

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 muchiiilor) 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).

$$I_D = H * I_S \tag{9.1}$$

$$I_D(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k H(i, j) \cdot I_S(x+i, y+j), \quad x=0..Height-1, \quad y=0..Width-1 \tag{9.2}$$

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 (x, y) a imaginii de ieșire I_D în conformitate cu (9.2). Nucleul de convoluție se poziționează cu elementul central peste poziția curentă (x, y) .

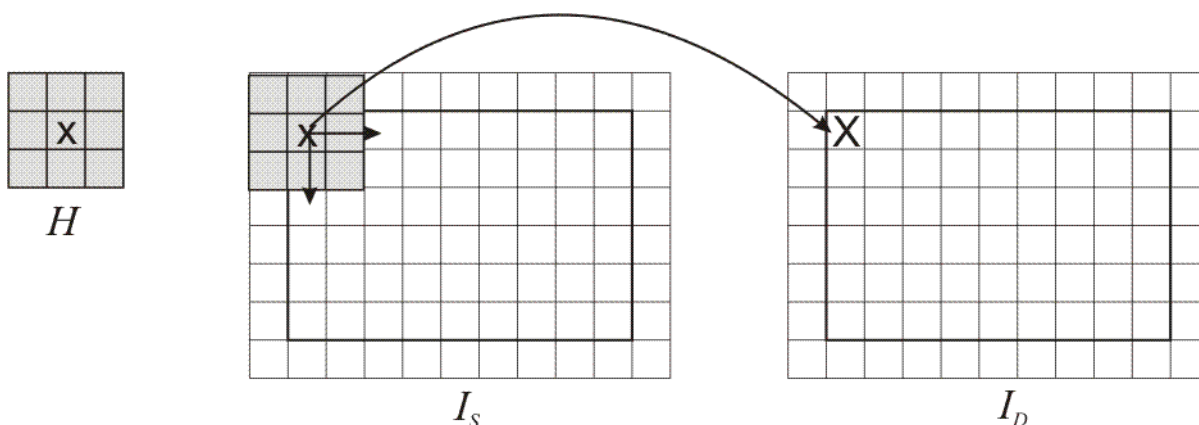


Fig. 9.1 Ilustrarea operației de convoluție

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:

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

Unde:

$$c = \sum_{i=-k}^k \sum_{j=-k}^k H(i, j) \quad (9.4)$$

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 nulă:

Filtre Laplace (deteecție 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

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

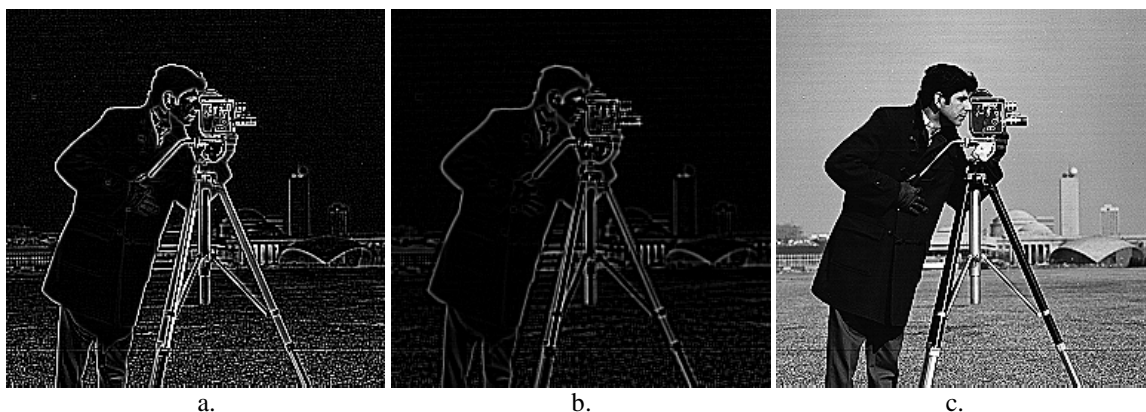


Fig. 9.3 a. Rezultatul aplicării filtrului Laplace de deteectie a muchiiilor (9.8) pe imaginea originală (Fig. 9.2a); b. Rezultatul aplicării filtrului Laplace de deteectie a muchiiilor (9.8) pe imaginea din Fig. 9.2b (filtrată în prealabil cu filtrul medie aritmetică); c. Rezultatul obținut în urma 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 $O(nm \ln(m+n))$ dată de transformata Fourier rapidă (FFT).

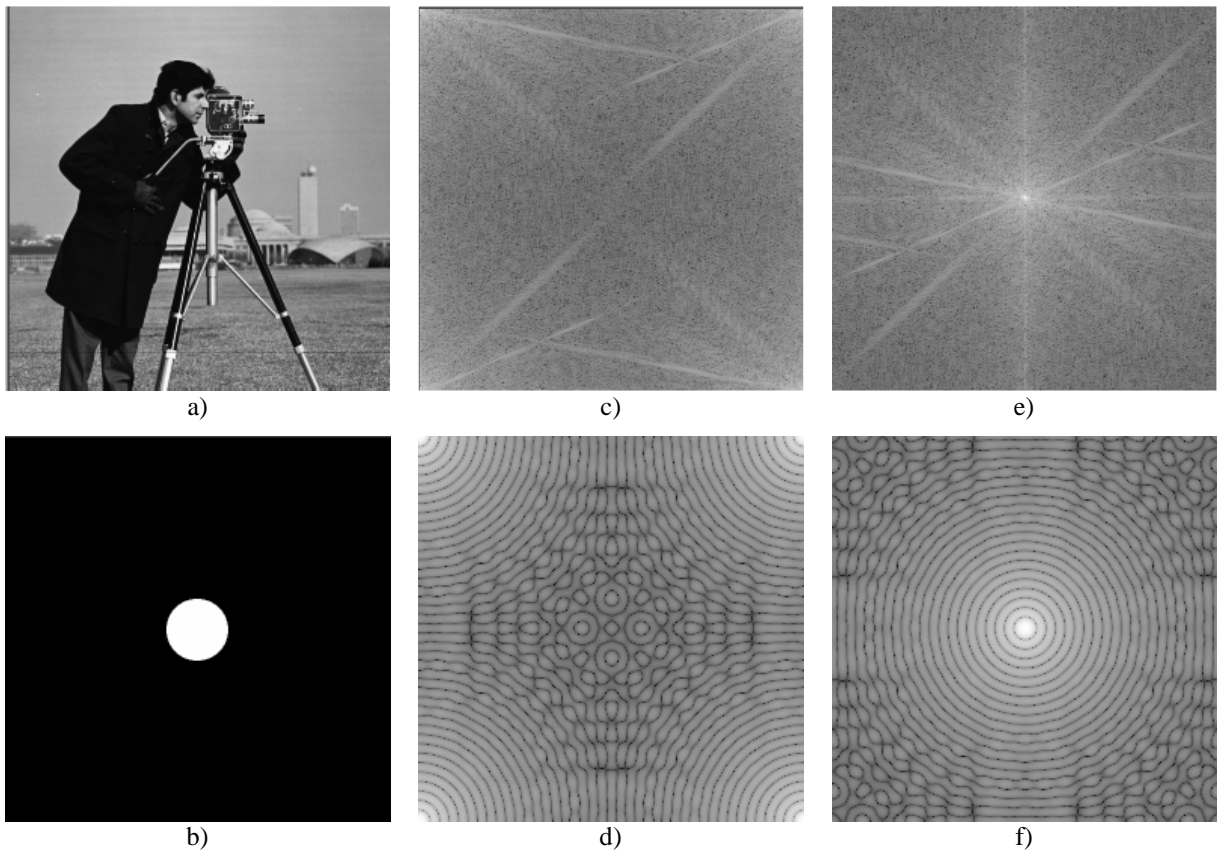


Fig. 9.4 a) și b) imagini originale; c) și d) logaritmul spectrului magnitudinii; e) și f) logaritm centrat al spectrului magnitudinii

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:

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

După aplicarea acestei transformări în spațiul 1D spectrul va conține în mijloc componentele de frecvență joasă, și 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).

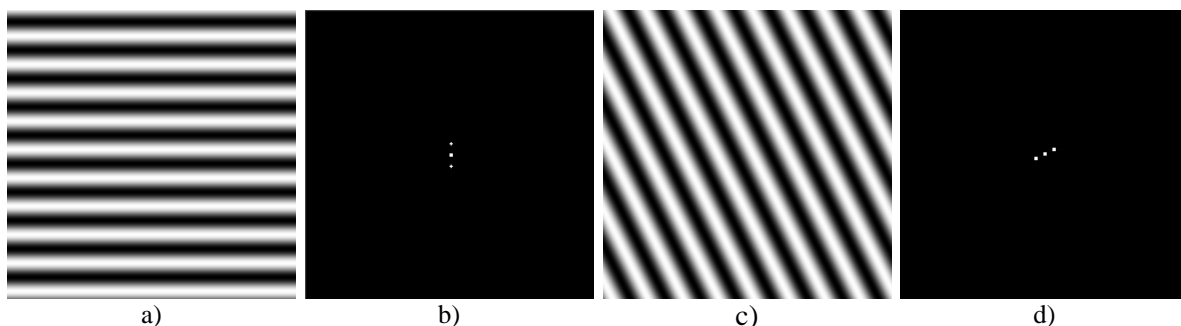


Fig. 9.5 Transformate Fourier ale imaginilor cu unde sinusoidale a) și c). Punctul de centru în b) și d) reprezintă componenta continuă, celelalte două puncte simetrice se datorează frecvenței undelor sinusoidale.

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:

- a) Se realizează transformata de centrare a imaginii pe imaginea originală (9.15)
- b) Se realizează transformata DFT
- c) Se schimbă coeficienții Fourier în funcție de filtrarea dorită
- d) Se realizează transformata IDFT
- e) 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 d) 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.

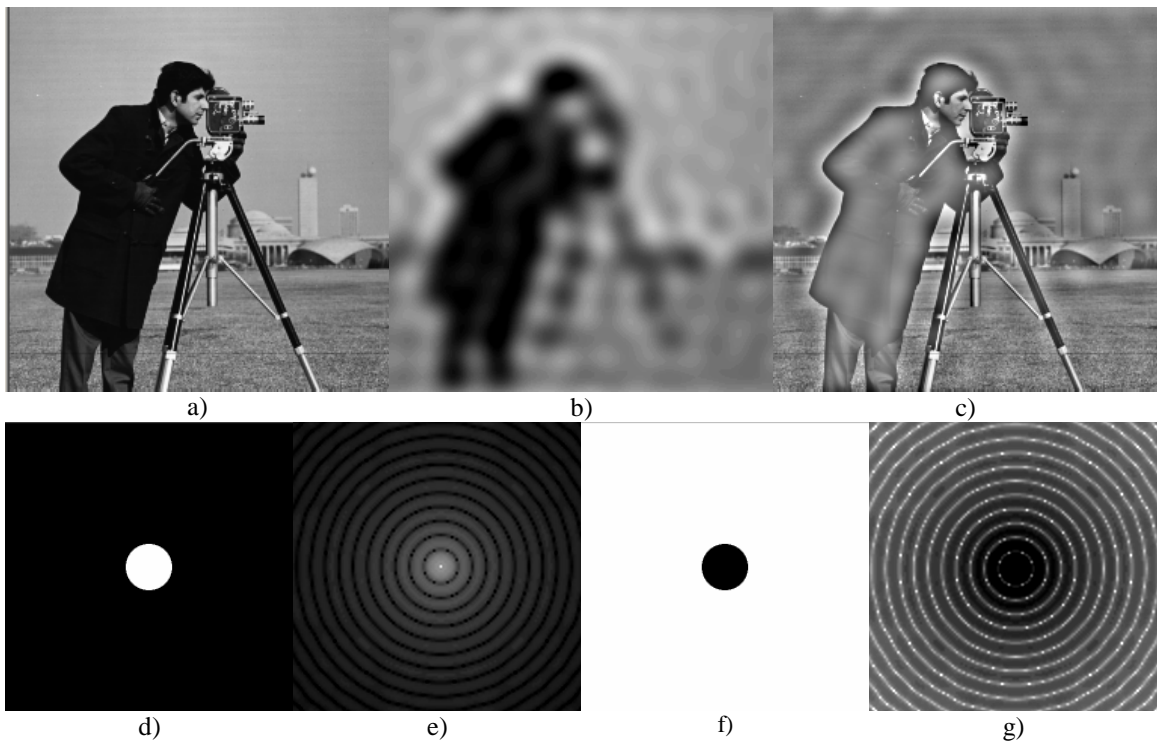


Fig. 9.6 a) imaginea originală; b) rezultatul aplicării filtrului ideal de tip „trece-jos”; c) rezultatul aplicării filtrului ideal de tip „trece-sus”; d) filtru ideal de tip „trece-jos” în domeniul frecvențial; e) filtru ideal de tip „trece-jos” corespunzător în domeniul spațial; f) filtru ideal de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

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.

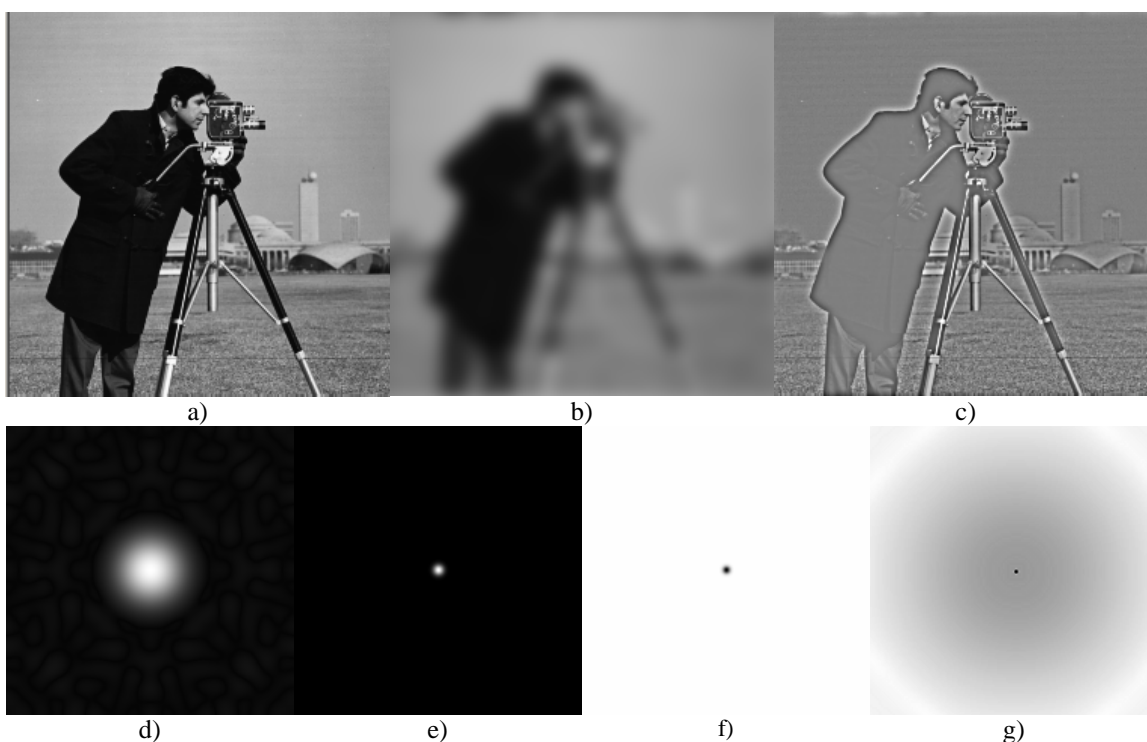


Fig. 9.7 a) imaginea originală; b) rezultatul aplicării unui filtru Gaussian de tip „trece-jos”; c) rezultatul aplicării unui filtru Gaussian de tip „trece-sus”; d) Filtru Gaussian de tip „trece-jos” în domeniul frecvențial; e) filtru Gaussian corespunzător de tip „trece-jos” în domeniul spațial; f)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

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:

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

$$S = \frac{1}{2 \max\{S_+, S_-\}} \quad (9.20)$$

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

În formula de mai sus, S_+ reprezintă suma coeficienților pozitivi din filtru, și S_- suma valorilor absolute a coeficienților negativi. Rezultatul aplicării filtrului de tip „trece-sus” este întotdeauna în intervalul $[-LS_-, LS_+]$ unde L este nivelul maxim de gri din imagine (255). Rezultatul acestei transformări va plasa scala rezultatului la $[-L/2, L/2]$ și apoi va muta nivelul 0 la $L/2$.

O altă abordare constă în realizarea tuturor operațiilor folosind numere întregi cu semn, apoi determinarea valorii minime și maxime din rezultat urmată de o transformare liniară a valorilor rezultate folosind formula:

$$D = \frac{L(S - \min)}{\max - \min} \quad (9.21)$$

A treia abordare calculează magnitudinea rezultatului și saturează toate valorile care depășesc nivelul maxim L .

9.4.2. Filtre în domeniul frecvențial

Pentru realizarea transformatei Fourier rapide veți avea la dispoziție o bibliotecă și un fișier header .h. Biblioteca se numește „dibfft.lib” iar fișierul header este „dibfft.h”. Pentru a folosi biblioteca trebuie să copiați fișierele „dibfft.lib” și „dibfft.h” în directorul „Diblook”.

Apoi faceți click dreapta pe proiectul „Diblook” din fereastra de lucru, selectați meniul „Add > Existing Item...”. Se va deschide automat fereastra „Add Existing Item - DibLook”. Va trebui să selectați și să adăugați fișierul „dibfft.h” la proiect.

Apoi se va include biblioteca „dibfft.lib” la linker. Pentru aceasta faceți click dreapta pe proiectul „Diblook” din fereastra de lucru și alegeți „Properties”. Se va deschide automat fereastra „DibLook Property Pages”. Va trebui să selectați la secțiunea „Configuration” opțiunea „All Configurations” și să adăugați biblioteca „dibfft.lib” la secțiunea „Linker” (vezi Fig. 9.8!).

De asemenea se va include header-ul „dibfft.h” în zona de „#include” de la începutul fișierului „dibview.cpp”.

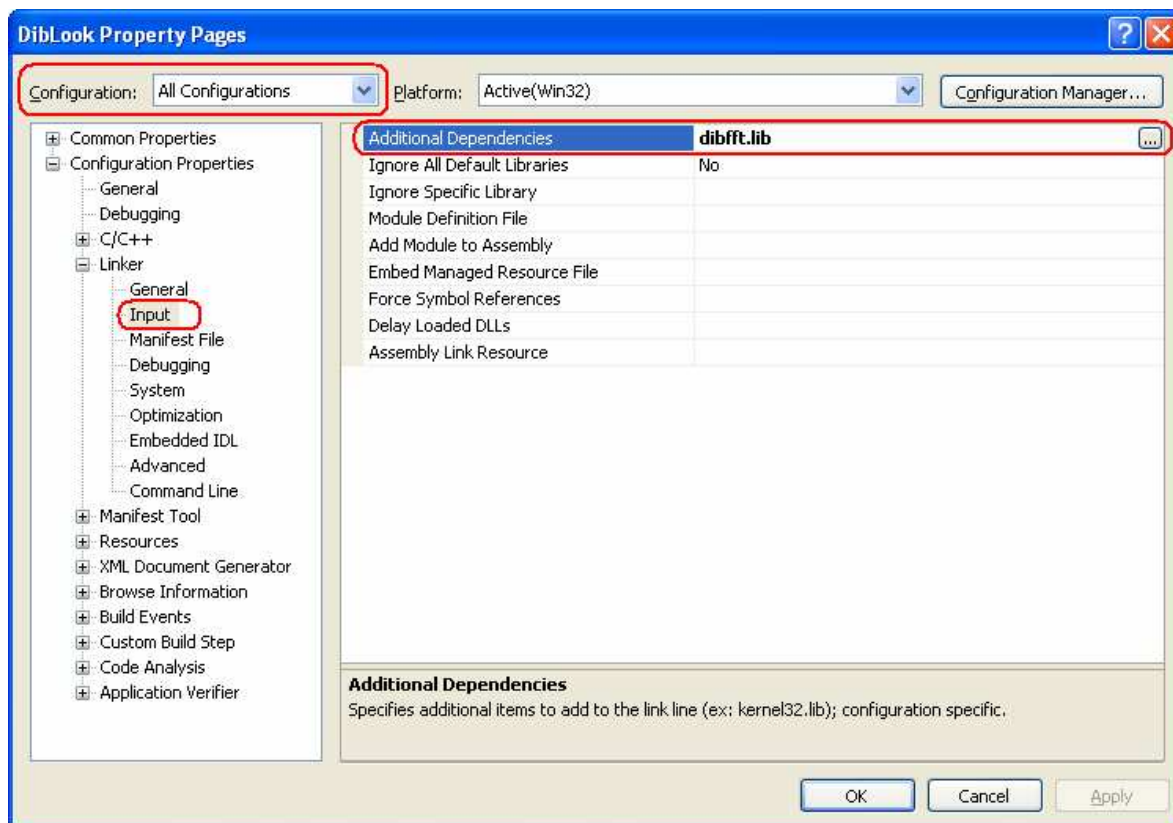


Fig. 9.8 Adăugarea unei biblioteci (.lib) la proiect

Biblioteca va furniza următoarele funcții:

```

/*Primii doi parametri sunt lățimea și înălțimea imaginii
Aceste funcții funcționează în mod corect numai pentru imagini care au lățimea și
înălțimea puteri ale lui 2 și >=4.
Parametrii 3 și 4 sunt partea reală și imaginară ale șirului de intrare care
conține lățime*înălțime valori. Tipurile acceptate sunt unsigned char (BYTE),
float și double. Partea imaginară este opțională, se poate specifica (T*)0 pentru a
anula existența acesteia, toate valorile fiind considerate 0.
Parametrii 5 și 6 sunt partea reală și imaginară ale șirului de ieșire. Tipurile
acceptate sunt unsigned char (BYTE), float și double. Partea imaginară este
opțională, se poate specifica (T*)0 pentru a anula existența acesteia.

/*realizează FFT pe randurile imaginii */
template<class T> void fftrows(int width, int height, const T *ix, const T *iy,
double *ox, double *oy);

/*realizează IFFT pe rândurile imaginii */
template<class T> void ifftrows(int width, int height, const double *ix, const
double *iy, T *ox, T *oy);

/*aplică FFT pe coloanele imaginii */
template<class T> void fftcols(int width, int height, const T *ix, const T *iy,
double *ox, double *oy);

/*aplică IFFT pe coloanele imaginii */
template<class T> void ifftcols(int width, int height, const double *ix, const
double *iy, T *ox, T *oy);

/*aplică FFT pe imagine */
template<class T> void fftimage(int width, int height, const T *inpx, const T*
inpy, double *ox, double *oy);

/*aplică IFFT pe imagine */
template<class T> void ifftimage(int width, int height, const double *ix, const
double *iy, T *outpx, T *outpy);
    
```

Funcțiile date sunt de tip template și se aplică pe intrări/iesiri BYTE, float și double. Părțile imaginare sunt opționale atât pentru intrări cât și pentru ieșiri. Folosiți pointeri NULL pentru a specifica intrări/ieșiri care lipsesc.

Următoarea secțiune de cod oferă un exemplu al aplicării FFT urmat de IFFT. Se obține imaginea originală.

```
BEGIN_PROCESSING();
double *real= new double[dwWidth*dwHeight];
double *imag= new double[dwWidth*dwHeight];
fftimage(dwWidth, dwHeight, lpSrc, (BYTE*)0, real, imag);
ifftimage(dwWidth, dwHeight, real, imag, lpDst, (BYTE*)0);
END_PROCESSING("FFT");
```

Câteva aspecte importante legate de lucrul cu valori în domeniul frecvențial:

1. Pentru FFT folosiți întotdeauna doar imagini grayscale având atât lățimea cât și înălțimea puteri ale lui 2 (de exemplu imaginea "cameraman.bmp" care are width=height=256=2⁸ pixeli)
2. Întotdeauna realizați transformata de centrare (9.15) înainte de aplicarea FFT și după aplicarea IFFT.

```
for(int i=0;i< dwHeight;i++) {
    for(int j=0;j< dwWidth;j++) {
        D[i*w+j] = ((i+j)&1)?-S[i*w+j]:S[i*w+j];
```

3. Coeficientul (componenta continuă) Fourier DC(0,0) îi domină pe ceilalți coeficienți. La afișarea magnitudinii coeficienților Fourier este bine să folosiți logaritm din valoarea modulului + 1 ! Trebuie să determinați valoarea maximă a logaritmului și să scalați valorile pentru a se potrivi domeniului de lucru (0-255).
4. Transformata Fourier a unei imagini este un șir complex de valori în virgulă mobilă ! Stocați atât valorile reale cât și cele imaginare ca valori în virgulă mobilă. La conversia înapoi, în domeniul spațial, valorile imaginare sunt îndepărtate (pentru filtre uzuale ar trebui să fie 0)

9.5. Activități practice

1. Implementați operația de convoluție folosind nucleele din ecuațiile (9.5)...(9.10)
2. Implementați un operator generalizat de convoluție de dimensiune 3x3 care să utilizeze valori introduse de utilizator într-o fereastră de dialog. Coeficientul de scalare trebuie calculat automat. Acesta este inversul sumei coeficienților din filtrul trece jos și pentru filtrele de tip „trece-sus” se calculează folosind ecuația (9.20)
3. Importați biblioteca *dibfft* în aplicația Diblook (vedeți secțiunea 9.4.2). Adăugați o funcție care aplică FFT unei imagini de intrare și transformă rezultatul înapoi în domeniul spațial folosind IFFT. Constatați că destinația este identică cu sursa.
4. Adăugați o funcție de procesare care calculează și afișează logaritmul magnitudinii transformatei Fourier a unei imagini de intrare.
5. Adăugați o funcție de procesare care aplică un filtrul ideal de tip trece-jos, trece-sus și apoi filtrul Gauss în domeniul frecvențial - 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!!!**

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