

Lucrarea 7

Filtrarea imaginilor

BREVIAR TEORETIC

Filtrarea imaginilor se înscrie în clasa operațiilor de îmbunătățire, principalul scop al acesteia fiind eliminarea zgomotului suprapus unei imagini.

Filtrarea reprezintă o operație de vecinătate. Prin aceasta se înțelege că la calculul noii valori a unui pixel vor contribui și valorile pixelilor vecini, nu doar vechea lui valoare, cum se întâmplă la operațiile punctuale. Vecinii unui pixel reprezintă o mulțime de pixeli, aflați în apropierea acestuia, care alcătuiesc o *vecinătate*. Această vecinătate poate avea diverse forme și dimensiuni, însă cele mai utilizate în practică sunt vecinătățile de formă pătrată, de dimensiuni impare.

7.1 Filtrarea liniară a imaginilor

După cum îi spune numele, acest tip de filtrare respectă principiul liniarității (sau al superpoziției).

Principiul liniarității: Fiind date două imagini $f_1(x, y)$ și $f_2(x, y)$, și două numere reale α și β , se numește **operator liniar**, un operator O care are următoarea proprietate:

$$O[\alpha \cdot f_1(x, y) + \beta \cdot f_2(x, y)] = \alpha \cdot O[f_1(x, y)] + \beta \cdot O[f_2(x, y)] \quad (7.1)$$

Operația de filtrare liniară calculează noua valoare a unui pixel al imaginii, din poziția (m, n) , ca o combinație liniară a unui număr de valori din imaginea originală, astfel:

$$g(m, n) = \sum_{(k, l) \in W} w_{kl} \cdot f(m - k, n - l) \quad (7.2)$$

unde $f(x, y)$ este imaginea originală (afectată sau nu, de zgomot), $g(x, y)$ este imaginea filtrată, W este o structură de puncte care definește vecinătatea

pixelului (m, n) , w_{kl} sănt niște valori constante care reprezintă **coeficientii filtrului**.

Filtrul este definit de vecinătatea W și de coeficientii w_{kl} . Un filtru poate fi specificat de o matrice V , care poartă numele de **mască de convoluție** sau **mască de filtrare**, care este caracterizată de formă, valorile coeficienților și de origine. În Figura 7.1 este prezentată o mască de filtrare de formă pătrată, de dimensiune 3x3, având originea în centru.

$w_{-1,-1}$	$w_{-1,0}$	$w_{-1,1}$
$w_{0,-1}$	$w_{0,0}$	$w_{0,1}$
$w_{1,-1}$	$w_{1,0}$	$w_{1,1}$

Figura 7.1: Mască de filtrare pătrată de dimensiune 3x3.

Nu este obligatoriu ca forma măștii de filtrare să fie pătrată, de dimensiune impară sau să aibă originea în centrul măștii.

Operația de filtrare liniară poate fi descrisă astfel: se suprapune masca de filtrare peste fiecare pixel al imaginii originale, astfel încât originea măștii să coincidă cu pixelul considerat, apoi se calculează toate produsele între coeficientii măștii și valorile pixelilor peste care se suprapun acești coeficienți, iar suma acestor produse reprezintă noua valoare a pixelului considerat. Această tehnică poartă numele de **tehnica ferestrei glisante**. Operația descrisă reprezintă de fapt o convoluție bidimensională.

7.1.1 Filtrele de netezire

Filtrele de netezire sunt echivalentele bidimensionale ale filtrelor trece-jos (FTJ), și la fel ca acestea, sănt folosite în general pentru eliminarea zgomotului, care se presupune că este de bandă largă.

Informația conținută într-o imagine, în general, se regăsește în componente de joasă frecvență, și deci este justificată o filtrare trece-jos pentru reducerea puterii zgomotului.

Zgomotul din imagine se presupune că este aditiv și pur aleator, adică se consideră următoarele ipoteze simplificatoare:

- $g(i, j) = f(i, j) + n(i, j)$ (zgomotul n este aditiv),
- n este un semnal staționar (comportamentul său statistic nu depinde de coordonatele i și j ale pixelului),

- $\bar{n} = 0$ (media zgomotului este zero),
- dacă zgomotul are dispersia σ_n , atunci:

$$\overline{n(i_1, j_1) \cdot n(i_2, j_2)} = \begin{cases} \sigma_n^2 & \text{pentru } i_1 = i_2 \& j_1 = j_2 \\ 0 & \text{în rest} \end{cases}$$

(zgomotul este complet decorrelat).

Condiția de normare a coeficienților filtrelor de netezire

Pentru un filtru trece-jos se impune următoarea condiție: componenta continuă a imaginii să nu fie alterată de filtru. Cu alte cuvinte, filtrul să conserve luminozitatea medie a imaginii.

Pentru aceasta, considerăm o imagine având un singur nivel de gri, constant, notat cu μ , adică: $f(i, j) = \mu$ pentru oricare i și j . Pentru ca filtrul să conserve luminozitatea medie, adică valoarea medie μ , impunem $g(i, j) = \mu$ pentru $\forall i, j$. Rezultă:

$$\mu = \sum_{(k,l) \in W} \sum w_{kl} \cdot \mu \quad (7.3)$$

$$\sum_{(k,l) \in W} w_{kl} = 1 \quad (7.4)$$

unde $w_{kl} \geq 0$.

Relația (7.4) poartă numele de condiție de normare pentru filtre de netezire (sau trece-jos).

Filtrul de mediere

Filtrul de mediere este cel mai simplu filtru de netezire. Caracteristic unui filtru de mediere este faptul că toți coeficienții măștii de filtrare sunt egali. Dacă ținem cont și de condiția de normare, atunci coeficienții unui filtru de mediere, care are o mască de filtrare de dimensiune $N \times N$, au valoarea $\frac{1}{N^2}$. În Figurile 7.2, 7.3 și 7.4 sunt prezentate măștile de filtrare de mediere, pentru $N = 3, 5$ și respectiv 7.

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Figura 7.2: Mască de filtrare pătrată de dimensiune 3×3 .

$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$
$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$
$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$
$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$
$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$

Figura 7.3: Mască de filtrare pătrată de dimensiune 5×5 .

$\frac{1}{49}$						
$\frac{1}{49}$						
$\frac{1}{49}$						
$\frac{1}{49}$						
$\frac{1}{49}$						
$\frac{1}{49}$						
$\frac{1}{49}$						

Figura 7.4: Mască de filtrare pătrată de dimensiune 7×7 .Figura 7.5: Filtrarea de mediere: (a) imaginea originală; (b) imaginea filtrată cu o mască 7×7 .

Filtrul de mediere nu este folosit în practică, deoarece, pe lângă zgomot, are de suferit și semnalul util (vezi Figura 7.5). Filtrarea de mediere este deranjantă pentru imagine în acele zone în care imaginea conține frecvențe înalte (variații brusă), pentru că duce la apariția fenomenului de încetășare¹.

¹(engl.) blurring.

Din punctul de vedere al zgomotului, pentru filtrare este utilă o mască de filtrare de dimensiune cât mai mare. Din punctul de vedere al semnalului util, al imaginii, este util ca masca să fie cât mai mică. În practică se realizează un compromis între cele două aspecte.

Alte măști de filtrare

Alte măști de filtrare, pentru filterele de mediere, sunt prezentate în continuare, deși toate au același randament nesatisfăcător.

$$\left(\begin{array}{ccc} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{array} \right) \quad \left(\begin{array}{ccc} \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\ \frac{1}{16} & \frac{1}{2} & \frac{1}{16} \\ \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \end{array} \right) \quad \left(\begin{array}{ccc} 0 & \frac{1}{5} & 0 \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ 0 & \frac{1}{5} & 0 \end{array} \right) \quad \left(\begin{array}{ccc} 0 & \frac{1}{8} & 0 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ 0 & \frac{1}{8} & 0 \end{array} \right)$$

7.1.2 Filtrele trece-sus

Filtrele trece-sus urmăresc eliminarea componentelor de frecvență joasă din imagine. Sunt folosite în general pentru detectarea frontierelor sau conturilor din imagine, acolo unde au loc treceri sau variații bruse ale luminanței.

Condiția de normare a coeficienților filtrelor trece-sus

Condiția de normare a coeficienților pentru un filtru trece-jos (relația (7.6)) se determină impunând condiția ca filtrul să rejecteze complet (sau să atenueze complet) componenta continuă a imaginii.

Pentru aceasta vom considera, la fel, o imagine $f(i, j) = \mu$ pentru $\forall i, j$. La ieșirea filtrului trece-sus vom avea $g(i, j) = 0$ pentru $\forall i, j$.

$$0 = \sum_{(k,l) \in W} w_{kl} \cdot \mu \quad (7.5)$$

$$\sum_{(k,l) \in W} w_{kl} = 0 \quad (7.6)$$

Filtrul de accentuare

Filtrul de accentuare nu este un filtru trece-sus, ci folosește filtrarea trece-sus pentru a realiza accentuarea. Prin accentuare se înțelege contrastarea unei imagini și are ca scop îmbunătățirea percepției vizuale a contururilor obiectelor. Cu alte cuvinte, îmbunătățirea detectabilității componentelor scenei de-a lungul frontierelor acestora. Acest lucru se realizează, în principiu, prin modificarea valorilor pixelilor situați de o parte și de alta a unei frontiere comune.

Sistemul uman are tendința de a adânci profilul zonelor de tranziție dintre regiuni uniforme. Studiul fiziologiei sistemului vizual a demonstrat că aceasta se realizează prin prelucrări de tip derivativ ce apar în diferitele etape pe care le parcurge informația vizuală. Efectul global poate fi modelat matematic prin scăderea din semnalul original a unei derive secunde ponderate.

În continuare sunt prezentate câteva măști de implementare a unei derive secunde de tip Laplace:

$$\begin{pmatrix} 0 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 1 & -\frac{1}{4} \\ 0 & -\frac{1}{4} & 0 \end{pmatrix} \quad \begin{pmatrix} \frac{1}{4} & -\frac{1}{2} & \frac{1}{4} \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ \frac{1}{4} & -\frac{1}{2} & \frac{1}{4} \end{pmatrix} \quad \begin{pmatrix} -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \\ -\frac{1}{8} & 1 & -\frac{1}{8} \\ -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \end{pmatrix}$$

În Figura 7.6 puteți observa efectele filtrării Laplace.

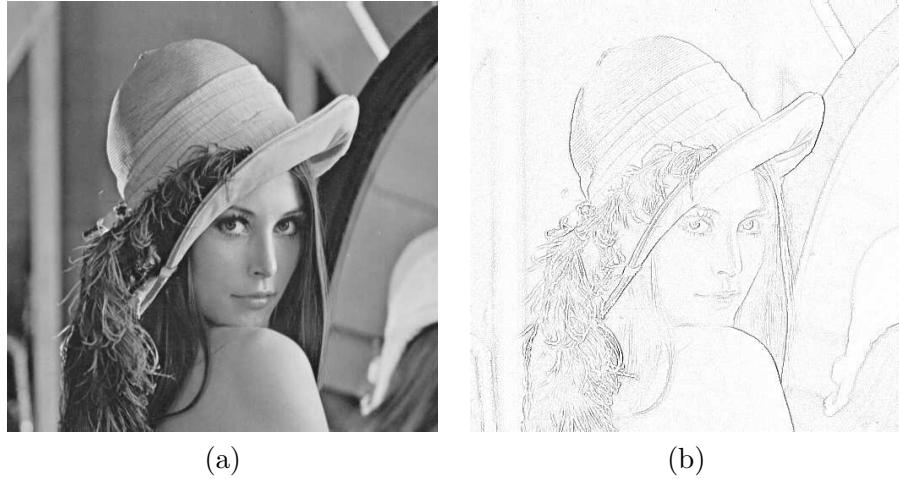


Figura 7.6: Filtrarea Laplace: (a) imaginea originală; (b) imaginea rezultată (negativată).

Filtrul de accentuare se implementează după schema prezentată în Figura 7.7.

7.2 Filtrarea neliniară a imaginilor

Filtrele neliniare nu respectă principiul liniarității sau al superpoziției enunțat la începutul lucrării. Acestea au apărut din necesitatea de a depăși limitările filtrelor liniare, în ceea ce privește zgomotele care nu au o distribuție normală sau nu sunt aditive.

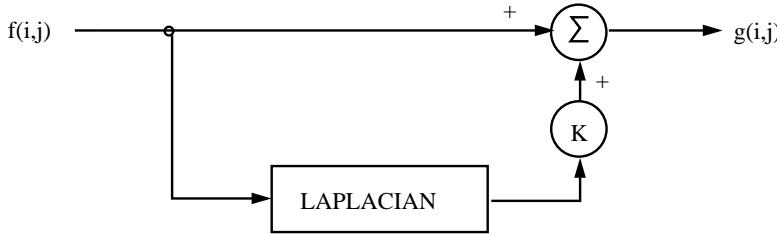


Figura 7.7: Filtrul de accentuare.

7.2.1 Filtrele de ordine

Filtrele de ordine sunt operatori locali, definiți la rândul lor de o fereastră, care selectează din imagine un număr de pixeli vecini pixelului curent, într-un mod identic cu tehnica ferestrei glisante. Valorile pixelilor selectați se ordonează crescător.

Să presupunem că fereastra conține n pixeli, iar valorile lor formează următoarea mulțime:

$$\{x_1, x_2, \dots, x_n\} \quad (7.7)$$

După ce aceste valori au fost ordonate crescător, vom avea:

$$\{x_{(1)}, x_{(2)}, \dots, x_{(n)}\} \quad (7.8)$$

pentru care sunt îndeplinite următoarele condiții:

$$x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)} \quad (7.9)$$

Ieșirea filtrului de ordin k , pentru $k \in [1; n]$ întreg, este statistica de ordinul k , cu alte cuvinte, elementul de pe poziția k din sirul ordonat:

$$rank_k\{x_1, x_2, \dots, x_n\} = x_{(k)} \quad (7.10)$$

Filtrul median

Filtrul median este un filtru de ordin a cărui ieșire este statistica de ordin central, adică elementul ce se află pe poziția de mijloc a sirului ordonat de valori selectate de fereastra de filtrare:

$$median\{x_1, x_2, \dots, x_n\} = \begin{cases} x_{(\frac{n+1}{2})} & \text{dacă } n \text{ este impar,} \\ \frac{1}{2}x_{(\frac{n}{2})} + \frac{1}{2}x_{(\frac{n}{2}+1)} & \text{dacă } n \text{ este par.} \end{cases} \quad (7.11)$$

Filtrul median este potrivit pentru eliminarea zgomotului de tip “sare și piper”. După ordonarea valorilor pixelilor, valorile zgomotului (adică 0 sau 255) se vor situa pe primele, respectiv ultimele poziții în mulțime, și deci, la ieșirea filtrului, vom avea o valoare diferită de valorile zgomotului. Totuși există și situații în care, după filtrare, mai există pixeli afectați de zgomot. În acest caz, spunem că filtrul a fost “străpuns” de zgomot. Acest lucru este posibil atunci când mai mult de jumătate din pixelii selectați de fereastra de filtrare, sunt afectați în același mod (sare sau piper, 255 sau 0) de zgomot.

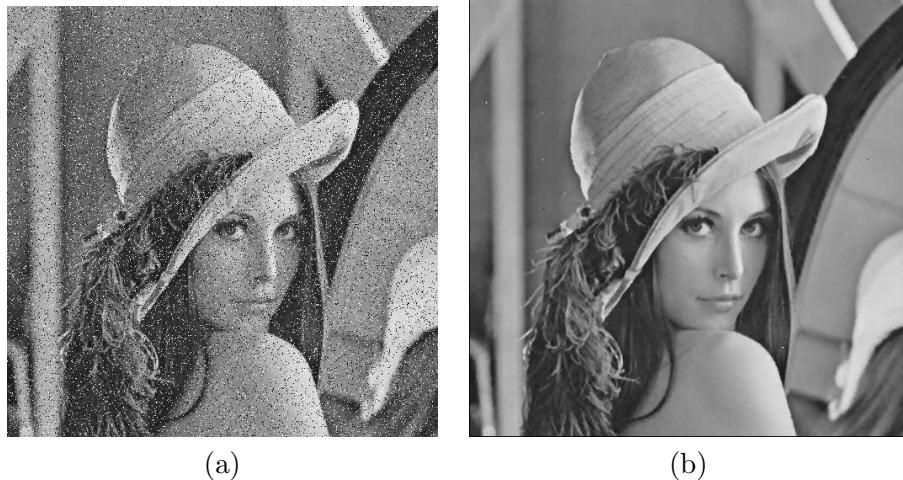


Figura 7.8: Filtrul median: (a) imaginea originală afectată de zgomot “sare și piper”; (b) imaginea filtrată.

Filtrul de minim

Filtrul de minim este un filtru de ordine a cărui ieșire este statistica de ordinul 1, adică valoarea $x_{(1)}$, care este cea mai mică valoare din mulțimea de valori ale pixelilor selectați de către fereastra de filtrare.

Filtrul de maxim

Filtrul de maxim este un filtru de ordine a cărui ieșire este statistica de ordinul n , adică valoarea $x_{(n)}$, care este cea mai mare valoare din mulțimea de valori luate în considerare.

DESFĂȘURAREA LUCRĂRII

Problema 1. Observați efectele filtrului de mediere pentru o imagine afectată de zgomot gaussian. Codul C al funcției care implementează filtrul de

mediere este prezentat în continuare:

```

void ImageViewer :: filtru_mediere( void )
{
    int i, j;
    int k, l;
    int w, h;

    double v[ 3 ][ 3 ];

    //coeficientii mastii de filtrare
    v[0][0] = 1./9;  v[0][1] = 1./9;  v[0][2] = 1./9;
    v[1][0] = 1./9;  v[1][1] = 1./9;  v[1][2] = 1./9;
    v[2][0] = 1./9;  v[2][1] = 1./9;  v[2][2] = 1./9;

    w = image.width();
    h = image.height();

    QImage image_fil( w, h, 32, 0, QImage::IgnoreEndian );

    for( i = 1; i < w - 1; i++ )
        for( j = 1; j < h - 1; j++ )
    {
        //suma ponderata
        double sum = 0;

        for( k = -1; k < 2; k++ )
            for( l = -1; l < 2; l++ )
                sum += v[ k + 1 ][ l + 1 ] *
qRed( image.pixel( i + k, j + l ));

        image_fil.setPixel( i, j,
                           qRgb( (int)sum, (int)sum, (int)sum ));
    }

    image = image_fil;
    pm = image;
    update();
}

```

Problema 2. Observați efectele filtrului de mediere pentru o imagine afectată de zgomot de tip “sare și piper”.

Problema 3. Observați efectul de “blurring” al filtrului de mediere pentru o imagine neafectată de zgomot.

Problema 4. Implementați un filtru de mediere cu o mască de filtrare de formă pătrată de dimensiune 5x5.

Problema 5. Observați efectul filtrului de accentuare (pentru o imagine neafactată de zgomot). Codul C al filtrului este următorul:

```

void ImageViewer :: filtru_accentuare( void )
{
    int i, j;
    int k, l;
    int w, h;

    double sum;
    double v[ 3 ][ 3 ];

    //coeficientii mastii
    v[0][0] = 0;      v[0][1] = -1./4; v[0][2] = 0;
    v[1][0] = -1./4; v[1][1] = 1;     v[1][2] = -1./4;
    v[2][0] = 0;      v[2][1] = -1./4; v[2][2] = 0;

    w = image.width();
    h = image.height();

    QImage image_fil( w, h, 32, 0, QImage::IgnoreEndian );

    for( i = 1; i < w - 1; i++ )
        for( j = 1; j < h - 1; j++ )
    {
        sum = 0;

        for( k = -1; k < 2; k++ )
            for( l = -1; l < 2; l++ )
                sum += 1. * v[ k + 1 ][ l + 1 ] *
                    qRed( image.pixel( i + k, j + l ) );

        int niv = qRed( image.pixel( i, j ) );
        niv = (int)( niv + 0.6 * sum );
        image_fil.setPixel( i, j, qRgb( niv, niv, niv ) );
    }

    image = image_fil;
    pm = image;
    update();
}

```

Problema 6. Observați efectele filtrului median pentru o imagine afecată de zgomot de tip “sare și piper”. Citiți și înțelegeți implementarea în C:

```

void ImageViewer :: filtru_median( void )
{
    int i, j;
    int w, h;
    int k, aux;
    int m, n;
    int med;
    int sir[ 9 ];

    w = image.width();
    h = image.height();

    QImage image_fil( w, h, 32, 0, QImage::IgnoreEndian );

    for( i = 1; i < w-1; i++ )
        for( j = 1; j < h-1; j++ )
    {
        //formarea unui sir din elementele vecinatatii 3x3
        k = 0;
        for( m = -1; m < 2; m++ )
            for( n = -1; n < 2; n++ )
            {
                sir[k] = qRed( image.pixel( i+m, j+n ) );
                k++;
            }
        //ordonarea crescatoare a valorilor pixelilor
        //metoda BUBBLE SORT
        k = 0;
        while( k == 0 )
        {
            k = 1;
            for( m = 0; m < 8; m++ )
                if( sir[ m ] > sir[ m + 1 ] )
                {
                    aux = sir[ m ];
                    sir[ m ] = sir[ m + 1 ];
                    sir[ m + 1 ] = aux;
                    k = 0;
                }
        }
    }
}
```

```
}

//elementul median
med = sir[ 4 ];

//noua valoare a pixelului
image_fil.setPixel( i, j, qRgb( med, med, med ) );
}

image = image_fil;
pm = image;
update();
}
```

Problema 7. Implementați filtrul de minim. Observați efectele lui asupra unei imagini neafectate de zgomot.

Problema 8. Implementați filtrul de maxim. Observați efectele lui asupra unei imagini neafectate de zgomot.