

9. Filtrarea imaginilor în domeniul spațial și frecvențial

9.1. Introducere

În această lucrare se va prezenta operatorul de convoluție. Acest operator stă la baza aplicării operațiilor liniare de filtrare a imaginilor aplicate în domeniul spațial (în planul imagine prin manipularea directă a pixelilor din imagine) sau în domeniul frecvențelor (aplicarea unei transformate Fourier, filtrare și apoi aplicarea transformatei Fourier inversă). Exemple de astfel de filtre sunt: filtre trece jos (de netezire a imaginilor, de eliminare a zgomotelor), filtre trece sus (de evidențiere a muchiilor) etc.

9.2. Operația de convoluție în domeniul spațial

Operația de convoluție implică folosirea unei măști/nucleu de convoluție H (de obicei de formă simetrică de dimensiune $w \times w$, cu $w=2k+1$) care se aplică peste imaginea sursă în conformitate cu (9.2).

$$I_D(i, j) = H * I_S \quad (9.1)$$

$$I_D(i, j) = \sum_{u=0}^{w-1} \sum_{v=0}^{w-1} H(u, v) \cdot I_S(i + u - k, j + v - k) \quad (9.2)$$

Aceasta implică parcurgerea imaginii sursă I_S , pixel cu pixel, **ignorând primele și ultimele k linii și coloane** (Fig. 9.1) și calcularea valorii intensității de la locația curentă (i, j) a imaginii de ieșire I_D în conformitate cu (9.2). Nucleul de convoluție se poziționează cu elementul central peste poziția curentă (i, j) .

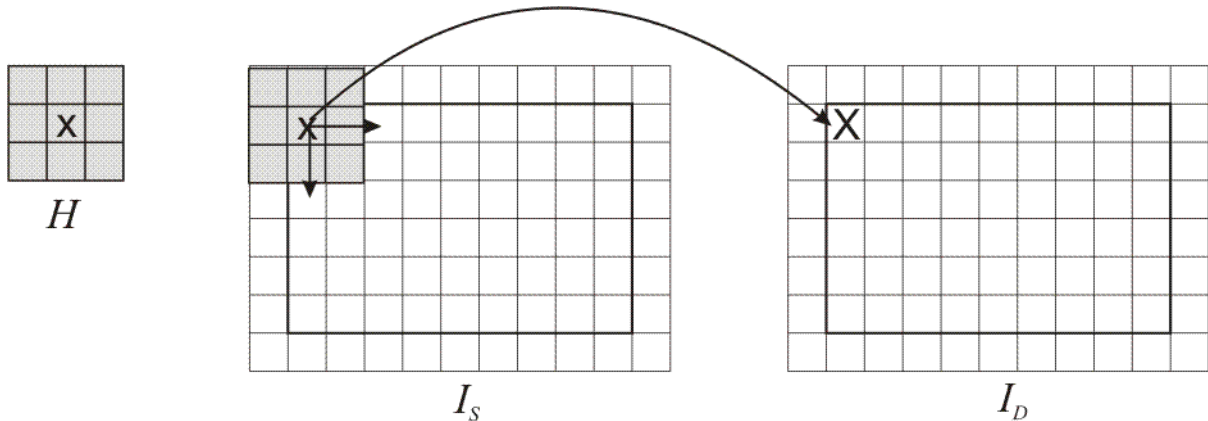


Fig. 9.1 Ilustrarea operației de convoluție.

Nucleele de convoluție pot avea și forme ne-simetrice (elementul central / de referință nu mai este poziționat în centrul de simetrie). Modul de aplicare a operației de convoluție cu astfel de nuclee este similar, dar astfel de exemple nu vor fi prezentate în lucrarea de față.

9.2.1. Filtre de tip „trece-jos”

Aceste nuclee se folosesc pentru operații de netezire și/sau filtrare a zgomotelor (sunt filtre de tip „trece-jos”/”low-pass” care permit trecerea doar a frecvențelor joase – vezi notele de curs). Efectul lor este o mediere a pixelului curent cu valorile vecinilor săi, observabilă prin netezirea (“blur”) a imaginii de ieșire. Aceste nuclee au doar elemente pozitive. Din acest motiv, o practică curentă este împărțirea rezultatului convoluției cu suma elementelor nucleului de

convoluție cu scopul de a scala rezultatul în domeniul de valori al intensității pixelilor din imaginea de ieșire:

$$I_D(i, j) = \frac{1}{c} \sum_{u=0}^{w-1} \sum_{v=0}^{w-1} H(u, v) \cdot I_S(i + u - k, j + v - k) \quad (9.3)$$

unde:

$$c = \sum_{u=0}^{w-1} \sum_{v=0}^{w-1} H(u, v) \quad (9.4)$$

Exemple:

Filtrul medie aritmetică (3x3):

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (9.5)$$

Filtrul gaussian (3x3):

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (9.6)$$

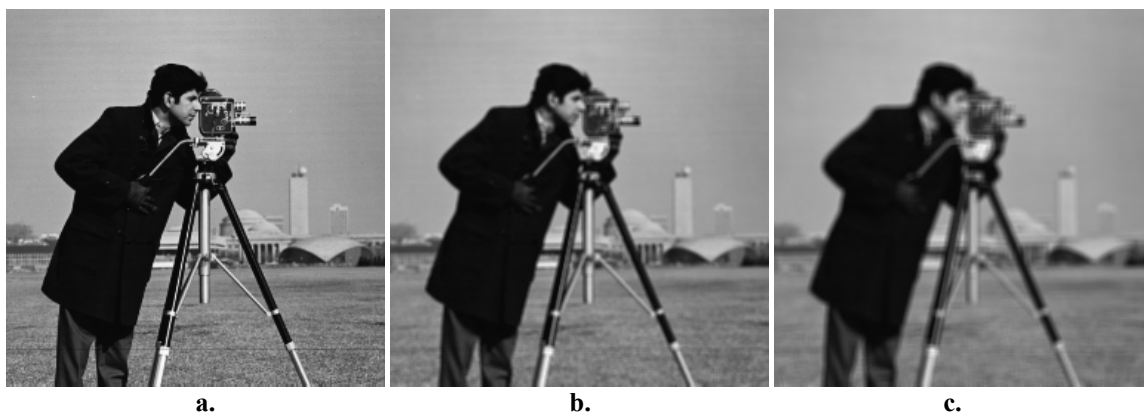


Fig. 9.2 a. Imaginea originală; b. Rezultatul obținut în urma filtrării de tip medie aritmetică cu un nucleu de dimensiune 3x3; c. Rezultatul obținut în urma filtrării de tip medie aritmetică cu un nucleu de 5x5.

9.2.2. Filtre de tip „trece-sus”

Operația de convoluție cu nuclee de acest tip are ca efect punerea în evidență a zonelor din imagine în care există variații bruște ale intensității pixelilor (cum sunt de exemplu muchiile). Ele realizează o filtrare de tip „trece-sus” (vor permite doar trecerea frecvențelor înalte din imagine – vezi note de curs).

Nucleele folosite pentru detecția punctelor de muchii au suma elementelor componente egală cu 0:

Filtre Laplace (dectecție de muchii) (3x3):

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (9.7)$$

sau

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (9.8)$$

Filtre high-pass (trece sus) (3x3):

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (9.9)$$

sau

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (9.10)$$



Fig. 9.3 a. Rezultatul aplicării filtrului Laplace de detecție a muchiiilor (9.8) pe imaginea originală (Fig. 9.2a); b. Rezultatul aplicării filtrului Laplace de detecție a muchiiilor (9.8) pe imaginea din Fig. 9.2b (filtrată în prealabil cu filtrul medie aritmetică); c. Rezultatul obținut în urma filtrării de tip high-pass cu nucleul (9.10).

9.3. Filtrarea imaginilor în domeniul frecvențial

Transformata Fourier discretă (DFT) unidimensională a unui șir format din N numere reale sau complexe este un șir de N numere complexe, date de:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi jkn}{N}}, \quad k = \overline{0 \dots N-1} \quad (9.11)$$

Inversa transformatei Fourier discrete (IDFT) este dată de:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi jkn}{N}}, \quad n = \overline{0 \dots N-1} \quad (9.12)$$

Transformata Fourier discretă bidimensională este obținută prin aplicarea DFT unidimensionale pe fiecare rând al imaginii de intrare și apoi pe fiecare coloană a rezultatului obținut la aplicarea pe linii.

Transformata inversă este obținută prin aplicarea IDFT unidimensionale pe fiecare coloană a imaginii DFT și apoi pe fiecare linie a rezultatului precedent. Setul de numere complexe rezultat după aplicarea DFT poate fi reprezentat și în coordonate polare (magnitudine și fază). Mulțimea magnitudinilor (numere reale) reprezintă spectrul de frecvență (frequency power spectrum) al șirului original.

DFT și inversa ei sunt realizate folosind o abordare recursivă a transformatei Fourier rapide (Fast Fourier Transform), care reduce timpul de calcul de la $O(n^2)$ la $O(n \ln n)$ fapt care reprezintă o creștere a vitezei de calcul, mai ales în cazul procesării imaginilor bidimensionale, la care o complexitate de $O(n^2 m^2)$ ar fi foarte mare în raport cu complexitatea aproape liniară, în număr de pixeli, de $O(nm \ln(nm))$ dată de transformata Fourier rapidă (FFT).

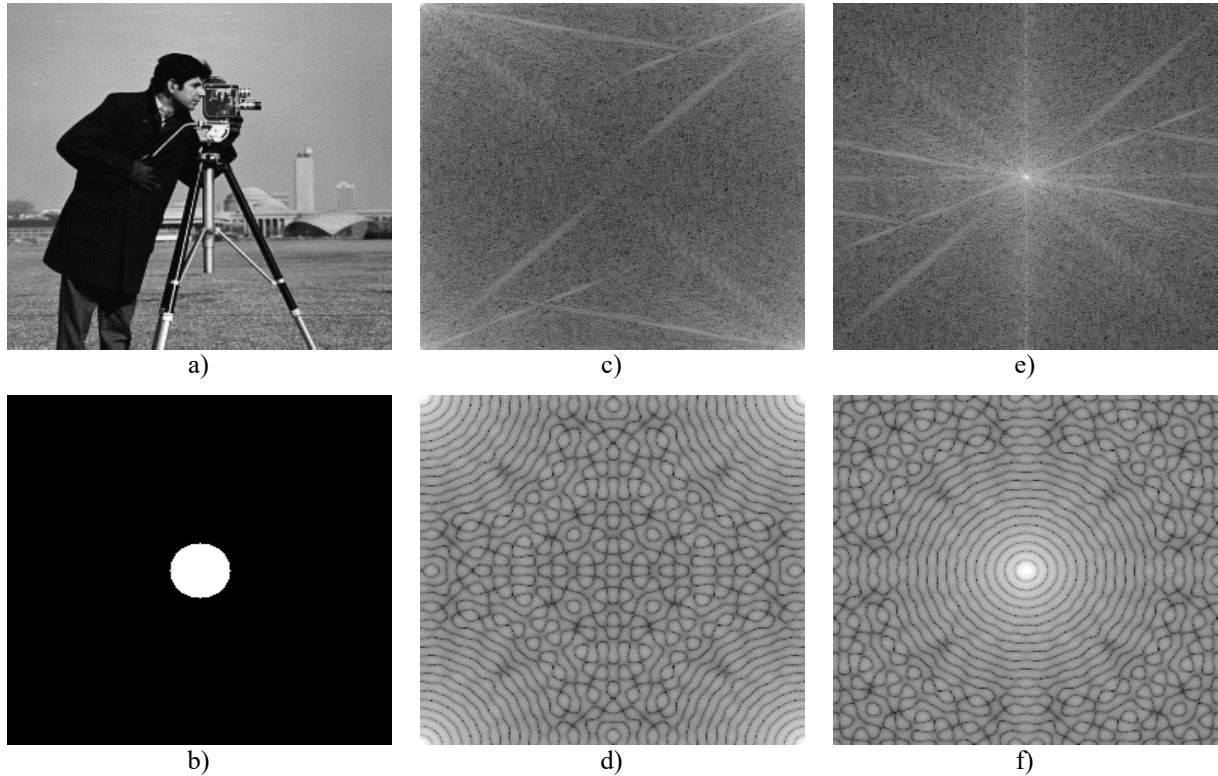


Fig. 9.4 a) și b) imagini originale; c) și d) logaritmul spectrului magnitudinii; e) și f) logaritm centrat al spectrului magnitudinii.

9.3.1. Aliasing

Fenomenul de aliasing este o consecință a limitei frecvenței Nyquist (un semnal eșantionat nu poate reprezenta frecvențe mai mari decât jumătate din frecvența de eșantionare). Astfel, jumătatea de sus a reprezentării în domeniul de frecvență este redundantă. Acest lucru poate fi observat din identitatea:

$$X_k = X_{N-k}^* \quad (9.13)$$

(unde * se referă la conjugata complexă) care este adevărată dacă numerele x_k din șirul de intrare sunt reale. Astfel, spectrul tipic Fourier 1D va conține componentele de frecvență joasă atât în partea de jos cât și în partea de sus, iar frecvențele înalte sunt localizate simetric în raport cu centrul. În spațiul 2D, componentele de frecvență joasă vor fi localizate lângă colțurile

imaginii iar componentele de frecvență înaltă vor fi în centru (vezi **Fig. 9.4c, d**). Acest lucru face ca spectrul să fie destul de greu de citit și de interpretat. Pentru a centra componentele de frecvență joasă în mijlocul spectrului, ca prim pas, ar trebui realizată următoarea transformare a datelor de intrare:

$$x_k \leftarrow (-1)^k x_k \quad (9.14)$$

În spațiul 2D transformarea de centrare devine:

$$x_{uv} \leftarrow (-1)^{u+v} x_{uv} \quad (9.15)$$

După aplicarea acestei transformări în spațiul 1D spectrul va conține în mijloc componentele de frecvență joasă, iar componentele de frecvență înaltă vor fi localizate simetric spre capetele stâng și drept ale spectrului. În 2D, componentele de frecvență joasă vor fi localizate în mijlocul imaginii, în timp ce diversele componente de frecvență înaltă vor fi localizate spre muchii.

Magnitudinile localizate pe orice linie care trece prin centrul imaginii DFT reprezintă componentele 1D din spectrul de frecvență al imaginii originale, pe direcția liniei. Toate liniile de acest fel sunt simetrice în raport cu mijlocul (centrul imaginii).

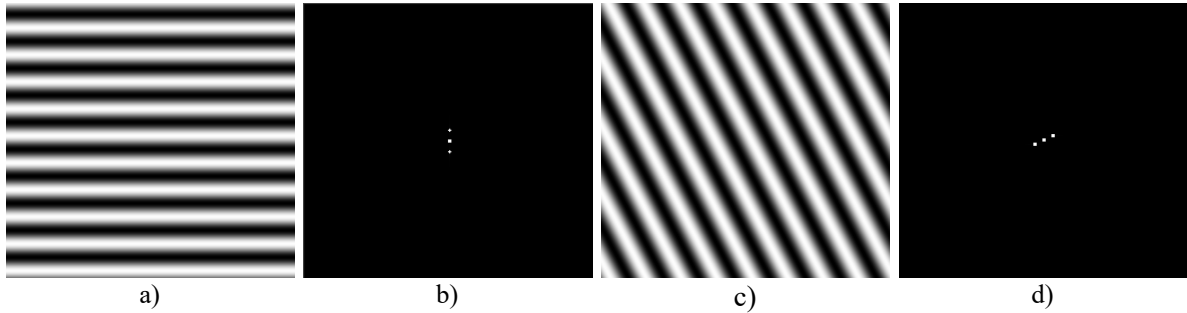


Fig. 9.5 Transformate Fourier ale imaginilor cu unde sinusoidale a) și c). Punctul de centru în b) și d) reprezintă componenta continuă, celelalte două puncte simetrice se datorează frecvenței undelor sinusoidale.

9.3.2. Filtre ideale de tip „trece-jos” și „trece-sus”, în domeniul frecvențial

Operația de convoluție în domeniul spațial este echivalentă cu înmulțirea scalară în domeniul frecvențial. Astfel, pentru nuclee de convoluție mari, este mai convenabil din punct de vedere computațional să se realizeze operația de convoluție în domeniul frecvențial.

Algoritmul de filtrare în domeniul frecvențial este următorul:

- Se realizează transformata de centrare a imaginii pe imaginea originală (9.15).
- Se realizează transformata DFT.
- Se schimbă coeficienții Fourier în funcție de filtrarea dorită.
- Se realizează transformata IDFT.
- Se realizează transformata de centrare a imaginii (anulează efectul primei centrări a imaginii).

Un filtru ideal de tip „trece-jos” va modifica toți coeficienții Fourier care sunt mai departe de centrul imaginii ($W/2, H/2$) decât o distanță R dată. Acești coeficienți vor primi valoarea 0 (W este lățimea imaginii și H este înălțimea imaginii):

$$X'_{uv} = \begin{cases} X_{uv} , & \left(\frac{H}{2} - u \right)^2 + \left(\frac{W}{2} - v \right)^2 \leq R^2 \\ 0 , & \left(\frac{H}{2} - u \right)^2 + \left(\frac{W}{2} - v \right)^2 > R^2 \end{cases} \quad (9.16)$$

Un filtru ideal de tip „trece-sus” va schimba în 0 toți coeficienții Fourier aflați la o distanță mai mică decât R față de centrul imaginii ($W/2, H/2$).

$$X'_{uv} = \begin{cases} X_{uv}, & \left(\frac{H}{2} - u\right)^2 + \left(\frac{W}{2} - v\right)^2 > R^2 \\ 0, & \left(\frac{H}{2} - u\right)^2 + \left(\frac{W}{2} - v\right)^2 \leq R^2 \end{cases} \quad (9.17)$$

Rezultatele aplicării unui filtru ideal de tip „trece-jos” și de tip „trece-sus” sunt prezentate în **Fig. 9.6 b)** și **c)**. Din păcate, filtrele spațiale corespunzătoare din **Fig. 9.6 e)** și **g)** nu sunt FIR (au un suport infinit) și oscilează îndepărtându-se de centrele lor. Din această cauză imaginile rezultate după aplicarea celor două filtre (trece-sus și trece-jos) au un aspect de undă circulară. Pentru a corecta acest lucru, tăierea (suprimarea) în domeniul frecvențial trebuie să fie mai netedă, așa cum este prezentat în secțiunea următoare.

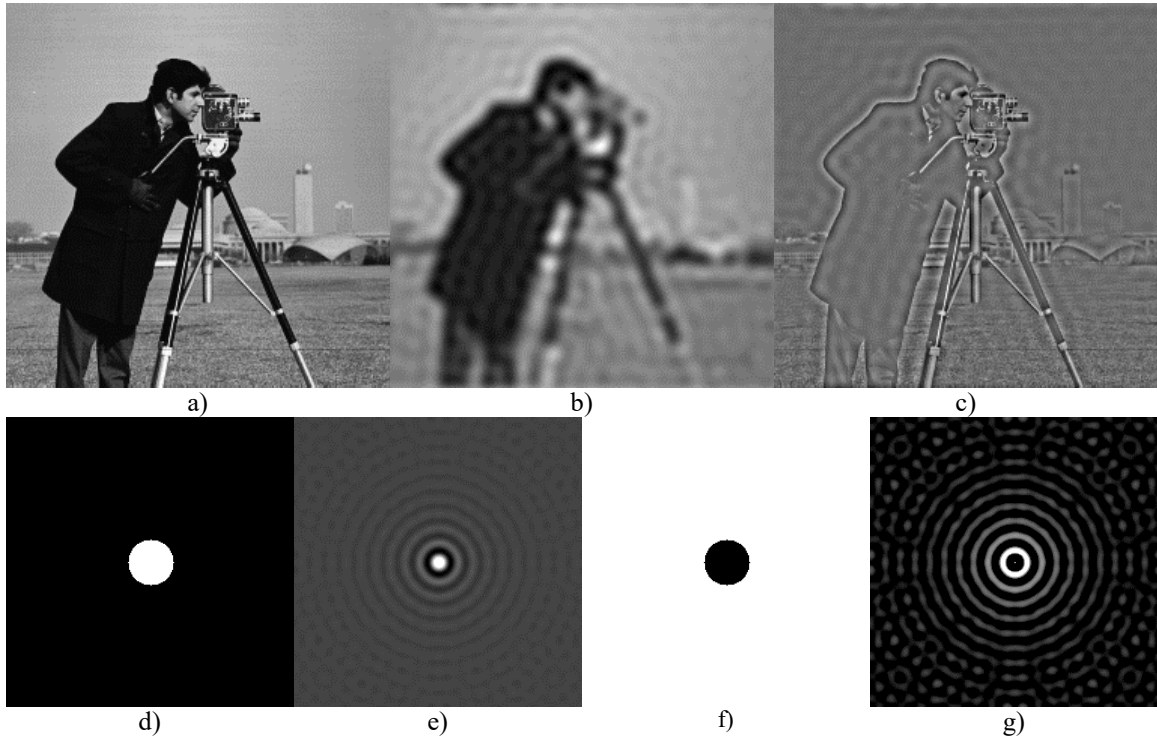


Fig. 9.6 a) imaginea originală; b) rezultatul aplicării filtrului ideal de tip „trece-jos”; c) rezultatul aplicării filtrului ideal de tip „trece-sus”; d) filtru ideal de tip „trece-jos” în domeniul frecvențial; e) filtru ideal de tip „trece-jos” corespunzător în domeniul spațial; f) filtru ideal de tip „trece-sus” în domeniul frecvențial; g) filtru ideal de tip „trece-sus” corespunzător în domeniul spațial. ($R = 20$)

9.3.3. Filtru Gaussian de tip „trece-jos” și „trece-sus” în domeniul frecvențial

În cazul filtrului de tip Gauss, coeficienții de frecvență nu sunt tăiați brusc, ci este folosit un proces de suprimare mai netedă. Acest proces ține cont și de faptul că DFT a unei funcții de tip Gauss este tot o funcție de tip Gauss. (**Fig. 9.7d-g**).

Filtrul Gaussian de tip „trece-jos” atenuează componentele din domeniul de frecvență care sunt mai îndepărtate față de centrul imaginii ($W/2, H/2$). $A \sim \frac{1}{\sigma}$ unde σ este deviația standard a filtrului Gaussian echivalent în domeniul spațial.

$$X'_{uv} = X_{uv} e^{-\frac{\left(\frac{H}{2}-u\right)^2 + \left(\frac{W}{2}-v\right)^2}{A^2}} \quad (9.18)$$

Filtrul Gaussian de tip „trece-sus” atenuează componentele de frecvență care sunt aproape de centrul imaginii ($W/2, H/2$):

$$X'_{uv} = X_{uv} \left(1 - e^{-\frac{\left(\frac{H}{2}-u\right)^2 + \left(\frac{W}{2}-v\right)^2}{A^2}} \right) \quad (9.19)$$

Fig. 9.7 arată rezultatele aplicării unui filtru de tip Gauss. A se remarca faptul că efectul de unde circulare vizibil în **Fig. 9.6** a dispărut.

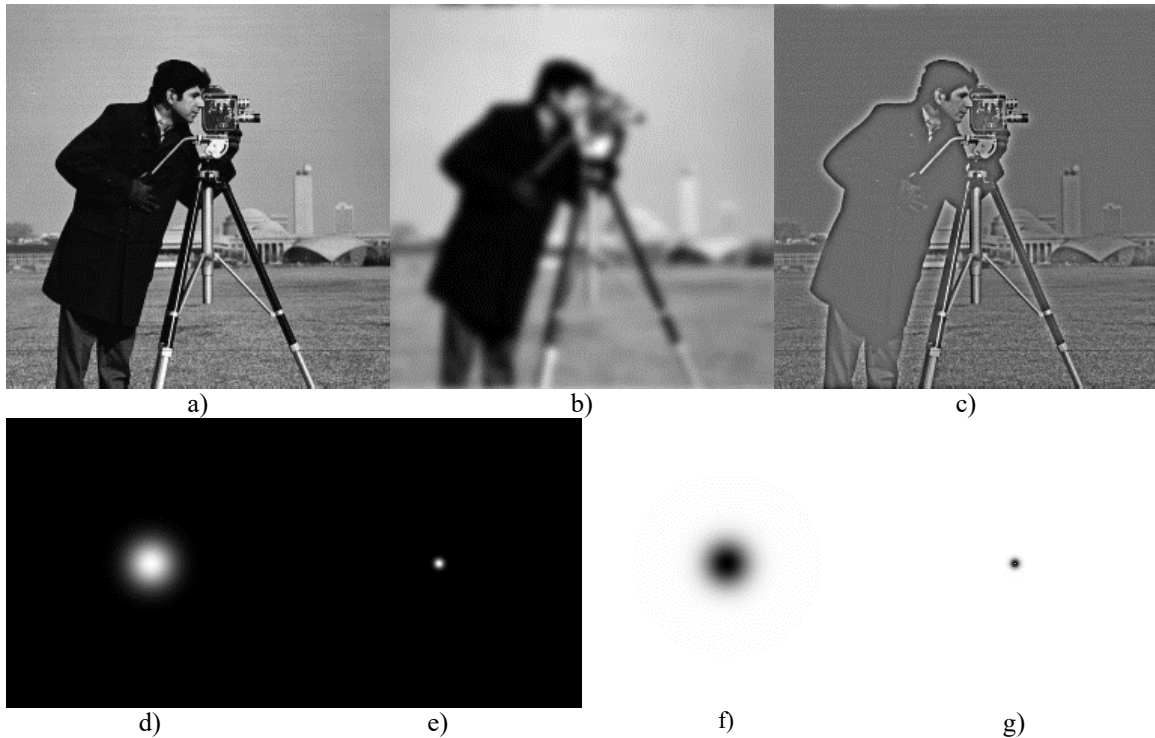


Fig. 9.7 a) imaginea originală; b) rezultatul aplicării unui filtru Gaussian de tip „trece-jos”; c) rezultatul aplicării unui filtru Gaussian de tip „trece-sus”; d) filtru Gaussian de tip „trece-jos” în domeniul frecvențial; e) filtru Gaussian corespunzător de tip „trece-jos” în domeniul spațial; f) filtru Gaussian de tip „trece-sus” în domeniul frecvențial; g) filtru Gaussian corespunzător de tip „trece-sus” în domeniul spațial. ($A = 20$)

9.4. Detalii de implementare

9.4.1. Filtre în domeniul spațial

Filtrele de tip „trece-jos” vor avea întotdeauna coeficienți pozitivi și astfel imaginea rezultată după aplicarea filtrului va conține valori pozitive. **Trebuie să vă asigurați că imaginea rezultată are valori cuprinse în intervalul dorit (în cazul nostru 0-255).** Pentru a realiza acest lucru, trebuie să vă asigurați că suma coeficienților din filtrul trece jos este 1. **Dacă folosiți operații întregi să fiți atenți la ordinea operațiilor! În mod normal, împărțirea este ultima operație care ar trebui efectuată, pentru a minimiza erorile datorate rotunjirii.**

Filtrele de tip „trece-sus” vor avea coeficienți pozitivi și negativi. **Trebuie să vă asigurați că valorile din imaginea rezultat sunt numere întregi cuprinse în intervalul 0 și 255!**

Există trei posibilități pentru a vă asigura că imaginea rezultat este în intervalul dorit. Prima metodă presupune calculul următor:

$$S_+ = \sum_{F_k > 0} F_k, \quad S_- = \sum_{F_k < 0} -F_k,$$

$$S = \frac{1}{2 \max\{S_+, S_-\}}$$

$$I_D(u, v) = S(F * I_S)(u, v) + \left\lfloor \frac{L}{2} \right\rfloor \quad (9.20)$$

În formula de mai sus, S_+ reprezintă suma coeficienților pozitivi din filtru, și S_- suma valorilor absolute a coeficienților negativi. Rezultatul operației $F * I_S$, adică al aplicării filtrului de tip „trece-sus” este întotdeauna în intervalul $[-LS_-, LS_+]$ unde L este nivelul maxim de gri din imagine (255). Înmulțirea cu S va plasa rezultatul în intervalul $[-L/2, L/2]$, iar adunarea cu $L/2$ va muta rezultatul în intervalul $[0, L]$.

O altă abordare constă în realizarea tuturor operațiilor folosind numere întregi cu semn (rezultă imaginea $F * I_S$), apoi determinarea valorii minime (min) și maxime (max) din rezultat (din $F * I_S$) urmată de o transformare liniară a valorilor rezultate folosind formula:

$$I_D(u, v) = L \cdot \frac{(F * I_S)(u, v) - \min}{\max - \min} \quad (9.21)$$

A treia abordare calculează magnitudinea rezultatului și saturează toate valorile care depășesc domeniul de reprezentare $[0, L]$.

9.4.2. Filtre în domeniul frecvențial

Pentru vizualizare și procesare în domeniul frecvențial este util să considerăm o reprezentare care conține coeficientul (0,0) în centrul imaginii. Acest lucru se poate realiza prin interschimbarea cadranelor opuse din transformata Fourier. În mod echivalent se poate preprocesa imaginea de intrare folosind 9.15. Filtrul prezentat mai jos folosește următoarea funcție pentru a realiza această operație de centrare atât pe imaginea de intrare cât și pe rezultat.

```
void centering_transform(Mat img){
    // imaginea trebuie să aibă elemente de tip float
    for (int i = 0; i < img.rows; i++){
        for (int j = 0; j < img.cols; j++){
            img.at<float>(i, j) = ((i + j) & 1) ? -img.at<float>(i, j) : img.at<float>(i, j);
        }
    }
}
```

Librăria OpenCV conține o funcție pentru calcularea Transformării Fourier Discrete. Următorul cod șablon ilustrează cum se face atât transformarea directă cât și cea inversă. Procesările în domeniul frecvențial se vor realiza pe canalul de magnitudine. Fiindcă DFT-ul lucrează cu imagini care au dimensiuni egale cu puteri ale lui doi, se va lucra cu imaginea *cameraman.bmp*.

```
Mat generic_frequency_domain_filter(Mat src){
    // imaginea trebuie să aibă elemente de tip float
    Mat srcf;
    src.convertTo(srcf, CV_32FC1);
```



```

//transformarea de centrare
centering_transform(srcf);

//aplicarea transformatei Fourier, se obține o imagine cu valori numere complexe
Mat fourier;
dft(srcf, fourier, DFT_COMPLEX_OUTPUT);
//divizare în două canale: partea reală și partea imaginară
Mat channels[] = { Mat::zeros(src.size(), CV_32F), Mat::zeros(src.size(), CV_32F) };
split(fourier, channels); // channels[0] = Re(DFT(I)), channels[1] = Im(DFT(I))

//calcularea magnitudinii și fazei în imaginile mag, respectiv phi, cu elemente de tip float
Mat mag, phi;
magnitude(channels[0], channels[1], mag);
phase(channels[0], channels[1], phi);

//aici afișați imaginile cu fazele și magnitudinile
// .....

//aici inserați operații de filtrare aplicate pe coeficienții Fourier
// .....

//memorați partea reală în channels[0] și partea imaginară în channels[1]
// .....

//aplicarea transformatei Fourier inversă și punerea rezultatului în dstf
Mat dst, dstf;
merge(channels, 2, fourier);
dft(fourier, dstf, DFT_INVERSE | DFT_REAL_OUTPUT | DFT_SCALE);

//transformarea de centrare inversă
centering_transform(dstf);

//normalizarea rezultatului în imaginea destinație
normalize(dstf, dst, 0, 255, NORM_MINMAX, CV_8UC1);
//Notă: normalizarea distorsionează rezultatul oferind o afișare îmbunătățită în intervalul
//[0,255]. Dacă se dorește afișarea rezultatului cu exactitate (vezi Activitatea 3) se va
//folosi în loc de normalizare conversia:
//dstf.convertTo(dst, CV_8UC1);

return dst;
}

```

9.5. Activități practice

1. Implementați un filtru general care realizează operația de convoluție cu un nucleu de convoluție oarecare. Coeficientul de scalare trebuie calculat automat. Acesta este inversul sumei coeficienților pentru filtre trece-jos, iar pentru filtrele trece-sus se calculează folosind ecuația (9.20).
2. Testați filtrul folosind nucleele din ecuațiile (9.5)...(9.10)
3. Studiați funcția șablon dată pentru un filtru în domeniul frecvențial. Realizați trecerea unei imagini sursă din domeniul spațial în domeniul frecvențial aplicând transformata Fourier directă (DFT), iar apoi aplicați transformarea Fourier inversă (IDFT) pe coeficienții spectrului Fourier obținuți și verificați că imaginea rezultată este identică cu imaginea sursă.
4. Adăugați o funcție de procesare care calculează și afișează logaritmul magnitudinii transformatei Fourier a unei imagini de intrare. Adăugați 1 la magnitudine pentru a evita $\log(0)$.
5. Adăugați funcții de procesare care aplică: un filtru ideal trece-jos, un filtru ideal trece-sus, un filtru Gaussian trece-jos și un filtru Gaussian trece-sus în domeniul frecvențial utilizând ecuațiile (9.16)...(9.19).
6. **Salvați-vă ceea ce ați lucrat. Utilizați aceeași aplicație în laboratoarele viitoare. La sfârșitul laboratorului de procesare a imaginilor va trebui să prezentați propria aplicație cu algoritmi implementați!!!**

9.6. Bibliografie

- [1] Umbaugh Scot E, *Computer Vision and Image Processing*, Prentice Hall, NJ, 1998, ISBN 0-13-264599-8.
- [2] R.C.Gonzales, R.E.Woods, *Digital Image Processing, 2-nd Edition*, Prentice Hall, 2002.