complexitate de timp semnificativ mai mare decât cele neadaptive.
- Filtrul bilateral modifică pixelul central în funcție de vecinii similari cu acesta, combinând filtrarea liniară cu filtrarea pe baza similitudinii.

Bibliografie

[1] Burger W, Burger M.J „Principals of Digital Image Processing. Core Algorithms”, Springer 2009

[2] Gonzalez R C:, Woods R.E, „Digital Image Processing, global edition”, Perason 2018