

Laborator 6 TPI 2022-2023

Aplicații ale transformatei wavelet în filtrarea imaginilor

1. Introducere. Obiectul lucrării

Tehnicile de filtrare a imaginilor bazate pe transformate wavelet separă pentru fiecare subbanda de înaltă frecvență - $I(x, y)$ - componentele datorate semnalului util $I_s(x, y)$ de cele datorate zgomotului $I_{zg}(x, y)$:

$$I(x, y) = I(\lambda) = \{I_s(x, y)\} \cup \{I_{zg}(x, y)\} \quad (1)$$

Conform formalismului wavelet, toți coeficienții de detaliu asociați unei descompuneri de tip DWT ce au valori superioare (în valoare absolută) unui prag λ sunt considerați drept semnal util - $I_s(x, y)$ - iar toți ceilalți coeficienți de detaliu sunt considerați ca provenind din zgomot $I_{zg}(x, y)$.

Tehnicile de filtrare a imaginilor prin DWT de tip *hard* presupun anularea tuturor coeficienților datorati zgomotului și păstrarea nemodificată a tuturor coeficienților proveniți din semnalul util.

Metodele consacrate de filtrare în domeniul wavelet presupun efectuarea unor operații premergătoare comparării propriu zise:

- estimarea nivelului de zgomot;
- estimarea pragului de separație între cele două clase de coeficienți λ .

Cele mai folosite tehnici, *VisuShrink* și *BayesShrink* pornesc de la banda de frecvențe $HH...H$ și estimează nivelul de zgomot prin deviația standard mediană (Median Absolute Deviation):

$$MAD = \frac{\text{mediana}(|I_{HH}|)}{0.6745} \quad (2)$$

Pragul de separație în modelul *VisuShrink* este estimat pe baza relației

$$\lambda = MAD \cdot \sqrt{21g N_{SB}} \quad (3)$$

cu N_{SB} este numărul total al coeficienților de detaliu din subbanda considerată.

Chiar dacă pragul este estimat doar pornind de la această bandă de frecvențe, filtrarea wavelet de tip *VisuShrink* presupune utilizarea acestuia în comparații în toate subbenzile de frecvențe – mai puțin $LLL...L$. Acest prag este recalculat pentru fiecare nivel de descompunere la care se aplică operația de filtrare.

Obiectul lucrării de laborator constă în studiul și implementarea unei metode de filtrare de *VisuShrink* și comparația acesteia cu alte tipuri de metode clasice de filtrare a imaginilor.

2. Implementarea algoritmului VisuShrink în VisualDSP ++

Scheletul aplicației ce va fi folosită pentru implementarea algoritmului VisuShrink corespunde unei implementări a DWT prin bancuri de filtre de tip Haar. Proiectul Visual DSP++ disponibil pe pagina de web a disciplinei include toate funcțiile necesare:

*float Estimare_prag(float * intrare, float * hh, int latime, int inaltime)* - funcție ce permite determinarea pragului de comparație conform formulelor (2) și (3)

*void sortare(float * valori, long numar_el)* – funcție ce permite sortarea în ordine descrescătoare a unui șir de valori de o anumită lungime – ambele transmise ca parametri în apelul funcției

*void comparare_prag(double * intrare, int lim_jos_x, int lim_sus_x, int lim_jos_y, int lim_sus_y, double prag)* – funcție ce permite compararea valorilor aferente unei regiuni dintr-o imagine cu un prag. Regiunea ce va fi analizată este specificată prin limitele pe x și y denumite sugestiv lim_jos(sus)_x(y).

3. Exerciții

- 1) Analizați modul de implementare al funcțiilor de mai sus (anexa 1).
- 2) Modificați proiectul *Filtrare_wavelet* astfel încât să permită filtrarea VisuShrink a unei imagini.
- 3) Folosind rezultatele de la punctul 2, procesați imaginea **lena_zg.bmp**; analizați rezultatele obținute și indicați care este efectul filtrării wavelet asupra coeficienților de detaliu . Salvați rezultatul obținut în directorul curent sub numele **lena_wavelet.bmp**. Care sunt valorile de prag date de ecuațiile 2 și 3?
- 4) Repetați punctul anterior pentru filtrarea imaginii **lena_wavelet.bmp**
- 5) Modificați condiția de comparare *if (fabs(intrare [i+j*dimx])<prag)* cu *if (fabs(intrare [i+j*dimx])>0)*
Rulați aplicația și explicați efectul noii condiții analizând efectele asupra imaginii obținute.
- 6). Propuneți implementări în cadrul unui alt proiect VisualDSP a unui filtru de medie și median (N=25). Procesați aceeași imagine și comparați rezultatele cu cele obținute la punctul 3.
- 7) Schițați modificările ce trebuie aduse proiectului *Filtrare_wavelet* astfel încât să permită aplicarea algoritmului pe mai multe nivele.

Bibliografie

[1] R. Terebes – Tehnologii de prelucrare a informației – notițe de curs 2022-2023, <http://ares.utcluj.ro/tpi>

Anexa 1. Funcții folosite în implementarea algoritmului VisuShrink în variantă Hard

```
float Estimare_prag(float* intrare, float * hh_abs, int latime, int inaltime)
{
    float val_ret;
    long i,j;

    for (i=latime/2;i<latime;i++)
        for (j=inaltime/2;j<inaltime;j++)
        {
            if (intrare[i+j*latime]<0)
                hh_abs[i-latime/2+(j-inaltime/2)*latime/2]=-intrare[i+j*latime];
            else
                hh_abs[i-latime/2+(j-inaltime/2)*latime/2]=intrare[i+j*latime];
        }

    sortare(hh_abs, latime*inaltime/4-1);

    val_ret=hh_abs[latime*inaltime/8-1]/0.6745;
    val_ret=val_ret*sqrt(2*log10(inaltime*latime/4));

return val_ret;
}

void sortare(float * valori, int numar_el)
{
    int i,j;
    double curent;
    for (i=0;i<numar_el;i++)
    {
        curent=valori[i];
        j=i-1;
        while (j>=0 && curent>valori[j])
        {
            valori[j+1]= valori[j];
            j=j-1;
        }
        valori[j+1]=curent;
    }
}

void comparare_prag(float * intrare, int lim_jos_x ,int lim_sus_x,int lim_jos_y,int lim_sus_y ,
int dimx, int dimy, float prag)
{
    int i,j;
    for (i=lim_jos_x;i<lim_sus_x;i++)
        for (j=lim_jos_y;j<lim_sus_y;j++)
            if (fabs(intrare [i+j*dimx])<prag)
                intrare[i+j*dimx]=0.0;
}
```