PROIECT DE DIPLOMĂ

PRELUCRAREA IMAGINILOR CU AJUTORUL TRANSFORMĂRII WAVELET

STUDENT: PAUL IROFTI

PROFESOR COORDONATOR: BOGDAN DUMITRESCU

Iunie 2008

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE,
UNIVERSITATEA POLITEHNICĂ BUCUREŞTI

Cuprins

Listă de figuri

Listă de tabele

Rezumat

Lucrarea prezintă prelucrarea imaginilor cu ajutorul transformării wavelet. Se propune pentru aceasta compararea mai multor metode de analiză şi prelucrare ce implică transformata wavelet. Într-o prima instanţă se va prezenta noţiunea de wavelet şi modul de reprezentare a funcţiilor cu ajutorul ei.

Transformata wavelet continuă a unei funcţii se discretizează şi astfel se obţine transformata wavelet discretă. Deficienţele şi problemele de proiectare ale acestei transformări duc la dezvoltarea de noi metode de prelucrare.

Pentru a îmbunătăţi performanţele şi a îndepărta problemele transformatei wavelet discrete se construieşte un arbore dual. Acesta este constituit din doua transformate wavelet discrete ce lucrează în paralel. Ramurile arborelui sunt interpretate drept partea reală şi, respectiv, imaginară a unui wavelet complex. Astfel se obţine transformata wavelet complexă.

O structură bazată pe arborele dual este arborele dual cu densitate dublă. Această abordare a prelucrării imaginii îmbină transformata wavelet complexă cu transformata cu densitate dublă. Cea din urmă este bazată tot pe două transformate wavelet discrete dar prezintă anumite proprietăţi ce sunt dorite şi în arborele dual.

Pentru aceste abordări ale prelucrării imaginii cu ajutorul transformatei wavelet, se implementează rutinele Matlab necesare pentru testarea şi experimentarea diferitelor caracteristici şi performanţe ale acestor structuri.

Cu ajutorul mai multor bancuri de filtre, specifice fiecărei structuri, se testează performanţele în eliminarea zgomotului pentru fiecare implementare şi se compara atât numeric cât şi grafic.

1 Introducere

Nevoia de a prelucra şi îmbunătăţi un semnal este prezentă în numeroase procese şi aplicaţii. De exemplu, de fiecare dată când un semnal este transmis de-a lungul unui canal, există riscul de a pierde informaţii. Aceste pierderi sunt cunoscute şi prevăzute în proiectarea canalului, astfel apărând algoritmi de corecţie şi algoritmi de verificare a integrităţii datelor. Un alt exemplu este nevoia de a comprima semnalul pentru a-l putea reprezenta pe un suport cât mai scurt fie pentru a putea fi transmis mai rapid, fie pentru a fi stocat într-o arhivă sau chiar pentru a menţine numai informaţiile esenţiale oferite de acesta.

În aceste cazuri, şi multe altele, pentru a obţine informaţia originală a semnalului, sau o copie cât mai fidelă a lui, sunt necesari diferiţi algoritmi de sinteza (reconstrucţie). În unele cazuri, semnalul este în aşa fel modificat încât reconstrucţia lui să fie robustă rezultând un semnal prelucrat identic cu cel iniţial, precum anumiţi algoritmi de compresie. Există şi cazuri în care semnalul original, odată modificat, este pierdut, iar algoritmii de sinteză sunt proiectaţi astfel încât să reproducă o copie cât mai fidelă. Aici se încadrează şi unele metodele de compresie de imagini sau sunete prin eliminarea anumitor porţiuni ce nu sunt sesizabile senzorilor de percepţie umani (JPG, MP3, OGG).

Cazul cel mai defavorabil pentru reconstrucţia unui semnal este când nu se cunoaşte sursa de unde a derivat acesta. Fără a şti cum era reprezentat semnalul original sau ce procese l-au modificat pe parcurs crearea algoritmilor specifici cu rezultate bune este destul de dificilă. Astfel se apelează la o reconstrucţie generică ce ia în calcul doar semnalul perturbat, încercându-se eliminarea pe cât mai mult posibil a zgomotelor apărute pe parcurs.

Daca reprezentăm o imagine drept o matrice bidimensionala, în care poziţia fiecărui element şi valoarea sa sunt corespondente poziţiei şi, respectiv, culorii unui pixel din imagine, atunci acestea pot fi considerate şi reprezentate drept semnale bidimensionale. Aceasta permite prelucrarea imaginilor cu noţiunile şi teoriile actuale din teoria prelucrării semnalelor.

Prelucrarea imaginii, şi a semnalelor în general, reprezintă un domeniu foarte larg în care diferite metode şi abordări apar frecvent. O metodă recentă este prelucrarea imaginilor cu ajutorul transformării wavelet.

Marele avantaj al wavelet-urilor îl reprezintă legătura dintre frecvenţă şi timp[3]. Datorită acestei proprietăţi, wavelet-urile sunt folosite în prezent în domenii cât mai diverse ducând la un număr mare de lucrări şi teorii ştiinţifice ce au la bază transformata wavelet.

Unul dintre aceste domenii este prelucrarea semnalelor, mai specific analiza, prelucrarea şi sinteza semnalelor cu ajutorul transformării wavelet.

2 Transformarea wavelet

Wavelet-ul este un tip de funcţie folosit pentru a împărţii o anumită funcţie sau semnal în componente diferite de timp-frecvenţă. Acestea pot fi studiate la o rezoluţie corespunzătoare scalei wavelet-ului.

Prin transformata wavelet se înţelege reprezentarea unei funcţii cu ajutorul wavelet-urilor. Wavelet-urile reprezintă copii scalate şi translatate ale unei unde oscilante de lungime finită. Aceste copii sunt cunoscute sub denumirea de wavelt-uri fiindcă, în timp ce undele poartă numele de wavelet-uri mamă.

Spre deosebire de transformata Fourier, transformata wavelet conferă posibilitatea de a reprezenta funcţii ce au discontinuităţi sau/şi vârfuri ascuţite. Un alt avantaj îl constituie capacitatea de a deconstrui (analiza) şi reconstrui (sintetiza) semnale neperiodice şi/sau nestaţionare.

Transformările wavelet sunt clasificate în transformări wavelet discrete (DWT) şi transformări wavelet continue (CWT). În timp ce transformările continue pot acţiona asupra oricărei translaţii sau scalări, transformările discrete folosesc o submulţime specifică de valori pentru aceste operaţii.

În continuare vor fi definite atât cele două tipuri de transformate wavelet cât şi operaţiile şi funcţiile aferente.

2.1 Definiţii

Wavelet-ul este o funcţie $\psi \in \mathbb{L}^2(\mathbb{R})$ ce îndeplineşte următoarele proprietăţi:

- are media nulă: $\int_{-\infty}^{+\infty} \psi(t)\, dt = 0$;
- este normată: $||\psi|| = 1$;
- este centrată în vecinătatea: $t = 0$.

O familie de timp-frecvenţă este obţinută prin scalarea lui ψ cu s şi translatare ei cu u:

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right)$$

Aceste prelucrări păstrează norma:

$$||\psi_{u,s}(t)|| = 1$$

Transformata wavelet a funcţiei $f \in \mathbb{L}^2(\mathbb{R})$ la momentul u şi scala s este:

$$\mathbf{W}f(u,s) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-u}{s} \right) dt$$

Filtrarea liniară

Transformata wavelet poate fi rescrisă sub forma unui produs de convoluţie:

$$Wf(u,s) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-u}{s} \right) dt \text{ cu:}$$

$$\bar{\psi}_s(t) = \frac{1}{\sqrt{s}} \psi^* \left(\frac{-t}{s} \right)$$

Transformata Fourier a lui $\bar{\psi}_s(t)$ este:

$$\widehat{\bar{\psi}_s}(w) = \sqrt{s} \hat{\psi}^*(s\omega)$$

Din moment ce $\hat{\psi}(0) = \int_{-\infty}^{+\infty} \psi(t)\, dt = 0$, se observă că $\hat{\psi}$ reprezintă funcţia de transfer a unui filtru trece bandă. Astfel convoluţia calculează transformata wavelet cu filtre trece bandă dilatate.

2.1.1 Transformarea Wavelet Reală

Fie ψ un wavelet real. Pentru că media sa este nulă, integrala wavelet

$\mathbf{W}f(u,s) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-u}{s} \right) dt$ măsoară variaţia lui f în vecinătatea lui u, a cărei mărime este proporţională cu s.

Transformarea reală wavelet este completă şi conservă energia semnalului, atâta timp cât wavelet-ul satisface o condiţie slabă de admisibilitate specificată de teorema Calderon-Grossmann-Morlet:

Teoremă: *Fie $\psi \in \mathbb{L}^2(\mathbb{R})$ o funcţie reală astfel încât:*

$$C_\psi = \int_0^{+\infty} \frac{\left| \hat{\psi}(\omega) \right|}{\omega} d\omega < +\infty$$

Atunci orice $f \in \mathbb{L}^2(\mathbb{R})$ satisface:

$$f(t) = \frac{1}{C_\psi} \int_0^{+\infty} \int_{-\infty}^{+\infty} Wf(u,s) \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) du \frac{ds}{s^2} \text{ şi}$$
$$\int_{-\infty}^{+\infty} |f(t)|^2 dt = \frac{1}{C_\psi} \int_0^{+\infty} \int_{-\infty}^{+\infty} |Wf(u,s)|^2 du \frac{ds}{s^2}$$

Ipoteza teoremei se numeşte condiţia de admisibilitate wavelet. Pentru a garanta că aceasta integrală este finită trebuie să ne asigurăm că $\hat{\psi}(0) = 0$, ceea ce explică condiţia impusă la început şi anume ca toate wavelet-urile să aibă media nulă. Condiţia este aproape suficientă. Dacă $\hat{\psi}(0) = 0$ şi $\hat{\psi}(\omega)$ este continuă şi derivabilă atunci condiţia de admisibilitate este satisfăcută.

Funcţia de scalare.

Când $Wf(u,s)$ este cunoscută doar pentru $s < s_0$, apare necesitatea unei informaţii suplimentare, pentru recuperarea lui f, corespunzătoare lui $Wf(u,s)$ pentru $s > s_0$. Aceasta informaţie este obţinută introducând o funcţie de scalare ϕ reprezentând o agregare a wavelet-urilor cu scala mai mare decât 1.

Modulul transformatei Fourier a funcţiei de scalare ϕ este:

$$\left|\hat{\phi}(\omega)\right|^2 = \int_1^{+\infty} \left|\hat{\psi}(s\omega)\right|^2 \frac{ds}{s}$$

Faza complexă a lui $\hat{\phi}(()\,\omega)$ poate fi aleasă arbitrar. Se poate verifica faptul că $||\phi|| = 1$ şi se poate evidenţia din condiţia de admisibilitatea faptul că:

$$\lim_{\omega \to 0} \left|\hat{\phi}(\omega)\right|^2 = C_\psi$$

Astfel funcţia de scală poate fi interpretată ca răspunsul la impuls a unui filtru trece-jos.

$$\phi_s(t) = \frac{1}{\sqrt{s}} \phi\left(\frac{t}{s}\right) \; ; \; \bar{\phi}_s(t) = \phi_s^*(-t).$$

Deci aproximarea pentru frecvenţe joase a lui f la scala s este:

$$Lf(u,s) = f \times \bar{\phi}_s(u)$$

2.1.2 Transformata Wavelet Discretă

Fie $\dot{f}(t)$ un semnal continuu uniform eşantionat în intervale de N^{-1} în $[0,1]$. Transformata wavelet a acestui semnal poate fi calculată doar pentru scale $N^{-1} < s < 1$. Pentru procesarea discretizată este mai uşor de normalizat eşantioanele la o distanţă de 1 şi astfel se consideră semnalul dilatat $f(t) = \dot{f}(N^{-1}t)$. Dacă se efectuează schimbarea de variabilă în cadrul transformatei wavelet se obţine:

$$W\dot{f}(u,s) = N^{-1/2} W f(Nu, Ns)$$

Se notează $f[n] = f(n)$ semnalul discret de mărime N. Transformata wavelet discretă asociată semnalului se calculează la scale $s = a^j$, cu $a = 2^{\frac{1}{v}}$, ce ofera v scale intermediare pentru fiecare octavă $[2^j, 2^{j+1})$.

Fie $\psi(t)$ un wavelet cu suport în intervalul $[-K/2, K/2]$. Pentru $2 \leq a^j \geq NK^{-1}$, un wavelet discret scalat cu a^j este definit astfel:

$$\psi_j[n] = \frac{1}{\sqrt{a^j}} \psi\left(\frac{n}{a^j}\right).$$

Acest wavelet discret are Ka^j valori nenule în $[-N/2, N/2]$. Scala a^j trebuie să fie mai mare decât 2, altfel intervalul de eşantionare ar putea fi mai mare decât suportul wavelet-ului.

Scalare discretă

O transformare wavelet calculată până la scala a^J nu este o reprezentare completă a semnalului. Astfel este necesară adăugarea frecvenţelor $Lf[n, a^J]$ joase corespunzătoare scalelor mai mari de a^J.

Un filtru de scalare discret şi periodic este obţinut prin eşantionarea funcţiei de scalare $\phi(t)$ definită anterior.

$$\phi_J[n] = \frac{1}{\sqrt{a^J}} \phi\left(\frac{n}{a^J}\right) \text{ pentru } n \in [-N/2, N/2].$$

2.2 Dezavantaje ale transformării wavelet

In pofida algoritmilor de calcul eficienţi şi a reprezentării compacte, transformarea wavelet prezintă patru deficiente fundamentale.

2.2.1 Oscilaţiile

Ţinând cont de faptul ca wavelet-urile sunt funcţii cu filtre trece-banda, coeficienţii wavelet şi în special coeficienţii funcţiei prelucrate cu ajutorul transformatei wavelet tind să oscileze pozitiv şi negativ în jurul singularităţilor. Aceasta reprezintă o problemă care complică considerabil procesarea cu ajutorul wavelet-urilor, ceea ce face ca extracţia singularităţilor şi, în particular, modelarea semnalului să devină o provocare.

2.2.2 Invarianţa translaţiilor

Trebuie precizat că o translaţie (oricât de mică) a semnalului poate perturba considerabil tiparul de oscilaţie al coeficienţilor wavelet în jurul singularităţilor. De asemenea, trebuie să se ţină cont de faptul că variaţia la translaţie complică şi procesarea domeniului wavelet.

Astfel, algoritmii trebuie să fie proiectaţi pentru a putea face faţă unei game largi de posibile tipare pentru coeficienţii wavelet cauzate de singularităţile translatate.

Pentru a înţelege mai bine oscilaţiile coeficienţilor wavelet şi variaţiile la translaţie, se considera un semnal lin $x\left(t-to\right)$ ca funcţia treaptă

$$u\left(t\right)=\begin{cases}0 & ,t>0\\ 1 & ,t<0\end{cases}$$

analizat de baze wavelet ce au un număr suficient de momente de anihilare. Coeficienţii wavelet sunt alcătuiţi din eşantioane ale răspunsului la treaptă al wavelet-ului:

$d\left(j,n\right)\approx 2^{-3j/2}\Delta\int_{-\infty}^{2^{j}t_{0}-n}\psi\left(t\right)dt$ unde Δ reprezintă înălţimea saltului.

În timp ce $\psi\left(t\right)$ este o funcţie trece bandă care oscilează în jurul lui 0, răspunsul sau treapta $d\left(j,n\right)$ este o funcţie a lui n. Mai mult, factorul 2^{j} în limita superioară ($j\geq 0$) amplifică sensibilitatea lui $d\left(j,n\right)$ la translaţia de timp t_0, ducând la o puternică variaţie la translaţie.

2.2.3 Aliasing

Din distanţarea largă a eşantioanelor coeficienţilor wavelet şi din faptul că aceşti coeficienţi wavelet sunt calculaţi prin operaţii recurente de subeşantionare discretă combinate cu filtre trece sus şi trece jos neideale, rezultă un alias substanţial. DWT-ul invers anulează desigur alias-ul, dar numai în cazul în care coeficienţii wavelet şi de scalare nu sunt schimbaţi. Orice coeficienţi

wavelet (threshold, filtrare si cuantificare) răstoarnă balanţa dintre transformata directă şi transformata inversă, acestea ducând la artefacte în reconstruirea semnalului.

2.2.4 Lipsa orientării

În timp ce sinusoidele Fourier din dimensiunile mai mari corespund undelor înalte orientate, produsul tensor standard pentru construirea wavelet-urilor produce un efect de checkboard care este direcţionat simultan în mai multe direcţii. Aceasta proprietate de lipsa a selecţiei orientării complică modelarea şi procesarea trăsăturilor imaginilor geometrice precum muchiile şi graniţele.

2.3 Implementare

2.3.1 Transformata discretă wavelet 1D

Pentru a putea vizualiza mai bine întregul proces de prelucrare a unui semnal cu ajutorul transformatei discrete wavelet trebuie studiat în detaliu modul de acţionare al setului de filtre atât în partea de analiza a eşantioanelor cât şi în cea de sinteza.

Bancuri de filtre

În cazul transformatei discrete wavelet 1D, filtrul de analiză descompune semnalul $x(n)$ în doua subbenzi reprezentate de semnalele $c(n)$ şi $d(n)$. Semnalul $c(n)$ reprezintă frecvenţele joase (partea grosieră) a semnalului $x(n)$, iar semnalul $d(n)$ reprezintă frecvenţele înalte (partea detaliata) a semnalului $x(n)$.

Bancul de filtre de analiză filtrează în prima faza $x(n)$ folosind un filtru trece jos şi unul trece sus. Notăm filtrul trece jos cu af1 şi filtrul trece sus cu af2. Ieşirea din fiecare filtru este subeşantionată cu 2 pentru a obţine cele doua semnale $c(n)$ şi $d(n)$.

Bancul de filtre de sinteza combină cele două semnale de subbandă ($c(n)$ şi $d(n)$) pentru a obţine un semnal unic $y(n)$. Filtrele de analiză întâi supraeşantionează cu 2 ambele semnale de subbandă. Semnalele sunt apoi filtrate folosind un filtru trece jos şi un filtru trece sus. Notăm filtrul trece jos cu sf1 şi filtrul trece sus cu sf2. Semnalele sunt apoi adunate pentru a obţine semnalul $y(n)$ (vezi figura 1).

Dacă cele patru filtre sunt proiectate astfel încât să garanteze că semnalul de ieşire $y(n)$ să fie egal cu semnalul de intrare $x(n)$, atunci filtrele satisfac condiţia de reconstrucţie perfectă.

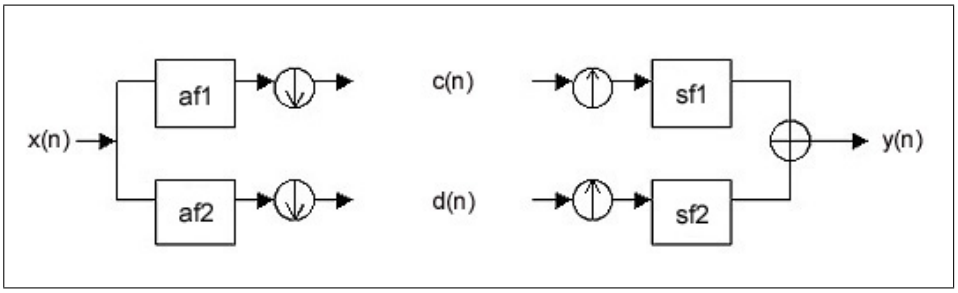

Figura 1: Aplicarea bancurilor de filtre

Transformata wavelet discretă — iterarea bancurilor de filtre

Transformata wavelet discretă produce o reprezentare multiscală a semnalului $x\,(n)$. Transformata wavelet discretă este implementată prin iterarea bancurilor de filtre de analiză prin cele două canale (precum a fost descris mai sus, vezi figura 2).

Mai exact transformata wavelet discretă a semnalului este obţinută prin aplicarea recursivă asupra ieşiri grosiere, asociată frecvenţelor joase, a descompunerii în frecvenţe înalte şi joase.

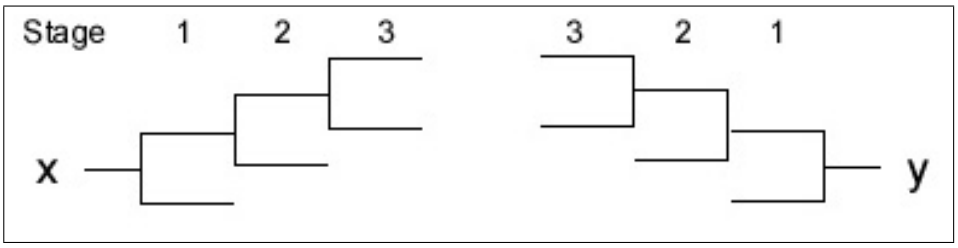

Figura 2: Aplicarea recursivă a bancurilor de filtre

Transformata discretă wavelet a semnalului x este colecţia semnalelor de tip subbandă. Transformata discretă wavelet inversă este obţinută prin aplicarea iterativă a filtrelor de sinteză. Proprietatea de reconstrucţie este garantată de modul în care au fost alese şi proiectate bancurile de filtre.

Wavelet-ul asociat bancurilor de filtre de sinteză poate fi reprezentat grafic folosind un semnal nul. Asupra acestui semnal se aplică transformarea discretă wavelet directă, iar apoi se aplică transformarea inversă. Semnalul obţinut ar trebui sa aibă un grafic similar cu cel din figura 3.

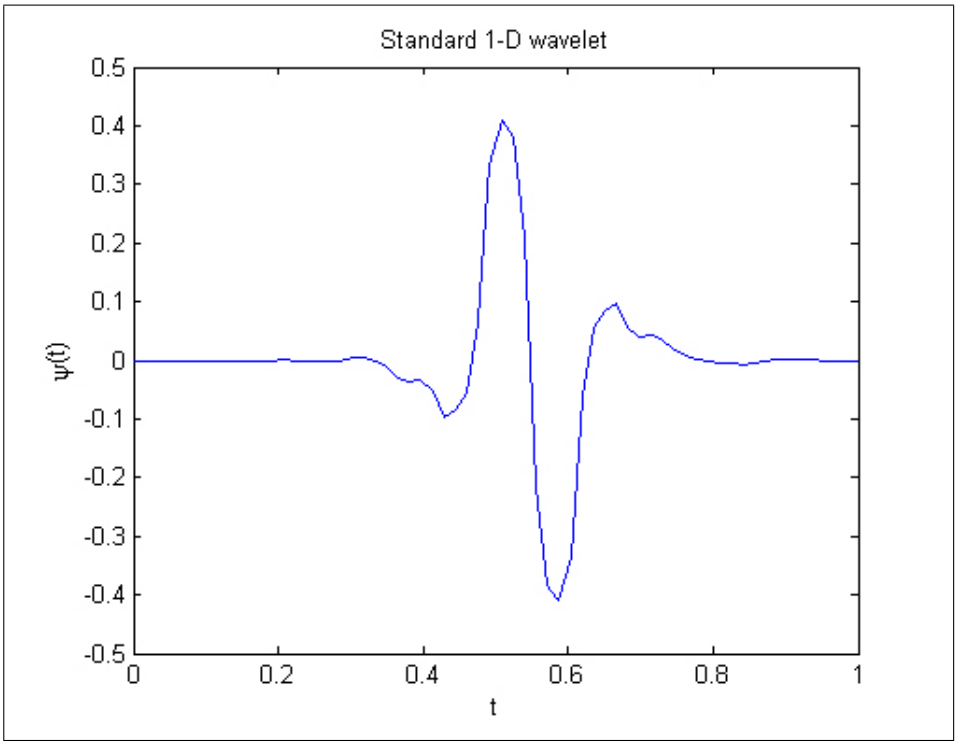

Figura 3: Wavelet 1D

2.3.2 Transformata discretă wavelet 2D

Bancuri de filtre

Pentru a folosi transformata wavelet în procesarea de imagine trebuie implementată o versiune 2D a bancurilor de filtre de analiză şi sinteză. În cazul 2D, filtrele de analiza 1D sunt aplicate întâi pe coloanele imaginii şi apoi linilor.

Dacă imaginea are N1 linii şi N2 coloane, atunci după aplicarea filtrelor 1D de analiză fiecărei coloane se obţin doua subbenzi de imagini, fiecare având N1/2 linii şi N2 coloane. După aplicarea filtrelor de analiza 1D asupra fiecărei linii a ambelor subbenzi, se obţin patru subbenzi conţinând fiecare câte o imagine (vezi figura 4). Fiecare imagine obţinută are N1/2 linii şi N2/2 coloane.

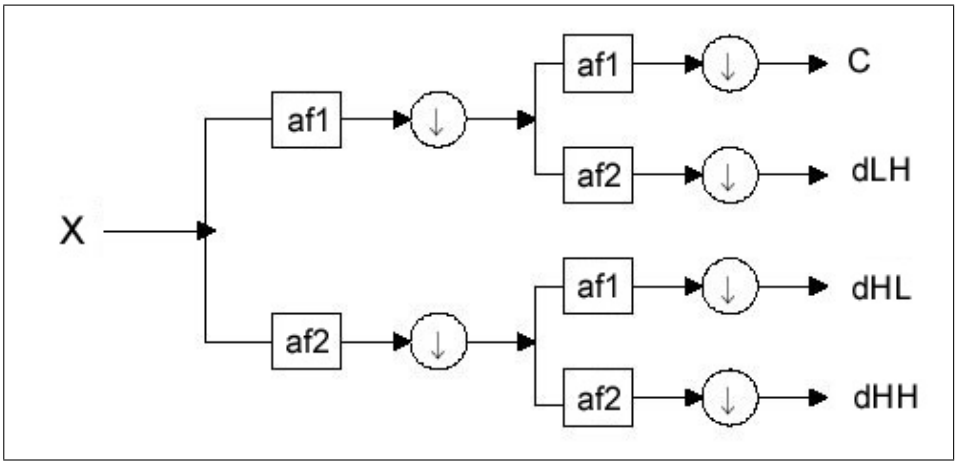

Figura 4: Aplicarea bancurilor de filtre 2D

Notatii:

- **C** – descompunerea frecvenţelor joase, setul de frecvenţe grosiere
- **LH** – descompunerea frecvenţelor joase, secţiunea cu frecvenţe superioare.
- **HL** – descompunerea frecvenţelor înalte, secţiunea cu frecvenţe joase
- **HH** – descompunerea frecvenţelor înalte, secţiunea cu frecvenţe înalte

Bancurile de filtre de sinteza 2D combină cele patru imaginii din subbenzi pentru a obţine imaginea iniţială de dimensiune $N1 \times N2$.

Transformata wavelet discretă – iterarea bancurilor de filtre

Ca şi în cazul 1D, transformata discretă wavelet a semnalului $x(n)$ este implementată prin iterarea bancurilor de filtre de analiză 2D asupra subbenzi cu frecvenţe joase a imaginii. În acest caz la fiecare scală există trei subbenzi în loc de una.

Transformata wavelet discretă inversă se aplica în faza de sinteză, reconstruind imaginea x. Proprietatea de reconstrucţie este din nou satisfăcută datorită alegerii filtrelor în mod corespunzător.

Transformatei wavelet îi sunt asociate trei wavelet-uri în 2D. Acestea sunt reprezentate drept imaginii în nuanţe de gri. Pentru a putea reprezenta grafic aceste wavelet-uri se aplică transformata wavelet directă asupra unui semnal

nul, iar apoi se aplică transformata inversă. Semnalul obţinut la ieşire este similar cu cel din figura 5.

Figura 5: Wavelet 2D

Se observă că primele două wavelet-uri sunt orientate în direcţie verticală şi, respectiv, orizontală. Totuşi, al treilea wavelet nu prezintă o orientare dominantă. Aceasta din urma combină două orientări diagonale indicând efectul de checkboard. Transformata wavelet discretă 2D izolează slab cele două orientări diagonale.

3 Arborele dual

3.1 Definiţii

Arborele dual este construit prin aplicarea simultană a două wavelet-uri discrete asupra unui semnal dat. Având de a face cu două transformate cu coeficienţi reali, cele două vor fi considerate partea reală şi, respectiv, parte imaginară a unei transformate complexe.

3.1.1 Wavelet-uri complexe

Având în vedere deficienţele transformatei wavelet discrete, prezentate în secţiunea anterioară, pentru a putea analiza, corecta şi sintetiza la loc un semnal dat, este nevoie de acumularea unor proprietăţi ce îmbunătăţesc procesarea şi elimină defectele precizate.

Într-o primă instanţă se poate observa că transformata Fourier nu este afectată de aceste probleme. În principal se pot sublinia următoarele proprietăţi ale acestei transformate:

- amplitudinea transformatei Fourier nu oscilează pozitiv şi negativ.
- transformata Fourier este perfect invariabilă la translatări, translatarea fiind asociată cu un simplu deplasament în fază.
- coeficienţii Fourier nu sunt alias şi nu se bazează pe proprietăţi complicate de anulare a efectului de alias pentru reconstrucţie.
- sinusoidele baze Fourier reprezintă planuri bine orientate.

Diferenţa între cele două transformări este că transformata wavelet discretă se bazează pe wavelet-uri reale oscilante, în timp ce transformata Fourier se bazează pe sinusoide oscilante cu valori complexe:

$$e^{j\omega t} = \cos(\omega t) + j \sin(\omega t)$$

Definiţie: Transformata Hilbert:

$$H(u)(t) = -\frac{1}{\pi} \lim_{\epsilon \downarrow 0} \int_{\epsilon}^{\infty} \frac{u(t+\tau) - u(t-\tau)}{\tau} d\tau$$

când transformata este aplicată de două ori succesiv asupra unei funcţii u rezultatul este -u:

$$H\left(H\left(u\right)\right)\left(t\right) = -u\left(t\right)$$

Componentele sinus şi cosinus (partea imaginară şi, respectiv, reală) formează o transformata Hilbert, ceea ce presupune un defazaj de 90° al uneia faţă de cealaltă. Împreună ele constituie un semnal analitic $e^{j\omega t}$ care are suport numai pe jumătate din axa frecvenţei ($\omega > 0$).

Astfel, pornind de la transformarea Fourier, se defineşte transformarea wavelet complexă odată cu funcţia de scalare complexă şi wavelet-urile complexe:

$$\psi_c\left(t\right) = \psi_r\left(t\right) + j\psi_i\left(t\right)$$

Prin analogie cu transformata Fourier, definim $\psi_r\left(t\right)$ fiind reală şi pară şi $j\psi_i\left(t\right)$ drept imaginară şi impară. Mai departe, dacă $\psi_r\left(t\right)$ şi $\psi_i\left(t\right)$ alcătuiesc o transformare Hilbert (defazate cu 90° una faţă de alta), atunci $\psi_c\left(t\right)$ reprezintă un semnal analitic cu suport numai pe jumătate din axa frecvenţei.

Funcţia de scalare complexă este definită similar. Dacă se proiectează semnalul astfel încât să se evidenţieze coeficienţii complecşi wavelet (ex. $2^{j/2}\psi_c\left(2^j t - n\right)$) se obţine:

$d_c\left(j,n\right) = d_r\left(j,n\right) + jd_i\left(j,n\right)$ cu amplitudinea:

$\left|d_c\left(j,n\right)\right| = \sqrt{\left|d_r\left(j,n\right)\right|^2 + \left|d_i\left(j,n\right)\right|^2}$ şi faza:

$$\angle d_c\left(j,n\right) = arctan\left(\frac{d_i\left(j,n\right)}{d_r\left(j,n\right)}\right)$$

când $\left|d_c\left(j,n\right) > 0\right|$. Similar cu transformata Fourier, wavelet-urile complexe pot fi folosite pentru a analiza şi reprezenta atât semnale cu valori reale (rezultând simetria coeficienţilor) cât şi semnale cu valori complexe. În oricare dintre cazuri, transformata wavelet complexă face loc unor noi algoritmi pentru procesarea semnalelor multiscală ce se folosesc de amplitudinea şi faza complexă rezultate.

De exemplu, o amplitudine mare indică prezenţa unei singularităţi, în timp ce faza indică poziţia sa în cadrul suportului wavelet-ului.

3.1.2 Analiticitatea

Proiectarea wavelet-urilor complexe analitice duce la apariţia câtorva probleme unice şi non-triviale ce nu apar la transformata wavelet discretă. Tratarea acestor probleme reprezintă în general un compromis numeric ce duce mai degrabă la o aproximare a wavelet-ului analitic.

Analiticitatea versus suportul finit

Este deseori de dorit, în procesarea semnalelor cu ajutorul wavelet-urilor, ca wavelet-ul să fie bine localizat în timp. În multe aplicaţii wavelet-ul $\psi(t)$ va fi folosit cu suport finit. Astfel wavelet-urile cu suport finit prezintă un interes sporit datorită faptului că transformata wavelet discretă, în acest caz, poate fi implementată uşor folosind filtre cu răspuns finit (FIR).

Cu toate acestea o funcţie cu suport finit nu poate fi niciodată perfect analitică pentru că transformata Fourier a unei funcţii cu suport finit nu poate fi niciodată exact zero pe un interval $[A, B]$ cu $B > A$, nici pe întreaga axă pozitivă sau negativă a frecvenţei. Deci orice wavelet analitic trebuie sa aibă suport infinit.

Astfel dacă se doreşte suport finit pentru wavelet-urile complexe apar următoarele constrângeri:

- trebuie folosite wavelet-uri care sunt aproximativ analitice
- transformata wavelet complexă trebuie sa fie aproximativ invariantă la translaţii ale fazei sau amplitudinii
- transformata wavelet trebuie să prezinte cât mai puţine aliasuri

Analiticitatea versus reconstrucţia perfectă

Reconstrucţia perfectă a semnalului procesat împreuna cu păstrarea proprietăţii de analiticitate a wavelet-ului reprezintă o problemă dificilă.

Problema proiectării filtrelor astfel încât condiţiile de reconstrucţie perfectă să fie satisfăcute şi wavelet-ul să aibă suport mic a fost rezolvată de Daubechies. Cu toate acestea, wavelet-urile Daubechies nu sunt analitice.

O altă abordare a fost împărţirea fiecărei ieşiri a bancurilor de filtre în componente de frecvenţă negativă şi pozitivă, folosind bancuri de filtre cu proprietatea de reconstrucţie drept transformatoare Hilbert. Deşi ieşirile fiecărui canal erau aproape analitice apăreau anumite 'umflături' pe axa frecvenţelor ce nu puteau fi înlăturate complet, ducând astfel la un suport wavelet mult mai mare.

Aplicarea transformatei Hilbert *a priori* O soluţie ce a dus la dezvoltarea arborelui dual o reprezintă aplicarea întâi a transformatei Hilbert asupra datelor. Transformata wavelet reală este aplicata atât datelor originale cât şi celor alterate prin transformata Hilbert, iar coeficienţii fiecărei transformări wavelet sunt combinaţii pentru a obţine transformata wavelet complexă.

Totuşi transformata ideală Hilbert este reprezentată de un impuls infinit de lung. Folosirea transformatei Hilbert aproximativ ideale împreună cu transformata wavelet creşte suportul wavelet-ului. Pentru ca wavelet-urile sa aibă un suport mic, ar trebui folosită o transformată aproximativă Hilbert localizată mai bine în timp.

Acurateţea transformării Hilbert aproximative depinde de scala transformării wavelet (scalele grosiere ar trebui acompaniate de transformate Hilbert mai precise).

Aplicarea transformatei Hilbert asupra setului de date duce, într-o prima instanţă, la aplicarea transformatei şi asupra coeficienţilor wavelet la toate scalele. Astfel nu se poate face o optimizare per scală. Prin implementarea arborelui dual, transformata Hilbert scalează simultan cu funcţia de scalare wavelet, eliminând astfel problemele optimizării.

3.1.3 Transformata wavelet complexă şi arborele dual

Arborele dual cu transformata wavelet complexă foloseşte două transformate wavelet reale. Prima transformată discretă reprezintă partea reală a transformatei complexe, iar cea de-a două partea imaginară.

Cele două transformate wavelet discrete folosesc două seturi diferite de filtre, fiecare set satisfăcând condiţia de reconstrucţie. Cele două seturi de filtre sunt astfel construite încât împreună să inducă transformatei wavelet complexe un caracter aproximativ analitic.

Fie $h_0(n)$, $h_1(n)$ perechile de filtrele trece jos/trece sus pentru bancurile de filtre superioare, şi $g_0(n)$, $g_1(n)$ perechile de filtre trece jos/trece sus pentru bancurile de filtre inferioare. Se vor nota cele două wavelet-uri discrete asociate cu cele două transformate wavelet discrete prin $\psi_h(t)$ şi $\psi_g(t)$.

În plus faţă de satisfacerea condiţiilor de reconstrucţie, filtre sunt proiectate astfel încât wavelet-ul complex:

$$\psi(t) = \psi_h(t) + \psi_g(t)$$

să fie aproximativ analitic. În mod echivalent, sunt astfel proiectate încât $\psi_g(t)$ să fie aproximativ transformata Hilbert a lui $\psi_h(t)$:

$$\psi_g(t) \approx \mathbb{H}\{\psi_h(t)\}$$

Bancurile de filtre au coeficienţi reali, deci nu este necesară aplicarea unei aritmetici complexe pentru a implementa arborele dual complex. O altă caracteristică a arborelui dual o reprezintă faptul că nu este critic eşantionat. Pentru 1D, el este două ori expansiv datorită datelor de ieşire ce reprezintă exact dublul datelor de intrare.

Inversa transformatei complexe cu arbore dual se obţine printr-un proces similar cu cea directă. Pentru a inversa partea imaginara şi cea reală se foloseşte inversa fiecărei transformate wavelet discrete, obţinându-se astfel două semnale reale. Aceste două semnale sunt apoi mediate pentru a obţine rezultatul final.

Dacă cele două transformate discrete reale sunt reprezentate cu ajutorul matricelor pătratice F_h şi F_g, atunci transformata complexă wavelet cu arbore dual poate fi reprezentată de matricea dreptunghiulară:

$$F = \begin{bmatrix} F_h \\ F_g \end{bmatrix}$$

Dacă vectorul x reprezintă un semnal real, atunci $\psi_h = F_h x$ reprezintă partea reală şi $\psi_g = F_g x$ reprezintă partea imaginară a arborelui dual complex. Coeficienţii complecşi sunt daţi de relaţia $\psi_h + j\psi_g$. Inversa, la ştanga, a lui F este data de relaţia:

$$F^{-1} = \frac{1}{2}\begin{bmatrix} F_h^{-1} & F_g^{-1} \end{bmatrix}$$

se verifică prin:

$$F^{-1} \times F = \frac{1}{2}\begin{bmatrix} F_h^{-1} & F_g^{-1} \end{bmatrix} \times \begin{bmatrix} F_h \\ F_g \end{bmatrix} = \frac{1}{2}[I + I] = I$$

Dacă se distribuie factorul de $1/2$ transformatei directe şi celei inverse se obţine:

$$F = \frac{1}{\sqrt{2}}\begin{bmatrix} F_h \\ F_g \end{bmatrix}; \quad F^{-1} = \frac{1}{\sqrt{2}}\begin{bmatrix} F_h^{-1} & F_g^{-1} \end{bmatrix}.$$

Dacă cele două transformate discrete reale sunt ortonormate, atunci transpusa lui F_h este inversa sa $F_h^t \times F_h = I$, proprietate valabilă şi pentru F_g. În acest caz transpusa matricei dreptunghiulare F este de asemenea inversa la ştanga a lui F, $F^t \times F = I$. Deci inversa arborelui dual complex poate fi calculată folosind transpusa transformării directe.

Transformata cu arbore dual ţine separate părţile reale şi imaginare ale

coeficienţilor wavelet-ului complex. Totuşi, aceşti coeficienţi pot fi calculaţi explicit folosind următoarele formule:

$$F_c = \frac{1}{2} \begin{bmatrix} I & jI \\ I & -jI \end{bmatrix} \times \begin{bmatrix} F_h \\ F_g \end{bmatrix},$$
$$F_c^{-1} = \frac{1}{2} \begin{bmatrix} F_h^{-1} & F_g^{-1} \end{bmatrix} \times \begin{bmatrix} I & I \\ -jI & jI \end{bmatrix}.$$

Se observă că matricea sumă/diferenţa complexă este unitară (conjugata transpusă a sa este chiar inversa). Deci dacă cele două transformate discrete reale sunt ortonormate, atunci arborele dual complex satisface relaţia:

$$F_c^* \times F_c = I$$

Când cele două transformate discrete reale sunt ortonormate şi este inclus şi factorul $1/\sqrt{2}$, arborele dual complex capătă proprietatea teoremei energiei lui Parseval: energia semnalului de intrare este egala cu energia în domeniul frecvenţei:

$$\sum_{j,n} \left(|d_h(j,n)|^2 + |d_g(j,n)|^2 \right) = \sum_n |x(n)|^2$$

Arborele dual complex este astfel uşor de implementat. Datorita faptului că nu există schimb de date între cele două transformate discrete reale, cele două pot fi implementate folosind metodele existente pentru DWT. Transformata poate fi astfel şi paralelizată pentru o putere de calcul crescută. În plus, pentru că arborele dual complex foloseşte cele două transformate reale, utilizarea sa poate fi explicată cu ajutorul teoriei si practicii curente a transformatelor reale.

3.2 Bancuri de filtre

Arborele dual complex necesită o proiectare nouă pentru bancurile de filtre. În primul rând necesită o pereche de filtre alese astfel încât wavelet-urile corespunzătoare să alcătuiască o pereche aproximativă de transformate Hilbert. Filtrele folosite pentru transformate reale nu sunt proiectate pentru arborele dual şi în general nu satisfac proprietăţile necesare implementării lui.

Proiectarea bancurilor de filtre trebuie să îndeplinească următoarele condiţii:

- întârzierea cu aproximativ jumătate de eşantion
- proprietatea de reconstrucţie

- suport finit (filtre FIR)
- atenuare buna în banda de oprire

Majoritatea proprietăţilor necesare sunt moştenite de la wavelet-urile complexe corespunzătoare. Mai mult, bancurile de filtre aplicate în prima etapă trebuie să difere de cele aplicate în restul etapelor. În continuare vor fi descrise proprietăţile şi cerinţele noi pentru proiectarea acestor filtre.

3.2.1 Condiţia de întârziere

Bancurile de filtre moştenesc proprietăţile wavelet-urilor. Din această cauză apar aceleaşi probleme de proiectare şi în cazul filtrelor. De exemplu un wavelet are K momente de anihilare dacă funcţia de transfer a filtrului trece jos este de forma:

$$H_0(z) = (1+z)^K Q(z), \forall Q(z)$$

Filtrele arborelui dual complex trebuie proiectate astfel încât wavelet-urile asociate să formeze o pereche aproximativă a transformatei Hilbert. Legătura între filtre şi wavelet-uri este exprimată prin următoarele relaţii:

$$\psi_h(t) = \sqrt{2}\sum_n h_1(n)\,\phi_h(t),$$
$$\phi_h(t) = \sqrt{2}\sum_n h_0(n)\,\phi_h(t),$$
$$h_1(n) = (-1)^n h_0(d-n).$$

$\psi_g(t)$, $\phi_g(t)$ şi $g_1(n)$ sunt definite similar. Pentru că wavelet-urile depind de funcţiile de scalare, şi funcţiile de scalare depind de filtre implicit, găsirea unei soluţii pentru proiectare este dificilă. În lucrările de specialitate[7] s-a demonstrat că cele două filtre trece-jos trebuie să satisfacă o proprietate destul de simplă: un filtru trebuie să fie aproximativ egal cu o translaţie de jumătate a celui de-al doilea.

$$g_0(n) \approx h_0(n-0,5) \rightarrow \psi_g(t) \approx \mathbb{H}\{\psi_h(t)\}$$

3.2.2 Alegerea bancurilor de filtre pentru prima etapă

Dacă aceleaşi filtre, ce asigură proprietatea de reconstrucţie, sunt aplicate pentru fiecare etapă atunci primele etape ale bancurilor de filtre nu vor fi

aproximativ analitice.

Acest efect apare din cauza alegerii filtrelor în funcţie de condiţia de translatare cu jumătate de eşantion, ce a fost impusă prin corelarea cu proprietatea de transformare Hilbert wavelet-urilor ($\psi_g(t) \approx \mathbb{H}\{\psi_h(t)\}$). Totuşi aceste funcţii wavelet sunt folositoare pentru a procesa bancurile de filtre la etape mari, cu $j \to \infty$.

Se arată că dacă filtrele trece-jos satisfac condiţia de translatare cu jumătate de eşantion, atunci şi funcţiile de scalare satisfac această condiţie:

$$g_0(n) \approx h_0(n-0,5) \to \phi_g(t) \approx \phi_h(n-0,5)$$

Extinderea wavelet reală pentru un semnal $x(t)$ aduce translatări cu un număr întreg a funcţiei de scalare. Astfel condiţia de întârziere a funcţiei de scalare presupune că translatările întregi ale lui $\phi_g(t)$ să fie la jumătate între translatările întregi ale lui $\phi_h(t)$. Adică cele două funcţii satisfac o proprietate de întrepătrundere. Pentru ca forma discretă a arborelui dual complex să fie aproximativ analitică la fiecare etapă j, este necesar ca bancurile de filtre ale arborelui dual să duplice această proprietate.

Astfel filtrele care sunt proiectate să îndeplinească proprietatea de translaţie cu jumătate de eşantion nu ar trebui folosite în prima etapă. Pentru prima etapă trebuie satisfăcută o condiţie similară cu:

$$g_0^{(1)} \approx h_0^{(1)}(n-1)$$

folosind aceleaşi set de filtre în ambele ramuri. Este necesară numai translatarea unui set de filtre cu un eşantion relativ la celalalt. Mai mult, orice banc de filtre ce satisface proprietatea de reconstrucţie poate fi folosit în prima etapa.

3.3 Implementare

3.3.1 Transformata wavelet cu arbore dual 1D

Transformata wavelet complexă cu arbore dual este implementată folosind două transformate wavelet discrete în paralel asupra aceluiaşi set de date (vezi figura 6).

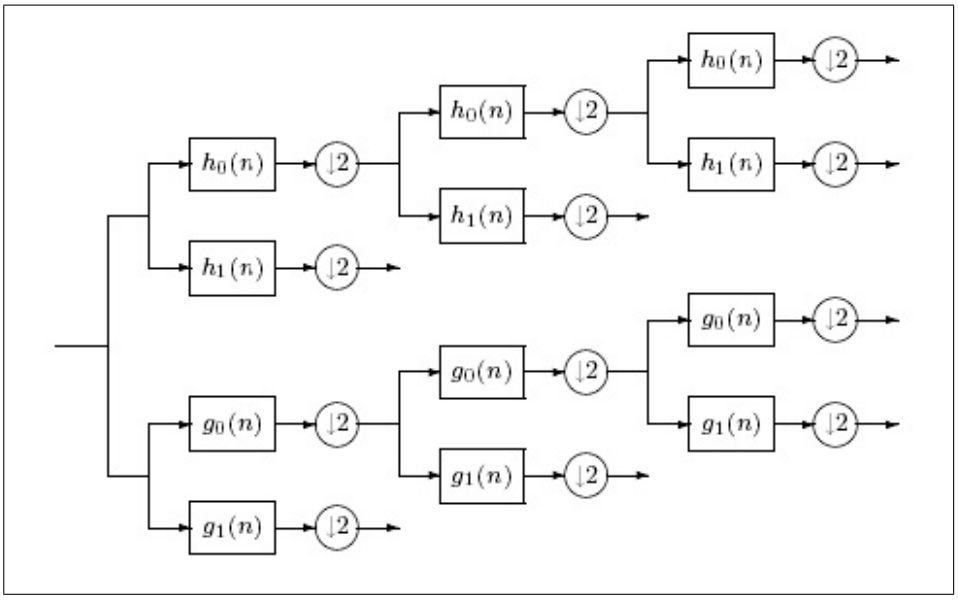

Figura 6: Aplicarea filtrelor de analiză

Transformata este de două ori expansivă pentru că pentru un semnal de N eşantioane întoarce 2N coeficienţi pentru transformata discretă. Dacă filtrele pentru transformata discretă superioară şi inferioară sunt aceleaşi, atunci nu se obţine nici un avantaj. Cu toate acestea, dacă filtrele sunt proiectate într-un mod aparte (discutat în secţiunile anterioare), atunci semnalele subbenzilor transformatei discrete superioare pot fi interpretate drept partea reală a transformatei wavelet complexe, şi semnalele subbenzilor transformatei discrete inferioare pot fi interpretate drept partea imaginară.

Anumite filtre sunt proiectate astfel încât partea superioară poate fi aproximată ca transformata Hilbert a celei inferioare. În acest caz transformata wavelet cu arbore dual este aproape invariantă la transpunere, diferit de transformarea discretă. În plus arborele dual poate fi folosit pentru a implementa transformate wavelet 2D astfel încât fiecare wavelet să fie orientat.

Bancuri de filtre

Filtrele folosite în prima etapă a transformatei wavelet complexe cu arbore dual trebuie să fie diferite de filtrele folosite în restul etapelor (discutat în secţiunile anterioare).

Transformata wavelet este aplicată similar celei discrete. Frecvenţele joase ale fiecărei etape sunt prelucrate de filtrele asociate părţii superioare şi

celei inferioare a arborelui dual.

Transformata wavelet inversă reconstruieşte semnalul iniţial prin aplicarea filtrelor de sinteza asupra fiecărei părţi, atât superioare cât şi inferioare. Proprietatea de reconstrucţie este asigurată prin alegerea filtrelor într-un mod specific discutat anterior.

Wavelet-urile asociate părţii reale şi celei imaginare pot fi evidenţiate prin aplicarea transformatei complexe asupra unui semnal nul, iar apoi aplicarea transformatei inverse. Semnalele obţinute la ieşire sunt similare celor indicate în figura 7.

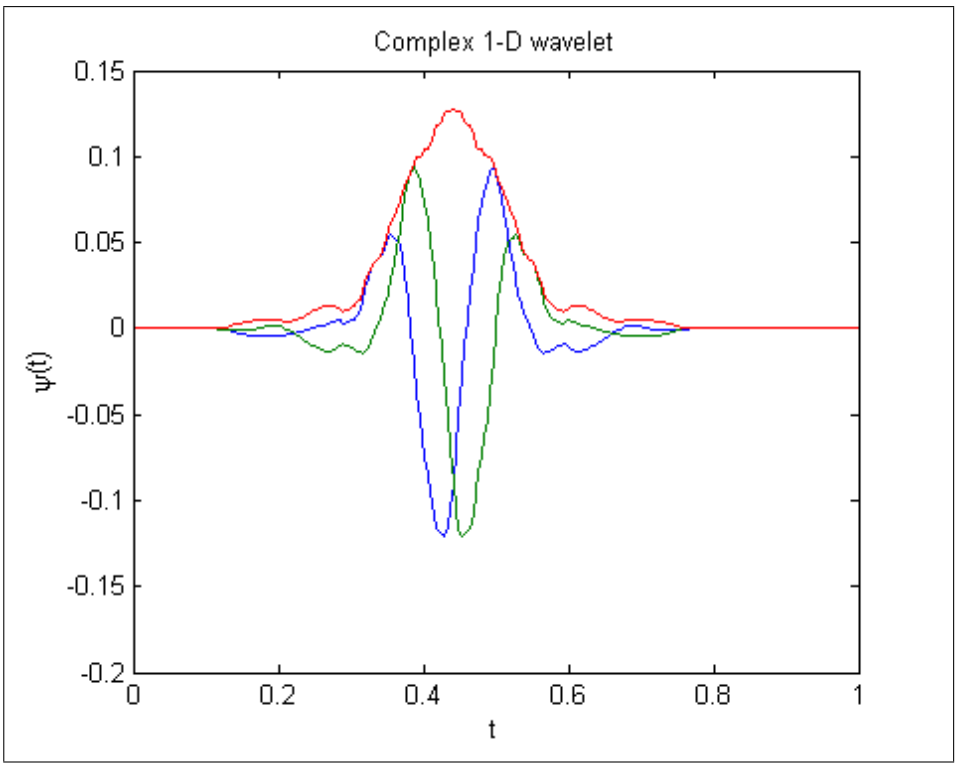

Figura 7: Wavelet complex 1D

Pe lângă cele două wavelet-uri (reale şi imaginare) s-a reprezentat şi amplitudinea lor.

3.3.2 Transformata wavelet cu arbore dual 2D

Transformata wavelete reală

Transformata reală este implementată folosind două transformate discrete 2D în paralel. Pentru fiecare pereche de subbenzi se calculează suma şi diferenţa.

Imaginea este recuperată folosind transformata inversă, iar proprietatea de reconstrucţie perfectă este asigurată prin alegerea filtrelor.

Cele şase wavelet-uri asociate transformatei reale cu arbore dual sunt calculate similar ca în exemplele precedente. Reprezentarea grafică a acestor wavelet-uri este similară cu cea din figura 9.

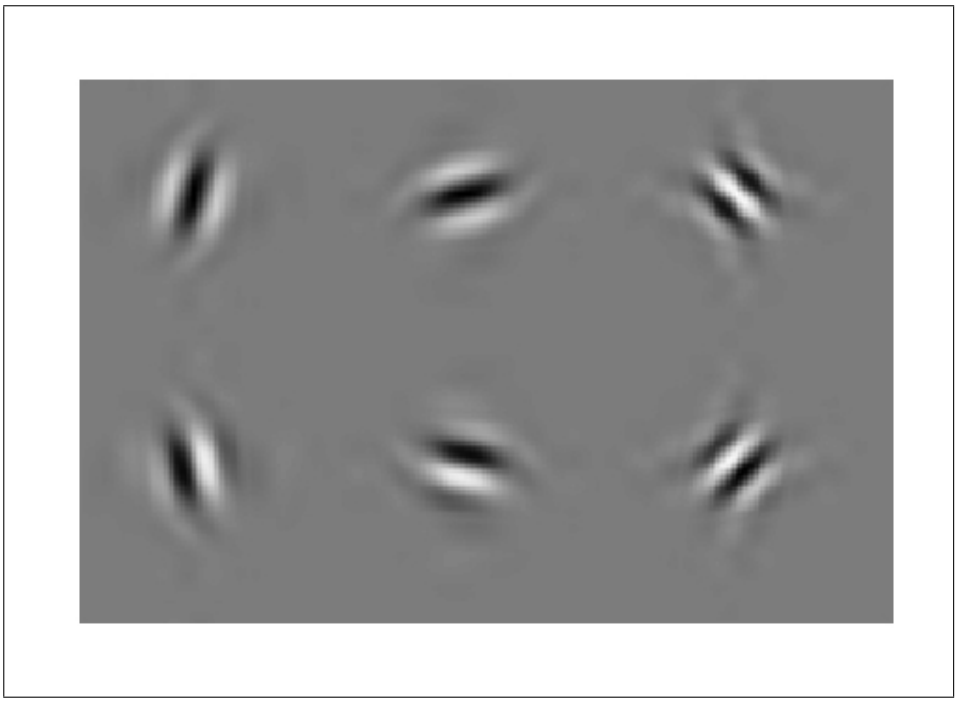

Figura 8: Wavelet real 2D

Se observă că fiecare wavelet este orientat într-o altă direcţie şi nu prezintă fenomenul de checkboard (în care un wavelet are două direcţii suprapuse) întâlnit la transformata discretă wavelet 2D. Fiecare subbandă a transformării 2D cu arbore dual corespunde unei orientări specifice.

Transformata wavelet complexă

Transformata wavelet complexă cu arbore dual prezintă tot şase wavelet-uri cu orientări distincte, dar în acest caz există câte două wavelet-uri pentru fiecare orientare. În fiecare direcţie una dintre wavelet-uri poate fi considerată partea reală a transformatei complexe, iar cealaltă partea imaginară.

Pentru că versiunea complexă are de două ori mai multe wavelet-uri decât versiunea reală a transformării, varianta complexă este de patru ori

expansivă.

Arborele dual 2D complex este implementat cu ajutorul a patru transformate wavelet discrete ce operează în paralel. Cu toate acestea, bancuri de filtre diferite sunt folosite de-a lungul liniilor şi coloanelor. Ca şi în cazul real, suma şi diferenţa imaginilor din subbenzi sunt calculate pentru a obţine wavelet-urile orientate.

Imaginea x este recuperată folosind transformata inversă ce operează similar celor descrise anterior, dar în concordanţă cu transformata directă 2D cu arbore dual. Proprietatea de reconstrucţie este asigurată, din nou, datorită filtrelor alese.

Cele doisprezece wavelet-uri asociate cu cele din arborele dual real sunt obţinute similar prin aplicarea transformărilor asupra unui semnal nul. Imaginea acestor wavelet-uri este similară cu cea prezentată în figura 9.

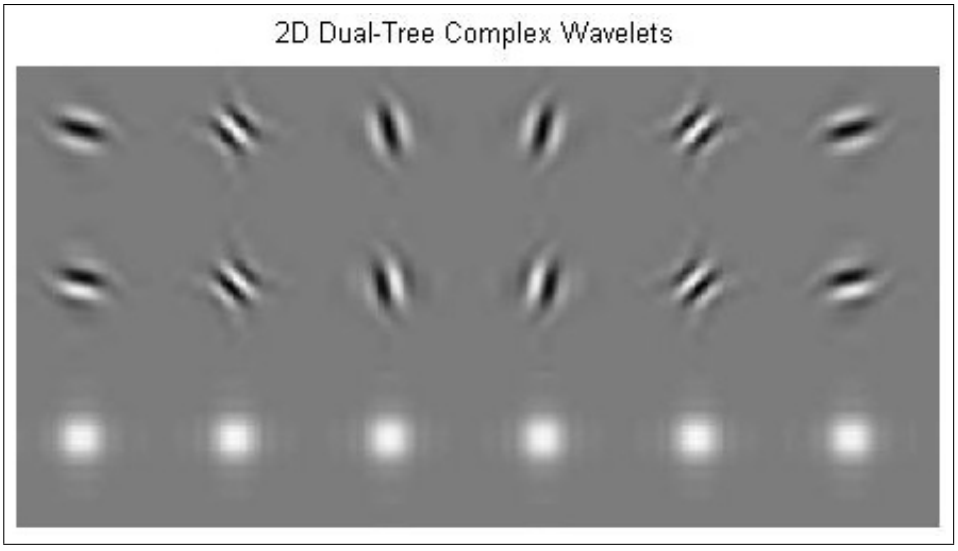

Figura 9: Wavelet complex 2D

Se observă, din figura de mai sus, ca wavelet-urile sunt orientate în aceiaşi direcţie cu cele din arborele dual, precum şi faptul că există câte o pereche pentru fiecare orientare. Dacă cele şase wavelet-uri de pe primul rând sunt asociate părţii reale, iar cele de pe al doilea părţii imaginare, atunci amplitudinea celor şase wavelet-uri complexe este prezentată în al treilea rând. Se observă că amplitudinea nu prezintă un caracter oscilatoriu.

4 Arborele dual cu densitate dubla

Arborele dual poate fi extins prin îmbinarea acestuia cu transformata wavelet discretă cu densitate dublă (DD). Această transformată DD se comportă mult mai bine decât transformata discretă, deţinând proprietatea de invarianţă la translaţii. Transformata DD are la bază o singură funcţie de scalare şi două wavelet-uri distincte.

Cele două wavelet-uri sunt proiectate astfel încât să fie deplasate la jumătate una faţă de cealaltă. Consecinţa acestui fapt este că translaţiile întregi ale unui wavelet ajung la mijlocul translaţiilor celeilalte:

$$\psi_2(t) \approx \psi_1(t - 0,5)$$

Astfel transformata continuă este mult mai bine aproximată cu ajutorul transformatei DD, faţă de transformata discretă, având în vedere că se dispune de mai multe wavelet-uri decât necesar pentru acest proces.

Există mai multe lucrări [6, 2] ce tratează subiectul proiectării diferitelor filtre de tip FIR şi legătura lor cu wavelet-urile folosite pentru transformarea DD. Continua cercetare şi experimentare cu acest tip de transformare wavelet a dus la următoarele concluzii: wavelet-urile folosite cu transformata DD sunt foarte line, cu un suport mic, iar transformata este aproximativ invariantă la translaţii.

Dacă se iau în vedere transformata DD şi cea cu arbore dual se observă câteva similarităţi. În principal ambele sunt redundante cu 2, ambele prezintă invarianţă la translaţii şi ambele se bazează pe filtre FIR ce deţin proprietatea de reconstrucţie. Evident, separat, amândouă se comportă mult mai bine în procesarea imaginilor faţă de transformata discretă simplă.

Din păcate există şi diferenţe ce trebuie remarcate între cele două implementări:

- pentru transformata discretă wavelet cu arbore dual, cele două wavelet-uri reprezintă o transformare aproximativ Hilbert, pe când pentru transformata DD ele sunt deplasate cu jumătate una faţă de altă
- pentru transformata cu arbore dual se dispune de mai puţine grade de libertate în proiectare, fiind constrânsă de transformarea Hilbert, pe când în transformarea DD se dispune de mai multe grade de libertate
- structurile filtrelor pentru cele două transformate sunt diferite
- transformata cu arbore dual poate fi interpretată drept o transformare complexă wavelet ce aduce multe avantaje în procesarea de imagini, pe când transformata DD nu are această proprietate

- transformata cu arbore dual poate fi folosită pentru transformări în 2D cu ajutorul wavelet-urilor orientate, proprietate care este necesară prelucrării imaginilor, pe când transformata DD nu poate fi utilizată în acest scop

Datorită faptului că cele două tipuri de transformări prezintă avantaje foarte atrăgătoare pentru a fi folosite în procesarea imaginilor, s-au elaborat mai multe lucrări ce au încercat să le îmbine pe cele două spre a obţine un nou tip de analiză pentru eliminarea zgomotului din imagini. Această îmbinare preia avantajele din ambele transformate şi fixează un mod robust de proiectare a filtrelor aferente.

4.1 Proiectare

Transformata wavelet discretă DD cu arbore dual (DD DT-DWT, double density dual-tree discrete wavelet transform) este proiectată pentru a deţine simultan avantajele transformatei cu arbore dual şi a celei cu DD. Pentru aceasta transformata are la bază două funcţii de scalare distincte şi patru wavelet-uri, de asemenea, distincte:

$\psi_{h,i}(t)$, $\psi_{g,i}(t)$, $i = 1, 2$
unde cele două wavelet-uri $\psi_{h,i}(t)$ sunt deplasate cu jumătate între ele, regula aplicându-se şi pentru $\psi_{g,i}(t)$:

$$\psi_{h,1}(t) = \psi_{h,2}(t - 0,5),\ \psi_{g,1}(t) = \psi_{g,2}(t - 0,5)$$

iar cele două wavelet-uri $\psi_{g,1}(t)$ şi $\psi_{h,1}(t)$ alcătuiesc aproximativ o pereche Hilbert, la fel şi pentru $\psi_{g,2}(t)$ si $\psi_{h,2}(t)$:

$$\psi_{g,1}(t) \approx \mathbb{H}\{\psi_{h,1}(t)\},\ \psi_{g,2}(t) \approx \mathbb{H}\{\psi_{h,2}(t)\}$$

Îmbinarea într-o singură transformată a acestor proprietăţi ridică mai multe probleme de implementare. Există mai multe soluţii ce au fost descrise în lucrări de specialitate. Una dintre aceste soluţii implică factorizarea spectrală a filtrelor şi completarea acestora cu ajutorul unei matrici paraunitare. În secţiunea următoare vor fi descrise principale concluzii alături de parametrii de implementare rezultaţi din aceste tratate.

4.2 Implementare

Structura bancurilor de filtre asociată cu transformata cu arbore dual cu DD conţine două ramuri supraeşantionate, similar cu arborele dual simplu. Diferenţa majoră este aplicarea unui set de trei filtre distincte asupra fiecărei ramuri.

Procesul de analiză este legat de cel de sinteză pentru a reconstrui imaginea/semnalul primit. Filtrele de analiză subeşantionează cu doi semnalul pentru ca la sfârşit filtre de analiză să-l supraeşantioneze cu doi şi să recupereze semnalul (vezi figura 10).

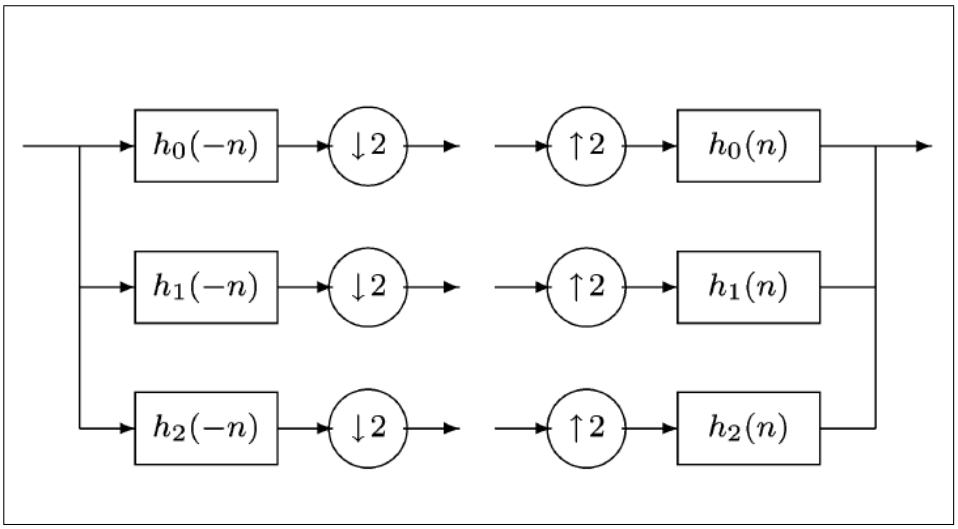

Figura 10: Filtrele supraeşantionate de analiză şi sinteză (DD DT-DWT)

Aceste filtre sunt iterate în cadrul procesării pentru o analiză cât mai în detaliu. Parcurgerea fiecărei etape implică aplicarea setului de filtre de analiză sau sinteză, în funcţie de momentul în care se află procesul de procesare a imaginii. La fiecare etapă sunt preluate eşantionele rezultate prin aplicarea filtrelor trece-jos din etapa anterioară (vezi figura 11).

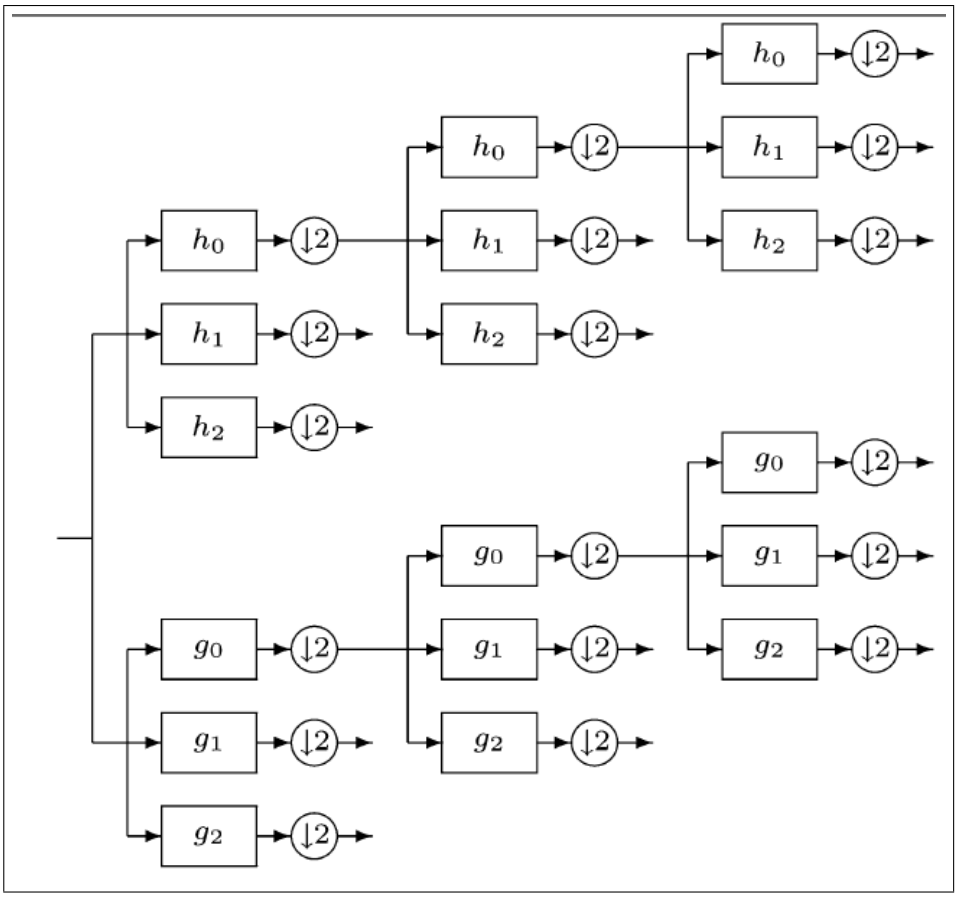

Figura 11: Iterarea filtrelor (DD DT-DWT)

Se notează filtrele din prima ramură cu $h_i(n)$ şi cele din a două cu $g_i(n)$, unde $i = 0, 1, 2$.

Filtrele de sinteză sunt deduse din cele de analiză într-un mod similar cu cel prezentat în secţiunile anterioare, având în vedere că ele sunt versiuni inversate în timp ale celor de analiză.

Cele şase filtre FIR trebuie să îndeplinească următoarele proprietăţi:

- reconstrucţie perfectă
- wavelet-urile trebuie să formeze perechi de transformate Hilbert
- wavelet-urile trebuie să aibă un număr specificat de momente de anihilare
- filtrele trebuie să aibă suport mic

Condiţiile de reconstrucţie perfectă, pentru fiecare set de filtre, sunt exprimate matematic sub următoarea formă:

$$\sum_{i=0}^{2} H_i(z) H_i(1/z) = 2$$

$$\sum_{i=0}^{2} H_i(z) H_i(-1/z) = 0 \text{ şi:}$$

$$\sum_{i=0}^{2} G_i(z) G_i(1/z) = 2$$

$$\sum_{i=0}^{2} G_i(z) G_i(-1/z) = 0$$

unde s-a notat cu $H_i(z)$ transformata Z a lui $h_i(n)$:

$$H_i(z) = ZT\{h_i(n)\} = \sum_n h_i(n) z^{-n}$$

Funcţiile wavelet şi cele de scalare sunt definite implicit cu ajutorul ecuaţiilor wavelet şi a celor de dilatare:

$$\phi_h(t) = \sqrt{2} \sum_n h_0(n) \psi_h(2t - n)$$

$$\psi_{h,1}(t) = \sqrt{2} \sum_n h_1(n) \psi_h(2t - n)$$

$$\psi_{h,2}(t) = \sqrt{2}\sum_{n} h_2(n)\,\psi_h(2t-n)$$

Iar $\phi_g(t)$ şi $\psi_{g,i}(t)$ sunt definite similar. Pentru a îndeplini următoarea condiţie, ceea ce presupune ca perechile wavelet să reprezinte transformări de tip Hilbert, este nevoie ca un set suplimentar de egalităţi să fie împlinit:

$$\psi_{g,1}(t) = \mathbb{H}\{\psi_{h,1}(t)\}$$

$$\psi_{g,2}(t) = \mathbb{H}\{\psi_{h,2}(t)\}$$

Explicit aceste relaţii pot fi reprezentate astfel (având în vedere definiţia transformatei Hilbert):

$$\hat{\psi}_{g,i}(\omega) = \begin{cases} -j\hat{\psi}_{h,i}(\omega) & ,\omega > 0 \\ j\hat{\psi}_{h,i}(\omega) & ,\omega < 0 \end{cases}$$

4.3 Filtre

Setul de filtre ce poate fi folosit împreună cu structura transformatei wavelet cu arbore dual cu DD trebuie, la rândul lui, să satisfacă un număr de condiţii.

Filtrele sunt determinate, în principiu, de numărul de momente de anihilare ($K_i, i = 0, 1, 2$). Pentru a obţine wavelet-uri foarte line se impune următoarea condiţie:

$$K_0 > K_1,\ K_0 > K_2$$

iar pentru echilibrarea activităţii între cele patru wavelet-uri şi a calculului se va alege $K_1 = K_2$.

Condiţiile ce trebuie îndeplinite de bancurile de filtre FIR cu suport mic sunt:

- reconstrucţia perfectă
- relaţiile Hilbert
- momente nule pentru K_1 şi K_2 date, cu $0 \leq k \leq K_i - 1$:

$$\int t^k \psi_{h,i}(t)\,dt = \int t^k \psi_{g,i}(t)\,dt = 0,\ i = 1, 2$$

- zerouri în frecvenţele joase la $\omega = \pi$, unde $H_0(z)$ şi $G_0(z)$ trebuie să fie divizibile cu $(1+z^{-1})^{K_0}$
- proprietatea de translatare

Având în vedere condiţiile şi parametrii discutaţi, filtrele pentru transformata wavelet cu arbore dual cu DD pot fi definite sub următoarea formă:

$$H_0(z) = D(z)\left(1+z^{-1}\right)^{K_0} Q_0(z)$$

$$H_1(z) = (-z)^{-L} D(-1/z)\left(1-z^{-1}\right)^{K_1} Q_1(z)$$

$$H_2(z) = (-z)^{-L} D(-1/z)\left(1-z^{-1}\right)^{K_2} Q_2(z)$$

$$G_0(z) = z^{-L} D(1/z)\left(1+z^{-1}\right)^{K_0} Q_0(z)$$

$$G_1(z) = D(-z)\left(1-z^{-1}\right)^{K_1} Q_1(z)$$

$$G_2(z) = D(-z)\left(1-z^{-1}\right)^{K_2} Q_2(z)$$

Echivalent se poate scrie:

$$h_0(n) = d(n)\, s_0(n)\, q_0(n)$$

$$h_1(n) = (-1)^n\, d(L-n)\, s_1(n)\, q_1(n)$$

$$h_2(n) = (-1)^n\, d(L-n)\, s_2(n)\, q_2(n)$$

$$g_0(n) = d(L-n)\, s_0(n)\, q_0(n)$$

$$g_1(n) = (-1)^n\, d(n)\, s_1(n)\, q_1(n)$$

$$g_2(n) = (-1)^n\, d(n)\, s_2(n)\, q_2(n)$$

unde:

$$s_0(n) = \begin{pmatrix} K_0 \\ n \end{pmatrix} = \frac{K_0!}{(K_0-n)!n!}$$

$$s_i(n) = (-1)^n \begin{pmatrix} K_i \\ n \end{pmatrix}$$

Cu această structură, necunoscutele $D(z)$ şi $Q_i(z)$ sunt determinate pentru a îndeplini condiţiile menţionate mai sus, cu L factorul de aproximare a reconstrucţiei perfecte. Metodele prin care se determină aceste valori sunt explicate şi detaliate în lucrările de specialitate.

5 Prelucrarea imaginilor cu transformarea wavelet

Pentru a verifica rezultatele teoretice au fost implementate o serie de rutine în Matlab. Aceste rutine aduc suportul necesar pentru a observa rezultatele obţinute prin aplicarea mai multor filtre asupra imaginilor cu zgomot.

Imaginile folosite sunt monocrome pentru a pune în evidenţă diferenţele dintre imaginea iniţială, imaginea cu zgomot şi imaginea prelucrată pentru îndepărtarea zgomotului. Aplicaţia prelucrează imaginile transpunându-le în matrici mari (ex. 512 linii şi coloane).

Rezultatele obţinute în urma aplicării filtrelor asupra unei singure imagini cu zgomot sunt greu de identificat fără a folosi o metodă specializată. De aceea se alege o metodă numerică de a verifica performanţele filtrelor pentru a le putea compara la sfârşit. Astfel imaginea iniţială este reţinută şi comparată cu cea prelucrată cu ajutorul algoritmului PSNR (Peak Signal-to-Noise Ratio).

În continuare vor fi prezentate în detaliu rutinele Matlab de eliminare a zgomotului cât şi calcularea erorii, în final fiind exemplificate diferite rezultate grafice cât şi comparaţiile între diferitele filtre.

5.1 Aplicaţia Matlab — arborele dual

5.1.1 Iniţializarea datelor – imginit

Pentru simplificarea încărcării imaginii iniţiale şi transpunerii sale în matricea bidimensională ce urmează a fi prelucrata se va folosi rutina imginit.

Această rutină are ca parametrii de intrare imaginea (ex. locaţia fizică de pe hard-disk sau orice alt mediu de stocare suportat pe platforma Windows) şi un set de flag-uri ce stabilesc ce filtre se vor folosi pentru prelucrarea imaginii cu ajutorul transformatei wavelet.

Parametrii de ieşire sunt semnalul imaginii iniţiale (ex. matricea originală), imaginea alterată (cu zgomot) şi setul de filtre. Matricea originală va fi utilizată pentru calcularea erorii la sfârşitul procesării. Imaginea alterată este folosită mai departe în funcţiile de eliminare a zgomotului. Setul de filtre reprezintă perechea pentru prima etapă a arborelui dual cât şi pentru următoarele.

```
function [xo, xn, filter] = imginit(img, flags)
```

Rutina citeşte întâi imaginea din spaţiul fizic cu ajutorul funcţiei imread. Această funcţie este oferită de bibliotecile Matlab şi primeşte ca parametrii calea către imagine şi formatul ei. Dacă formatul nu este specificat atunci

funcţia încearcă să-l ghicească. Dacă imaginea este grayscale, funcţia întoarce o matrice bidimensională. Altfel, în funcţie de tip matricea este tridimensională şi necesită o prelucrare suplimentară pentru a transforma paleta coloră într-una grayscale.

După apelarea funcţiei imread se aplică asupra ieşirii acesteia o altă funcţie Matlab: double. Această transformare este necesară pentru a asigura că tipul elementelor matricei să fie în virgulă mobilă.

```
xo = double(imread(img));
```

În continuare se creează imaginea cu zgomot prin adăugarea unor valori aleatoare imaginii originale. Această noua matrice este salvată separat pentru a fi prelucrata de rutinele de eliminare a zgomotului.

```
xn = xo + 20*randn(size(xo));
```

Etapa următoare setează bancurile de filtre în funcţie de valoarea parametrului de intrare flags.

```
[Faf, Fsf] = FSfarras;
if flags == 0
        [af, sf] = dualfilt1;
elseif flags == 1
        [af, sf] = htpwb_N10K4L5;
...
filter = {Faf, Fsf, af, sf};
```

În final se afişează imaginea cu zgomot pentru a observa diferenţele faţă de imaginea originală.

```
colormap(gray);
imagesc(xn);
```

5.1.2 Rutina principală – mydenoise

Odată obţinute datele iniţiale, în principal imaginea cu zgomot, se poate trece la pasul următor: eliminarea perturbaţiilor cu ajutorul transformatei wavelet şi a bancurilor de filtre asociate.

Rutina principală ce trebuie apelată primeşte ca parametrii de intrare matricea bidimensională reprezentând imaginea cu zgomot, nivelul de threshold, numărul de etape pentru arborele dual, bancul de filtre şi un set de flag-uri ce asigură un anume flux pe parcursul rutinei.

După prelucrare rutina întoarce o matrice bidimensională reprezentând imaginea prelucrata (fără zgomot).

```
function y = mydenoise(x,T, J, filters , flags)
```

Primele instrucţiuni ale rutinei reprezintă extragerea filtrelor de analiză şi sinteză. Prima celulă a parametrului filters reprezintă filtrele de analiză pentru prima etapă a arborelui dual wavelet, iar a doua celulă este filtrul pereche de sinteză. Filtrele de analiză pentru celelalte etape sunt conţinute de cea de-a treia celulă, în timp ce filtrele de sinteză corespondente acesteia sunt cuprinse de cea de-a patra celulă. Astfel rutina Matlab execută:

```
Faf = filters{1};
Fsf = filters{2};
af = filters{3};
sf = filters{4};
```

În acest moment se poate calcula transformata wavelet complexă cu arbore dual cu ajutorul filtrelor de analiză şi a numărului de etape dorite. În continuare va fi descrisă această rutină, pornind de la apelul acesteia:

```
w = cplxdual2D(x,J,Faf,af);
```

Mai departe se parcurg coeficienţii wavelet pentru aplicarea thresholding-ului. Parcurgerea se face întâi pe scală, apoi pe primul set de subbenzi iar la sfârşit pe al doilea.

```
% loop thru scales:
for j = 1:J
    % loop thru subbands
    for s1 = 1:2
        for s2 = 1:3
```

Pentru fiecare iteraţie se construieşte numărul complex format din partea reală a transformatei wavelet şi partea imaginară:

```
C = w{j}{1}{s1}{s2} + I*w{j}{2}{s1}{s2};
```

Notatii:

- **j** – scala
- **i** – ia valoarea de 1 sau 2 şi reprezintă partea reală, respectiv, imaginară a transformatei wavelet
- **s1** – ia valoarea de 1 sau 2
- **s2** – ia valori între 1 si 3
- **(s1, s2)** – perechea reprezintă orientarea wavelet

În funcţie de flag-uri, se aplică asupra numărului complex thresholding-ul ales (soft sau hard). Implementarea celor două tipuri de thresholding va fi explicată în continuare.

```
if flags == 0
        C = soft(C, T);
else
        C = hard(C, T);
end
```

La sfârşitul iteraţiilor se construieşte transformata wavelet complexă inversă. Pentru a o obţinerea acesteia se folosesc: transformata wavelet complexă directă, numărul de etape, filtrele de sinteză iniţiale şi cele generale. Rutina icplxdual2D se apelează prin intermediul acestor parametrii, ea reuşind să producă astfel matricea asociată imaginii fără zgomot.

```
y = icplxdual2D(w,J,Fsf,sf);
```

La sfârşitul rutinei mydenoise se afişează imaginea obţinută pentru a se observa diferenţele (evidente) între imaginea cu zgomot şi cea prelucrată:

```
imagesc(y)
colormap(gray)
axis image
```

5.1.3 Transformata wavelet complexă directă – cplxdual2D

Transformata wavelet complexă directă se obţine prin parcurgerea ambelor ramuri (reală şi imaginară) pentru fiecare etapă (scală). Această parcurgere are loc întâi pentru primul parametru ce determină orientarea (subbanda) iar apoi pentru cel de-al doilea.

```
for m = 1:2
        for n = 1:2
```

La fiecare pas se aplică filtrele pe linii şi coloane cu ajutorul rutinei afb2D ce va fi discutată în continuare.

```
[lo w{1}{m}{n}] = afb2D(x, Faf{m}, Faf{n});
for j = 2:J
        [lo w{j}{m}{n}] = afb2D(lo, af{m}, af{n});
end
```

Se observă că întâi sunt aplicate filtrele pentru prima etapă iar apoi succesiv filtrele dedicate restului de etape.

Se salvează la sfârşitul fiecărei iteraţii al subbandei ultimul filtru trece jos de la ultima etapă:

```
        w{J+1}{m}{n} = lo;
```

După aceasta se reia iteraţia pentru următoarea subbandă.

La sfârşit se efectuează suma şi diferenţa imaginilor din fiecare subbandă pentru a obţine wavelet-urile orientate. Acestea se obţin prin iterarea fiecărei etape de-a lungul fiecărei perechi de orientare.

```
for j = 1:J
  for m = 1:3
    [w{j}{1}{1}{m} w{j}{2}{2}{m}] =
        pm(w{j}{1}{1}{m},w{j}{2}{2}{m});
    [w{j}{1}{2}{m} w{j}{2}{1}{m}] =
        pm(w{j}{1}{2}{m},w{j}{2}{1}{m});
  end
end
```

5.1.4 Transformata wavelet complexă inversă – icplxdual2D

Imaginea este recuperată din transformarea wavelet directă folosind transformata inversă. Aceasta este implementată în rutină icplxdual2D.

Această rutină asigură proprietatea de reconstrucţie a transformatei. Calcularea erorii acestei reconstrucţii este detaliată în secţiunea următoare.

Rutina primeşte ca parametrii de intrare transformata wavelet directă, numărul de etape folosit, filtrele de sinteză pentru prima etapă şi cele pentru etapele următoare.

Rutină întoarce la ieşire matricea asociată imaginii prelucrate (fără zgomot).

```
function y = icplxdual2D(w, J, Fsf, sf)
```

În interiorul rutinei iteraţiile şi etapele de reconstrucţie a imaginii au loc în ordine inversă faţă de transformarea directă. Astfel întâi se vor calcula suma şi diferenţa între imaginilor conţinute de fiecare subbandă ce determină orientarea wavelet-urilor.

```
for j = 1:J
    for m = 1:3
        [w{j}{1}{1}{m} w{j}{2}{2}{m}] =
            pm(w{j}{1}{1}{m},w{j}{2}{2}{m});
        [w{j}{1}{2}{m} w{j}{2}{1}{m}] =
            pm(w{j}{1}{2}{m},w{j}{2}{1}{m});
    end
end
```

Ca şi înainte, întâi se iterează etapele iar apoi fiecare pereche de orientare. Urmează iniţializarea matricei asociată imaginii prelucrate.

```
y = zeros(size(w{1}{1}{1}{1})*2);
```

De aici poate începe reconstrucţia cu ajutorul filtrelor de sinteză pentru fiecare etapă, reţinând filtrele trece jos ale fiecărei perechi de orientare. Acestea vor fi folosite ca parametri pentru rutinele sfb2D ce aplică aceste filtre pe fiecare linie şi coloană a imaginilor conţinute în fiecare subbandă.

```
for m = 1:2
    for n = 1:2
        lo = w{J+1}{m}{n};
        for j = J:-1:2
            lo = sfb2D(lo, w{j}{m}{n}, sf{m}, sf{n});
        end
        lo = sfb2D(lo, w{1}{m}{n}, Fsf{m}, Fsf{n});
        y = y + lo;
    end
end
```

Din nou se poate observa că iteraţiile au loc în ordine inversă faţă de cele de la transformata directă. Întâi se aplică etapelor superioare iar în final ultimei etape, cu setul special de filtre de sinteză asociat.

5.1.5 Threshold - hard, soft

Tipul de thresholding aplicat imaginii poate fi de două tipuri: hard şi soft. Primul anulează eşantionul dacă nu depăşeşte nivelul de threshold iar al doilea îl atenuează funcţie de nivel, fără a-l face nul.

Cele două tipuri de threshold duc la rezultate diferite în eliminarea zgomotului din imagini. Rezultatele vor fi comparate în următoarea secţiune. Aici va fi prezentată implementarea lor.

Ambele funcţii primesc ca intrare eşantionul curent şi nivelul de threshold. La ieşire rutinele întorc eşantionul modificat.

```
function y = soft(x,T)
```

Implementarea rutinelor, mai specific a instrucţiunilor de prelucrare a eşantionului dat, este prezentat în continuare:

- **Hard** – prelucrarea eşantionului are formula

  ```
  y = (abs(x) > T).*x;
  ```

- **Soft** – prelucrarea eşantionului are formula

```
y = max(abs(x) - T, 0);
y = y./(y+T) .* x;
```

  Se observă că şi eşantioanele ce depăşesc nivelul de threshold sunt micşorate în funcţie de acesta când procedura de soft threshold este folosită.

5.1.6 Analiză: aplicarea bancurilor de filtre - afb2D

Aplicarea filtrelor pe linii şi pe coloanele matricei asociate imaginii cu zgomot se face cu ajutor rutinei afb2D.

Rutina primeşte ca parametrii de intrare matricea cu zgomot, filtrele de analiză pentru coloane şi cele pentru rânduri. Atenţie, matricea de intrare de dimensiune NxM trebuie să îndeplinească următoarele condiţii:

- M şi N să fie ambele pare
- $M \geq 2 \times af1l$, unde af1l este lungimea filtrului af1 (de analiză pe coloane)
- $N \geq 2 \times af2l$, unde af2l este lungimea filtrului af2 (de analiză pe linii)

La ieşire rutina întoarce setul de filtre trece jos împreună cu cele trei tipuri de filtre mixte trece sus astfel:

- **lo** - subbanda trece jos
- **hi{1}** - subbanda 'lohi'
- **hi{2}** - subbanda 'hilo'
- **hi{3}** - subbanda 'hihi'

Având în vedere precizările de mai sus definiţia funcţiei este:

```
function [lo, hi] = afb2D(x, af1, af2)
```

Implementarea rutinei începe cu verificarea parametrilor de intrare, astfel încât dacă numărul de parametri este doi, atunci va fi folosit acelaşi filtru atât pentru coloane cât şi pentru linii:

```
if nargin < 3
   af2 = af1;
end
```

Pentru filtrarea pe coloane se foloseşte o rutină suplimentară ce aplică filtrele într-o singură dimensiune. Această rutină se numeşte afb2D_A şi va fi descrisă în continuare.

```
[L, H] = afb2D_A(x, af1, 1);
```

În cazul filtrării pe linii, se va lua în considerare aplicarea filtrelor pentru cele două subbenzi generate anterior, astfel fiind generate cele patru tipuri de subbenzi ce trebuie întoarse la ieşire:

```
[lo,     hi{1}] = afb2D_A(L, af2, 2);
[hi{2}, hi{3}] = afb2D_A(H, af2, 2);
```

5.1.7 Analiză: aplicarea bancurilor de filtre unidimensional - afb2D_A

Precum s-a demonstrat anterior, pentru aplicarea filtrelor de analiză pe coloane şi apoi pe linii, în rutina afb2D a fost apelată rutina afb2D_A.

Această rutină foloseşte o singură dimensiune pentru a procesa semnalul de la intrare şi a-i aplica filtrul dorit.

Parametrii de intrare sunt matricea asociată imaginii, filtrul de analiză şi dimensiunea ce specifică dacă e vorba de coloane sau linii. La fel ca rutina precedentă, matricea de intrare trebuie să îndeplinească un set de condiţii:

- $min(N, M) > 2 \times afl$, unde afl este lungimea filtrului de analiză
- N, M să fie pare

În ceea ce priveşte filtrul de analiză, prima coloană reprezintă filtrul trece jos iar a două filtrul trece sus.

La ieşire rutina returnează subbenzile trece jos şi trece sus. Definiţia rutinei este:

```
function [lo, hi] = afb2D_A(x, af, d)
```

În faza de iniţializare sunt setate filtrele de analiză de tip trece jos şi trece sus. Matricea se transpune în cazul în care rutina este folosită pentru aplicarea filtrelor pe linii, după care se află numărul de linii (sau coloane în caz de transpunere).

```
lpf = af(:, 1);      % lowpass filter
hpf = af(:, 2);      % highpass filter

if d == 2
   x = x';
end
N = size(x,1);
```

Rezultatul obţinut după aplicarea filtrelor de sinteză diferă de semnalul original, în cazul în care filtrele sunt cauzale, printr-o rotaţie circulară. Pentru înlăturarea acestui efect semnalul va fi rotit atât în faza de analiză cât şi în ce de sinteză.

```
L = size(af,1)/2;
x = cshift2D(x,-L);
```

În continuare pentru filtrare şi reducerea numărului de eşantioane se va folosi funcţia Matlab upfirdn (disponibilă în toolbox-ul de Signal Processing).

Dimensiunea de N/2 dorită în subbenzi este depăşită cu L/2, unde L reprezintă lungimea filtrelor. Pentru a elimina excesul, ultimele L/2 eşantioane se adaugă primelor L/2. Aceasta reprezintă însă o soluţie de convenienţă deoarece operaţia efectuată poate ridica anumite probleme la începutul şi sfârşitul semnalului din subbenzi.

```
lo = upfirdn(x, lpf, 1, 2);
lo(1:L, :) = lo(1:L, :) + lo([1:L]+N/2, :);
lo = lo(1:N/2, :);

hi = upfirdn(x, hpf, 1, 2);
hi(1:L, :) = hi(1:L, :) + hi([1:L]+N/2, :);
hi = hi(1:N/2, :);
```

Odată ce se obţin subbenzile trece-sus şi trece-jos, trebuie stabilit tipul operaţiei (pe coloane sau pe linii) şi în funcţie de asta transpunerea filtrelor.

```
if d == 2
    lo = lo';
    hi = hi';
end
```

5.1.8 Sinteză: aplicarea bancurilor de filtre - sfb2D

Pentru obţinerea transformatei inverse wavelet este necesară aplicarea filtrelor de sinteză pentru fiecare pereche de orientare wavelet. Rutina folosită de icplxdual2D se numestet sfb2D şi este strâns legată de rutina de analiză afb2D.

Această rutină primeşte că parametri de intrare subbenzile trece jos şi trece sus cât şi filtrele de sinteză pentru linii şi coloane. La ieşire rutina întoarce semnalul iniţial obţinut la etapa corespunzătoare. Definiţia rutinei este:

```
function y = sfb2D(lo, hi, sf1, sf2)
```

În cazul în care funcţia a fost apelată cu trei parametri, se vor folosi aceleaşi filtre de sinteză atât pentru coloane cât şi pentru linii.

```
if nargin < 4
    sf2 = sf1;
end
```

Pe acelaşi principiu va fi construit şi semnalul iniţial prin aplicarea filtrelor de sinteză în ordine inversă faţă de cele de analiză.

Întâi pe linii:

```
lo = sfb2D_A(lo,     hi{1}, sf2, 2);
hi = sfb2D_A(hi{2}, hi{3}, sf2, 2);
```

Apoi pe coloane:

```
y = sfb2D_A(lo, hi, sf1, 1);
```

Se observă că rutina apelează la o altă rutină sfb2D_A. La sfârşitul acestei operaţii semnalul obţinut este întors la ieşire.

5.1.9 Sinteză: aplicarea bancurilor de filtre unidimensional - sfb2D_A

După cum s-a observat anterior rutina de sinteză apelează la o rutină sfb2D_A pentru a apleca filtrele pe linii şi pe coloane. Această rutină este similară cu cea folosită de rutina de analiză afb2D_A.

Aceasta rutină primeşte la intrare subbenzile trece jos şi trece sus, filtrele de analiză şi dimensiunea (ce specifică dacă filtrele se aplică pe linii sau pe coloane).

Rutina întoarce la ieşire subbanda rezultată din sinteza celor două primite ca parametrii cu ajutorul filtrelor respective. Definiţia rutinei este:

```
function y = sfb2D_A(lo, hi, sf, d)
```

În faza de iniţializare se preiau filtrele de analiză trece sus şi trece jos, se transpun subbenzile în caz de analiză pe linii şi se extrage numărul de coloane (sau linii în caz de transpunere) rezultante:

```
lpf = sf(:, 1);      % lowpass filter
hpf = sf(:, 2);      % highpass filter

if d == 2
    lo = lo';
    hi = hi';
end

N = 2*size(lo,1);
```

Similar cu rutina de analiză se foloseşte funcţia Matlab pentru filtrare şi reconstrucţie a eşantioanelor:

```
y = upfirdn(lo, lpf, 2, 1) + upfirdn(hi, hpf, 2, 1);
```

După care se efectuează operaţiile de adunare şi shift-are circulară în ordine inversă. Motivul acestor operaţii a fost discutată în cazul rutinei de analiză:

```
y = upfirdn(lo, lpf, 2, 1) + upfirdn(hi, hpf, 2, 1);
y(1:L-2, :) = y(1:L-2, :) + y(N+[1:L-2], :);
y = y(1:N, :);
y = cshift2D(y, 1-L/2);
```

La final se verifică dacă s-a operat pe linii sau pe coloane şi se transpune semnalul rezultat dacă este necesar, după care se întoarce la ieşire semnalul obţinut.

```
if d == 2
   y = y';
end
```

5.2 Aplicaţia Matlab — arborele dual cu DD

Extinderea aplicaţiei Matlab pentru a cuprinde şi implementarea arborelui dual aduce noi parametri în funcţiile de nivel înalt. Principalele funcţii afectate sunt it imginit şi it mydenoise. Aceste funcţii capătă noi parametrii şi apeluri la rutine specifice arborelui dual cu DD.

Cu toate acestea există multe similarităţi între modul de implementare al arborelui dual simplu şi cel cu DD, astfel explicaţiile necesare acestei extinderi devin mult mai uşor de expus.

5.2.1 Iniţializarea datelor – imginit

Rutină de iniţializare a datelor primeşte un nou flag ca parametru: *dd*. Acesta stabileşte dacă apelul funcţiei este făcut pentru arborele simplu sau pentru arborele cu DD şi iniţializează filtrele necesare pentru procesarea imaginii.

Iniţializarea filtrelor este aproape identică cu cea pentru cazul arborelui simplu, singura diferenţă fiind folosirea filtrelor pentru prima etapă care nu se mai păstrează.

```
if dd == 0
...
```

```
else
        [Faf, Fsf] = FSdoubledualfilt;
        if flags == 0
                [af, sf] = doubledualfilt;
        elseif == 1
                ...
        end
end
```

5.2.2 Rutina principală – mydenoise

Odată cu extinderea aplicaţie pentru arborele cu DD, apar schimbări şi în rutina principală. Din nou, similarităţile între arborele simplu şi cel cu DD fac ca această extindere să fie destul de uşor integrat.

În primul rând *mydenoise* capătă un nou parametru **dd**. În funcţie de acesta rutina foloseşte arborele cu DD sau nu. În cazul în care variabila este setată, rutina aplică transformata wavelet cu arbore dual cu DD direct după care parcurge fiecare subbandă pentru a filtra eşantioanele ce depăşesc nivelul de threshold impus.

Numărul de subbenzi a crescut odată cu extinderea arborelui dual, dar principiile după care se analizează imaginea rămân aceleaşi. Odată terminat procesul de analiză imaginea trebuie reconstruită. Pentru aceasta se apelează la rutina transformatei wavelet cu arbore cu DD inversă ce întoarce imaginea prelucrată.

```
if dd == 0
...
else
  w = cplxdoubledual_f2D(x,J,Faf,af);
  % loop thru scales
  for j = 1:J
    % loop thru subbands
    for s1 = 1:2
      for s2 = 1:8
        C = w{j}{1}{s1}{s2} + I*w{j}{2}{s1}{s2};
        C = soft(C,T);
        w{j}{1}{s1}{s2} = real(C);
        w{j}{2}{s1}{s2} = imag(C);
      end
    end
  end
```

```
  y = cplxdoubledual_i2D(w,J,Fsf,sf);
end
```

5.2.3 Aplicarea filtrelor

Rutinele de aplicare a filtrelor de sinteză şi analiză pentru arborele dual cu DD sunt asemănătoare celor explicate pentru arborele dual simplu. Filtrele sunt aplicate întâi pe coloane iar apoi pe linii, diferenţa majoră fiind numărul mare de subbenzi şi tipul de frecvenţe asociate acestora.

În principiu rutinele întorc aceleiaşi parametrii, de frecvenţe înalte şi frecvenţe joase, diferenţă fiind că numărul elementelor din tabela de frecvenţe înalte a crescut semnificativ.

```
% filter along columns
[L, H1, H2] = afb3_2D_A(x, af1, 1);

% filter along rows
[lo,    hi{1}, hi{2}] = afb3_2D_A(L, af2, 2);
[hi{3}, hi{4}, hi{5}] = afb3_2D_A(H1, af2, 2);
[hi{6}, hi{7}, hi{8}] = afb3_2D_A(H2, af2, 2);
```

5.3 Calcularea erorii

5.3.1 Proprietatea de reconstrucţie a transformatei wavelet

O imagine este prelucrata pentru eliminarea zgomotului cu ajutorul transformatei wavelet directe. Această transformare a fost implementată cu ajutorul Matlab într-o rutină prezentată mai sus (cplxdual2D).

După prelucrare este necesară extragerea imaginii. Pentru aceasta se foloseşte transformata wavelet inversă. Dea semeni există o rutină Matlab icplxdual2D a cărei descriere poate fi găsită în secţiunea anterioară.

Pentru a verifica proprietatea de reconstrucţie a transformatei wavelet se poate folosi o secvenţă de instrucţiuni similară cu următoarea:

```
>> x = rand(256, 128);
>> J = 4;
>> [Faf, Fsf] = FSfarras;
>> [af, sf] = dualfilt1;
>> w = cplxdual2D(x, J, Faf, af);
>> y = icplxdual2D(w, J, Fsf, sf);
>> err = x - y;
>> max(max(abs(err)))
```

```
ans =

    3.8693e-008
```

Se observă că semnalul original a fost reconstruit integral.

5.3.2 Proprietatea de reconstrucţie a bancurilor de filtre

Procesul de aplicare a bancurilor de filtre de analiză asupra semnalului iniţial a fost implementat în Matlab printr-o rutină numită afb2D. Această rutină este descrisă în secţiunea anterioară.

După analiza semnalului şi procesarea sa, se vor aplica bancurile de filtre de analiză corespunzătoare. Metoda a fost implementată în rutină sfb2D şi este de asemeni descrisă anterior.

Cele două procesări trebuie să îndeplinească proprietatea de reconstrucţie. Pentru a verifică această proprietate următoarele instrucţiuni pot fi introduse în consola Matlab:

```
>> x = rand(256, 128);
>> [af, sf] = dualfilt1;
>> [lo, hi] = afb2D(x, af{1}, af{2});
>> y = sfb2D(lo, hi, sf{1}, sf{2});
>> err = x - y;
>> max(max(abs(err)))

ans =

    1.8974e-008
```

Se observă reconstrucţia perfectă a semnalului x.

5.3.3 PSNR

Peak Signal-to-Noise Ratio (PSNR) indică raportul dintre puterea maximă a unui semnal şi puterea zgomotului ce perturbă fidelitatea reprezentării lui.

Raportul este cel mai uşor definit cu ajutorul MSE (mean square error) care pentru două imagini monocrome de mxn I şi K, unde o imagine este considerată reprezentarea cu zgomot a celeilalte, este definit astfel:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ||I(i,j) - K(i,j)||^2$$

Cu ajutorul MSE, PSNR-ul este definit ca:

$$PSNR = 10 \lg \frac{MAX_i^2}{MSE}$$

În următoarele comparaţii între filtre şi cele între tipuri de threshold se va folosi drept comparaţie PSNR-ul, fiind implementată în acest scop şi o rutină Matlab cu acelaşi nume.

Prin intermediul PSNR şi a unei rutine Matlab cu aceeaşi denumire, se vor realiză în continuare comparaţii atât între filtre cât şi între tipurile de treshold.

Pentru a observa diferenţele între cele două tipuri de threshold (soft şi hard) sau diferenţele între tipuri de filtre diferite se vor folosi secvenţe de instrucţiuni similare cu:

```
>> [xo, xn, filter] = imginit('barbara.png', 0);
>> xn = mydenoise(xn, 20, 4, filter, 0);
>> err = psnr(xo, xn)

err =

    27.6475

>> [xo, xn, filter] = imginit('barbara.png', 0);
>> xn = mydenoise(xn, 20, 4, filter, 1);
>> err = psnr(xo, xn)

err =

    25.9154
```

Pentru comparaţii threshold se foloseşte aceeaşi imagine cu acelaşi set de filtre, singura diferenţă între apeluri fiind tipul de threshold ales.

Bancurile de filtre vor fi comparate folosind aceeaşi imagine, cu acelaşi tip de threshold aplicând iterativ filtrele disponibile şi calculându-se PSNR-ul pentru fiecare pas.

Folosind rutinele Matlab necesare procesării imaginii (atât pentru arborele dual simplu cât şi pentru arborele dual DD) şi un set dat de bancuri de filtre proiectat după mai multe criterii de performanţă, se pot realiza mai multe comparaţii şi experimente. Acestea vor fi prezentate în detaliu în secţiunea următoare.

6 Rezultate experimentale

Filtrele folosite pentru eliminarea zgomotului cu ajutorul transformatei arborelui dual simplu sunt preluate din tratate de specialitate prezentând caracteristici relativ diferite, concentrându-se asupra anumitor aspecte din cadrul celor prezentate în secţiunea 3.

S-au folosit filtre prezentate în lucrarea "Hilbert Transform Pairs of Wavelet Bases" [4] de Ivan Selesnick. Această lucrare prezintă proiectarea filtrelor ce aproximează bine întârzierea cu jumătate de eşantion a filtrelor de scalare. În lucrarea curentă filtrele au fost notate cu "htpwb".

A doua serie de filtre urmează tiparul folosit în lucrarea "The Design of Approximate Hilbert Transform Pairs of Wavelet Bases" [5] scrisă de Ivan Selesnick. Filtrele sunt construite cu ajutorul factorizării şi au fost notate cu "dahtpwp".

Un alt set de filtre a fost preluat din lucrarea "Optimization of Symmetric Self-Hilbertian Filters for the Dual-Tree Complex Wavelet Transform"[1] scrisă de Bogdan Dumitrescu et al., în care filtrele sunt simetrice şi pot avea unul sau două grade de libertate. Aceste filtre au fost prescurtate în lucrare cu "bd".

Pentru transformata wavelet cu arbore dual cu densitate dublă s-au folosit două tipuri de filtre. Primul tip este propus tot de Ivan Selesnick, în lucrarea "The Double-Density Dual-Tree DWT"[6], şi satisface restrângerile descrise în secţiunea 4 rezultatele prezentând un grad ridicat de netezime în urma prelucrării.

O a doua serie de filtre este preluată din lucrarea "A Method For Designing The Double-Density Dual-Tree Discrete Wavelet Transform"[2], scrisă de Bogdan Dumitrescu şi Ali Bahrami Rad, unde filtrele pentru arborele dual cu DD prezintă o analiticitate crescută faţă de cele prezentate de Selesnick.

În continuare vor fi prezentate rezultatele experimentale obţinute prin aplicarea acestor filtre asupra imaginilor cu zgomot.

6.1 Rezultate numerice

6.1.1 Rezultatele obţinute cu threshold-uri diferite pentru arborele dual simplu

Pentru a putea observa diferenţele dintre cele două tipuri de thresholding, s-au aplicat toate seturile de filtre, disponibile pentru arborele dual simplu, asupra unei singure imagini. Rezultatele din tabela (1) au dus la concluzia că în cazul arborelui dual simplu threshold-ul soft prezintă rezultate mult mai bune fata de cel hard (aproximativ 3dB diferenţă).

Tabela 1: DT: Diferenţe între tipurile de threshold ($N = 4 : 22$, house)

Filtru	Threshold(PSNR)	
	Soft	Hard
dualfilt1	29.8792	26.4951
htpwb_N10K4L5	29.8546	26.5565
htpwb_N10K3L7	29.9467	26.4479
dahtpwb_N12K4L2	29.9629	26.4723
dahtpwb_N12K3L3	29.9206	26.4936
bd1_4	29.4002	26.3873
bd1_6	29.7108	26.3278
bd1_8	29.5301	26.3235
bd1_10	29.4609	26.4212
bd1_12	29.5985	26.2145
bd1_14	29.3915	26.3336
bd1_16	29.3071	26.4673
bd1_18	29.4058	26.3770
bd1_20	29.5160	26.2439
bd1_22	29.1659	26.3403
bd2_6	29.7535	26.2829
bd2_8	29.4681	26.3335
bd2_10	29.2653	26.3911
bd2_12	29.4896	26.2842
bd2_14	29.5735	26.3310
bd2_16	29.4177	26.3601
bd2_18	29.2089	26.3657
bd2_20	29.5196	26.3224
bd2_22	29.1265	26.3338

6.1.2 Rezultatele obţinute cu bancurile de filtre pentru arborele dual simplu

Pentru a putea compara diferitele moduri de proiectare a bancurilor de filtre, acestea din urma au fost aplicate asupra unei singure imagini. În toate cazurile s-a folosit soft threshold în urma rezultatelor din secţiunea anterioară. Filtrele au fost aranjate pe categorii în funcţie de lungimea lor. Conform rezultatelor numerice (tabelele 2 şi 3), filtrele propuse în [4] şi [5] au avut rezultate mai bune decât cele din [1] cu aproximativ 0,5 dB.

Tabela 2: DT: Filtre de lungime 10 ($N = 10$, barbara)

Filtru	PSNR
htpwb_N10K4L5	27.6518
htpwb_N10K3L7	27.6686
bd1_10	27.2796
bd2_10	27.0810
dualfilt1	27.6149

Tabela 3: DT: Filtre de lungime 12 ($N = 12$, flinstones)

Filtru	PSNR
dahtpwb_N12K4L2	26.3250
dahtpwb_N12K3L3	26.3393
bd1_12	25.9049
bd2_12	25.8117

Similar s-au comparat filtrele propuse în [1] în funcţie de gradele de libertate şi lungime. Rezultatele (vezi tabela 4) indică o performanţă mai buna a celor cu două grade de libertate, cu o diferenţa de aproximativ 0,3 dB faţă de cele cu un singur grad de libertate.

Tabela 4: DT: Filtre cu unul şi două grade de libertate ($N = 4 : 22$, boat)

Lungime(N)	Grade de libertate(PSNR)	
	1	2
4	28.1800	–
6	28.2806	28.3021
8	28.2568	28.2891
10	28.2758	28.0110
12	28.2735	28.1998
14	28.2247	28.2339
16	28.0461	28.2413
18	28.2371	28.0024
20	28.1961	28.2624
22	27.9778	28.0313

6.1.3 Rezultatele obţinute pentru arborele dual cu DD

Rezultatele (vezi tabela 5) obţinute pentru tipurile de thresholding, aplicând aceiaşi metodă ca şi în cazul arborelui dual simplu, indică o diferenţă semnificativă între cele două, dar de data asta thresholding-ul hard prezintă un PSNR mai bun cu aproximativ 3dB.

Tabela 5: DD: Diferente între tipurile de threshold ($N = 10, 16$, house)

Filtru	Threshold(PSNR)	
	Soft	Hard
selesnick1	22.2610	26.6941
selesnick2	22.9719	26.6359
selesnick3	23.1216	26.5133
dd_bd1	23.1338	26.5107
dd_bd2	23.0435	26.5850

Un al doilea set de teste (vezi tabelele 6 şi 7) a fost efectuat asupra bancurilor de filtre de aceiaşi lungime, similar cu testele pentru arborele dual simplu. Diferenţele în acest caz au fost foarte mici între modurile de proiectare propuse în [6] şi [2] (maxim 0,1 dB).

Tabela 6: DD: Filtre de lungime 10 ($N = 10$, flinstones)

Filtru	PSNR
selesnick1	23.2232
selesnick2	23.1037
dd_bd2	23.0946

Tabela 7: DD: Filtre de lungime 16 ($N = 16$, flinstones)

Filtru	PSNR
dd_bd1	23.0144
selesnick3	23.0104

6.2 Rezultate grafice

Pentru a exemplifica şi confirma grafic rezultatele numerice, în continuare vor fi prezentate diferite imagini şi modul cum au fost prelucrate. În fiecare caz se va porni de la original, apoi se va prezenta poza cu zgomot, iar la final rezultatele prelucrării.

6.2.1 Tipuri de threshold

- **Barbara, DT, dualfilt1**

  Pentru a evidenţia diferenţa între tipul de threshold soft şi cel hard la arborele dual simplu, s-a ales o imagine de test (vezi figura 12 asupra căreia s-a aplicat zgomot (vezi figura 13).

Figura 12: DT rezultate threshold: Imaginea originală

Folosind filtrul dualfilt1 propus în [4] şi optând pentru hard threshold, se obţine o imagine (vezi figura 14) un pic mai clară. Totuşi este încă vizibil o mare parte din zgomotul iniţial. Diferenţa în cazul alegerii soft thresholding-ului este uşor de perceput cu ochiul liber(vezi figura 15). Imaginea este mult mai clară, zgomotul iniţial fiind eliminat aproape complet.

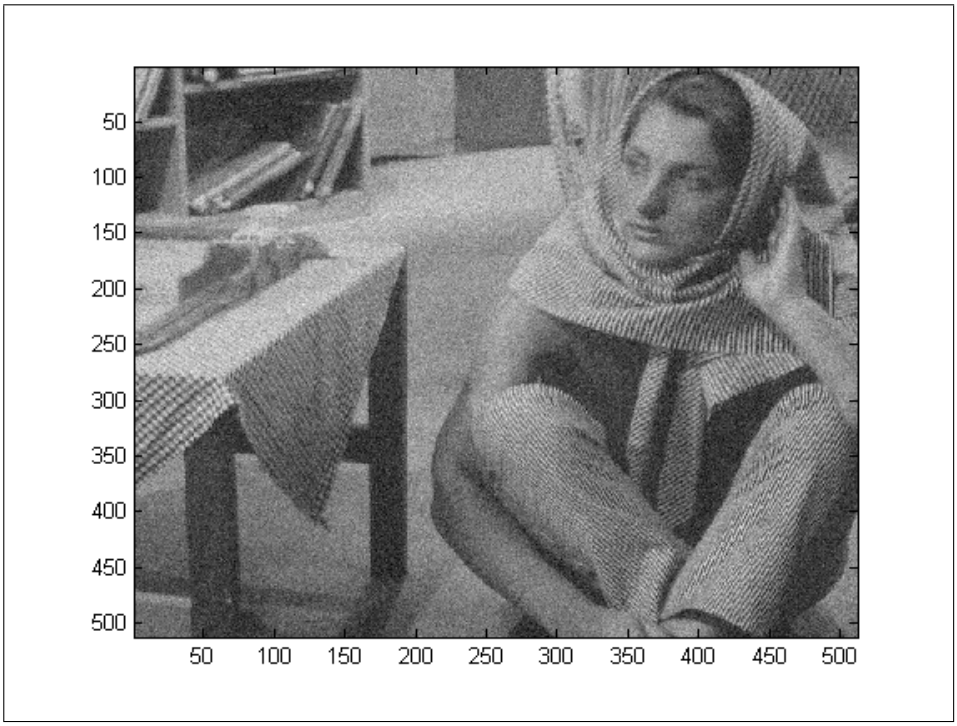

Figura 13: DT rezultate threshold: Imaginea cu zgomot

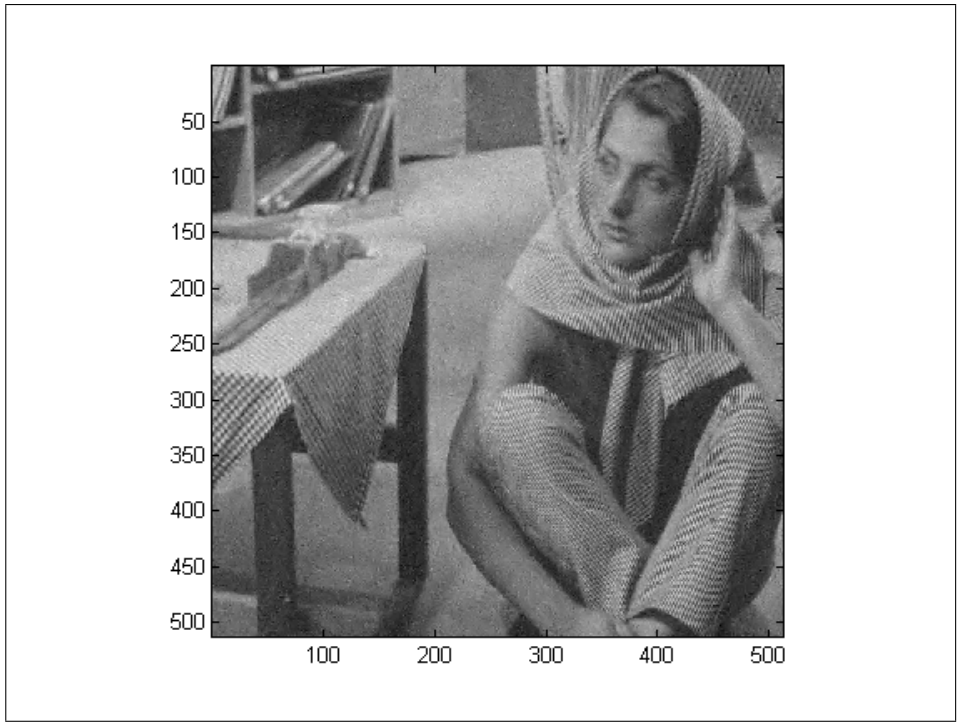

Figura 14: DT rezultate threshold: Hard Thresholding

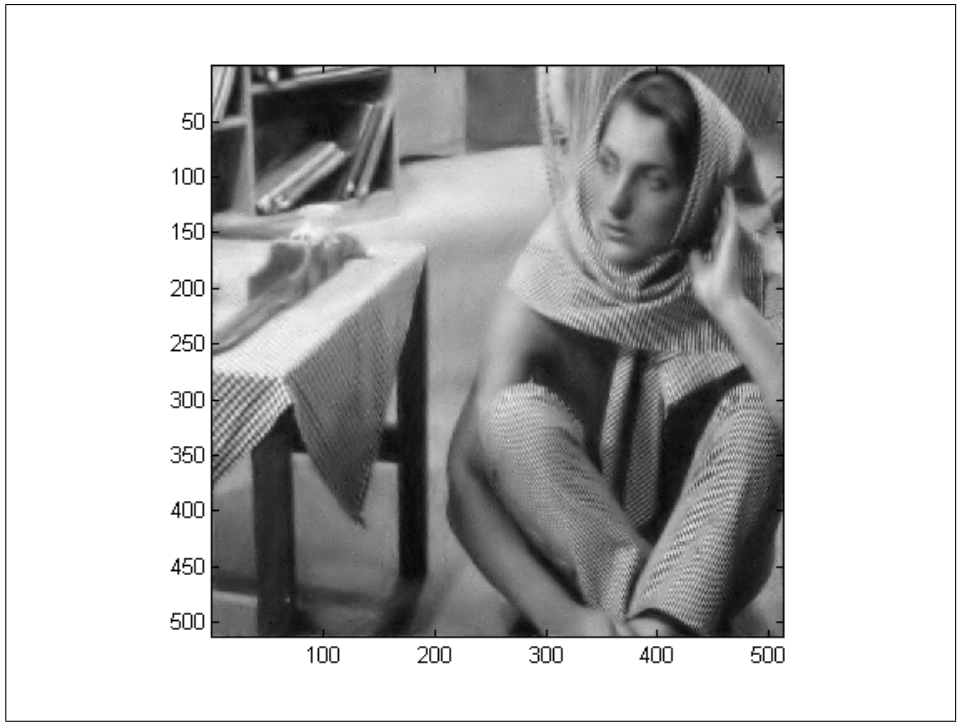

Figura 15: DT rezultate threshold: Soft Thresholding

- **Lena, DD, dd_bd_2**

În cazul arborelui dual cu DD s-a ales o alta imagine(figura 16), scopul fiind tot de a compara cele două tipuri de threshold. Se aplică din nou zgomot asupra originalului şi astfel se obţine o imagine perturbată (figura 17).

Conform rezultatelor numerice (vezi tabela 5), aplicarea threshold-ului hard îmbunătăţeşte semnificativ imaginea, rezultatul (vezi figura 18 fiind comparabil cu cel obţinut în cazul soft-thresholding-ului de la arborele dual simplu. Imaginea obţinută prin soft thresholding este şi ea clară (vezi figura 19, dar totuşi mult prea mată, muchiile fiind pierdute în urma prelucrării.

Figura 16: DD rezultate threshold: Imaginea originală

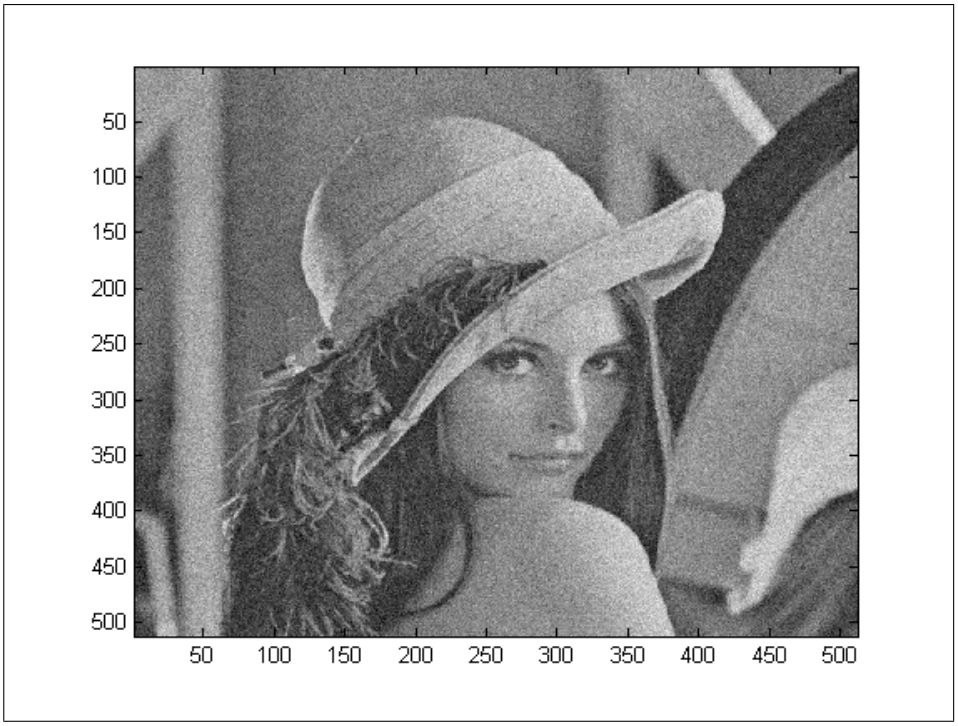

Figura 17: DD rezultate threshold: Imaginea cu zgomot

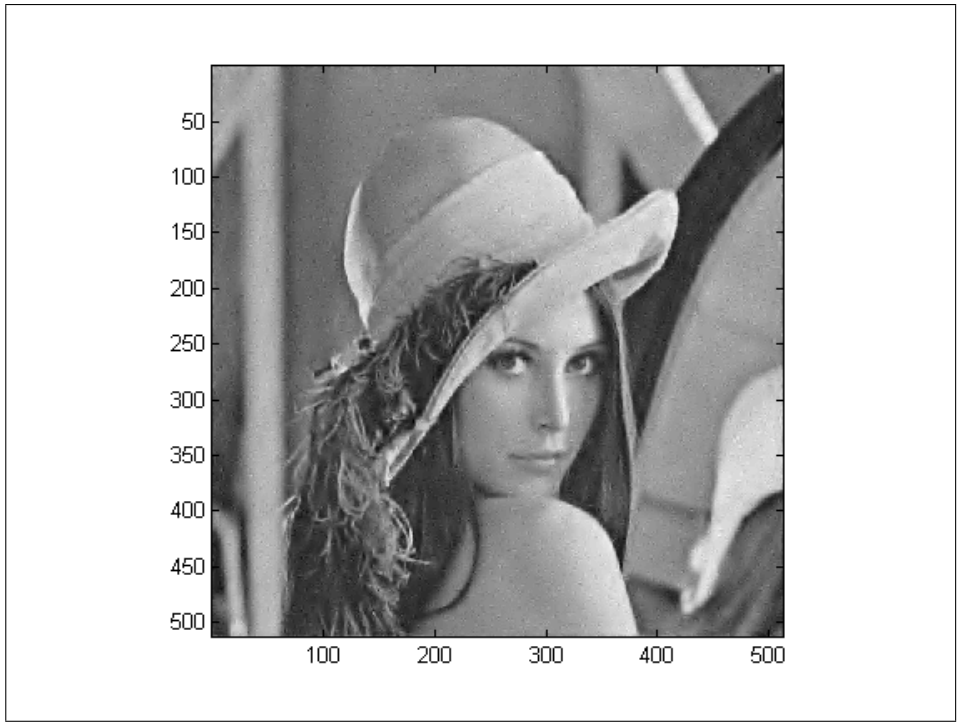

Figura 18: DD rezultate threshold: Hard Thresholding

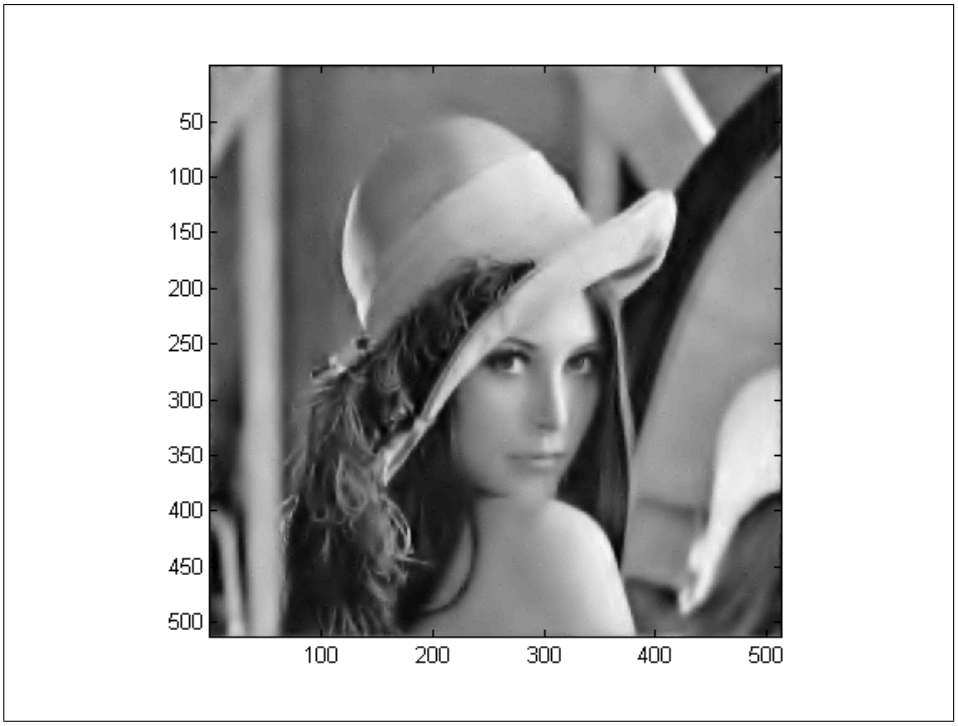

Figura 19: DD rezultate threshold: Soft Thresholding

6.2.2 Bancuri de filtre — arbore dual

În cazul arborelui dual simplu se compară rezultatele obţinute cu cele două tipuri de filtre (propuse în [4] [5] şi, respectiv, [1]). Pentru primul set de filtre, propuse de Selesnick, se alege o imagine clară (20) asupra căreia se aplică zgomot (vezi figura 21). Aplicând filtrele din [4] cu soft thresholding se obţine o imagine fără zgomot (figura 22).

- **Peppers, dahtpwb_N12K3L3**

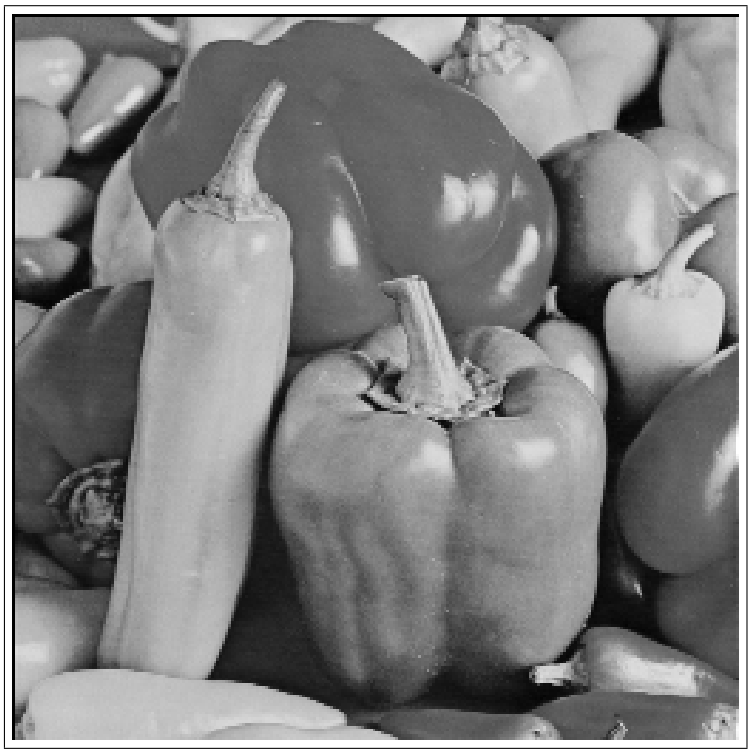

Figura 20: DT rezultate filtre: Peppers original

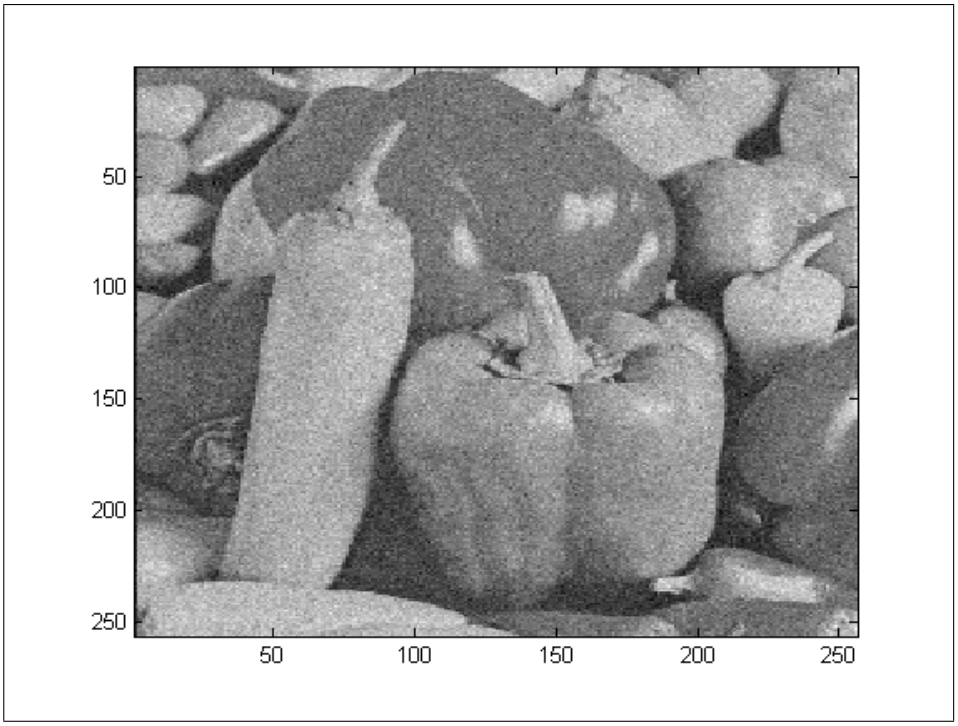

Figura 21: DT rezultate filtre: Peppers cu zgomot

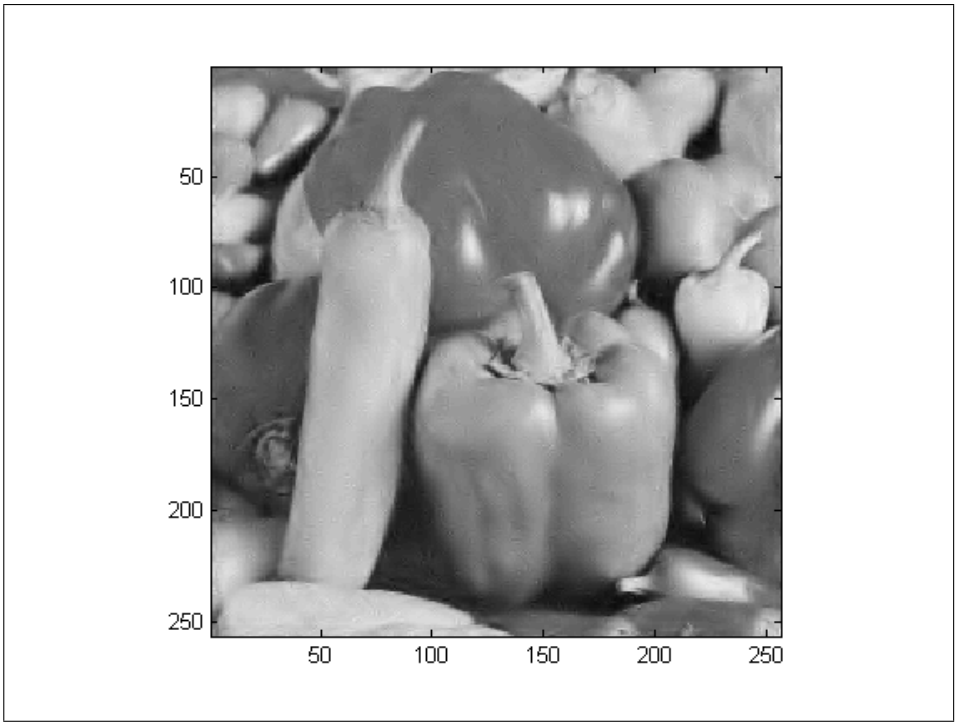

Figura 22: DT rezultate filtre: Peppers prelucrat

- **Flinstones, bd2_12**

Pentru a compara cu filtrele propuse în [1] se alege din nou o imagine (figura 23) asupra căreia se aplică zgomot (figura 24). Astfel se obţine, cu threshold soft şi filtre proiectate conform [1], o imagine fără zgomot (vezi figura 25). Comparând cele două imagini prelucrate (22 şi 25), se observă că rezultatul grafic coincide cu cel numeric (vezi tabelele 2 şi 3).

Figura 23: DT rezultate filtre: Flinstones original

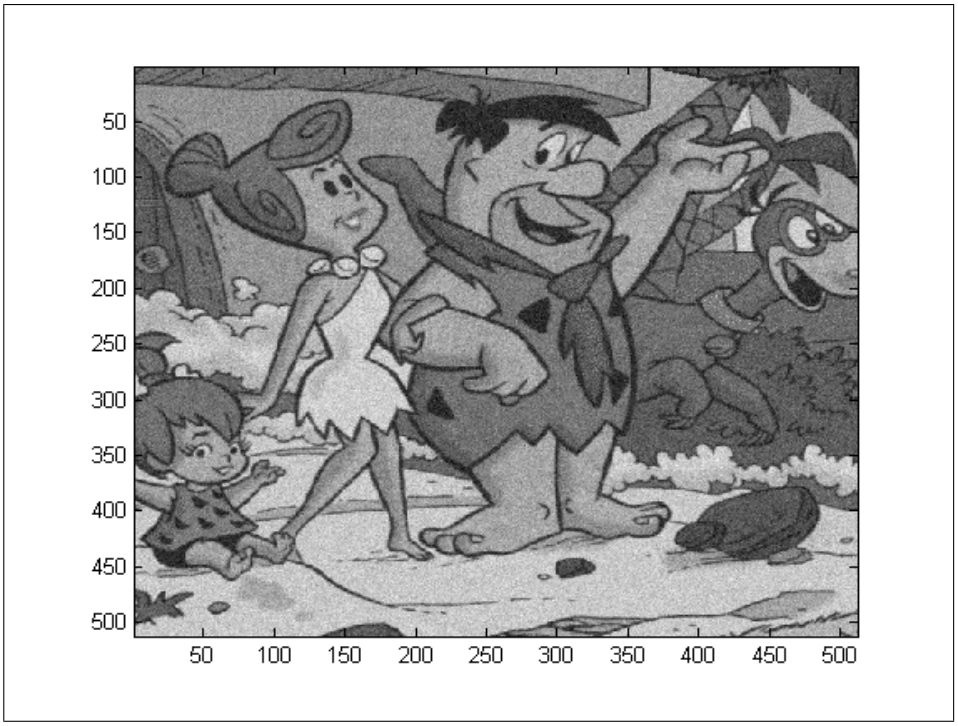

Figura 24: DT rezultate filtre: Flinstones cu zgomot

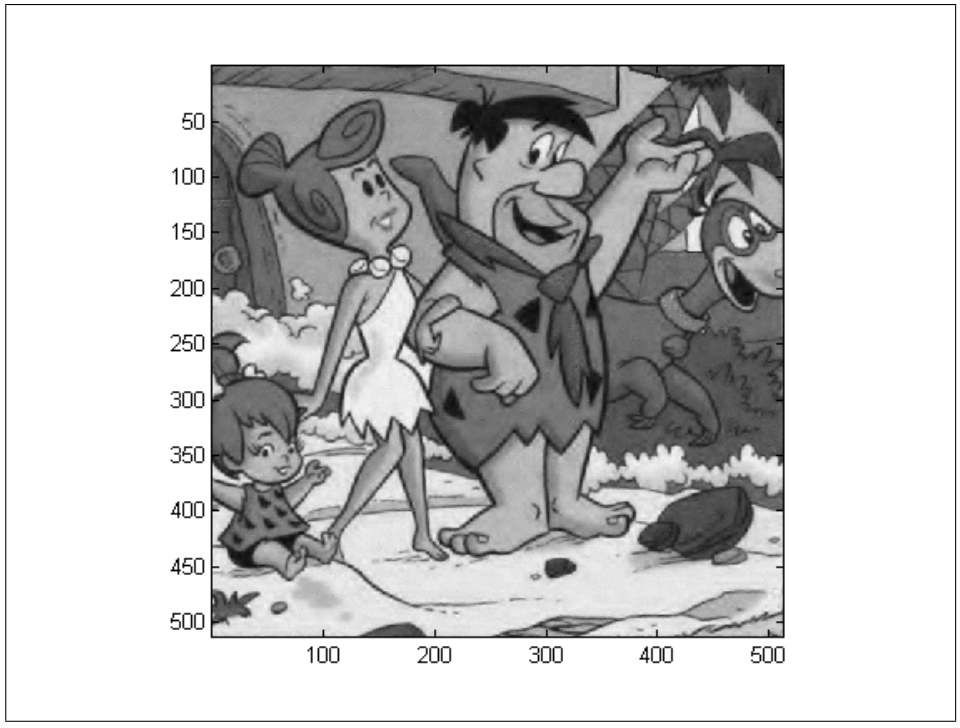

Figura 25: DT rezultate filtre: Flinstones prelucrat

6.2.3 Bancuri de filtre — arbore dual cu DD

- **House, doubledualfilt**

În cazul arborelui dual DD, având în vedere diferenţa foarte mica între cele două tipuri de proiectări (vezi tabelele 6 şi 7), s-a ales exemplificarea numai cu un singur tip de filtre. Astfel, se alege o imagine fără zgomot (figura 26), căreia se aplică o perturbaţie (vezi figura 27). Folosind hard thresholding, fiind conform rezultatelor numerice mai bun decât cel soft, se aplică filtrele şi se prelucrează imaginea obţinând o imagine mult mai clară decât cea cu zgomot (figura 28).

Figura 26: DD rezultate filtre: House original

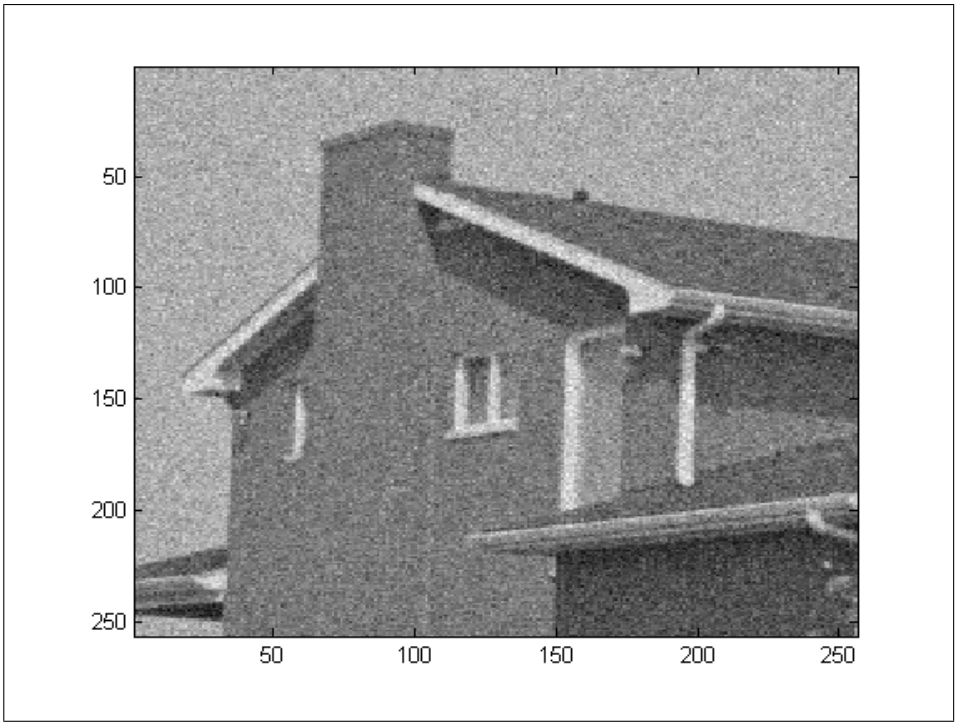

Figura 27: DD rezultate filtre: House cu zgomot

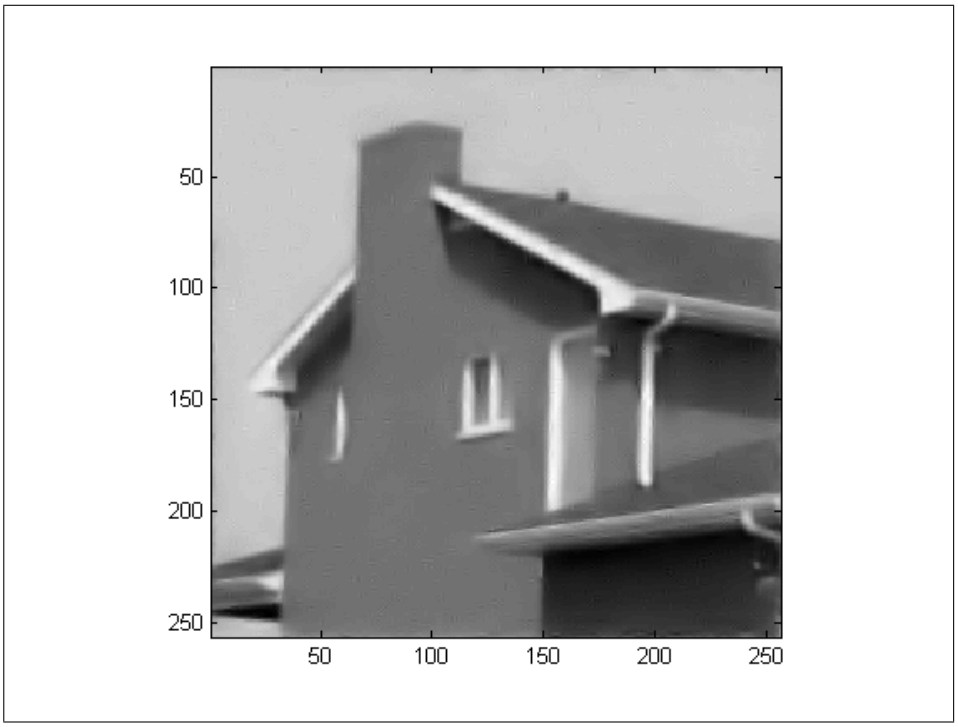

Figura 28: DD rezultate filtre: House prelucrat

7 Concluzii

In urma rezultatelor experimentale grafice şi, în special, a celor numerice, se pot trage o serie de concluzii legate de efectul diferitelor transformări wavelet asupra prelucrării imaginilor. Diferenţele constau atât în diferitele tipuri de transformări wavelet cât şi în diferenţele între proiectarea filtrelor, a lungimii acestora şi a utilizării tipurilor de threshold.

Privind în ansamblu, se observă că prelucrarea imaginilor cu arborele dual cu DD este mai puţin eficientă decât cea cu arborele dual simplu. PSNR-ul indică o diferenţa de până la 5dB în ciuda proprietarilor şi a proiectării mai ample a arborelui dual cu DD.

Ivan Selesnick prezintă ca principală cauză a acestei discrepanţe metoda de aplicare a threshold-ului în arborele dual cu DD. Pe scurt, fiecare subbandă ar trebui sa-şi calculeze un threshold specific ce ar înlătura forma extrem de netedă a imaginii prelucrate şi ar duce la valori mult mai ridicate a PSNR-ului.

Analizând diferenţele apărute în urma aplicării filtrelor de acelaşi tip, dar de lungime diferită, se observă că lungimea filtrului nu afectează proporţional PSNR-ul, rezultatele pentru un filtru mai lung putând fi mai slabe decât cele pentru un filtru mai scurt şi vice-versa.

Diferenţele între tipurile de implementare ale filtrelor nu sunt foarte mari, PSNR-ul putând oscila, în general, de la 0,5dB până la maximum 1dB.

Mult mai pregnantă este diferenţa între cele două tipuri de threshold. În cazul arborelui dual simplu PSNR-ul pentru threshold-ul soft are în medie cu peste 3dB fata de threshold-ul hard. Rezultatele numerice se pot observa cu uşurinţa şi în exemplele grafice, imaginea prelucrată cu soft threshold fiind mult mai netedă şi mai clară decât imaginea prelucrată cu hard threshold ce încă prezintă "purici".

Pentru arborele dual cu DD rolurile se inversează, threshold-ul hard are un PSNR mai bun faţă de cel soft la o diferenţă similara de 3dB. După cum se observă din rezultatele grafice, imaginile create cu soft threshold sunt foarte netede, aproape lipsite de muchii, pe când cele cu hard threshold sunt mai clare, mai rigide, menţinând însă vizibil "puricii".

Prelucrarea imaginilor cu ajutorul transformării wavelet reprezintă o soluţie viabilă. Privind în comparaţie cu PSNR-ul obţinut între imaginile comprimate şi cele originale, care se afla în intervalul 30-40dB, rezultatele obţinute în lucrarea curentă se apropie de pragul de 30dB.

Eliminarea zgomotului cu ajutorul transformării wavelet este aplicată în mai multe domenii ca de exemplu în fotografie, aparate TV, software (aplicaţii video şi audio), medicină (mai ales pentru cazurile 3D).

Bibliografie

[1] B. Dumitrescu, I. Bayram, and I.W. Selesnick. Optimization of symmetric self-hilbertian filters for the dual-tree complex wavelet transform. *IEEE Signal Processing Lett.*, 15:146–149, 2008.

[2] B. Dumitrescu and A.B. Rad. A method for designing the double-density dual-tree discrete wavelet transform. *LNLA*, Lausanne, 2008.

[3] S. Mallat. *A Wavelet Tour of Signal Processing*. New York: Academic, 1998.

[4] I.W. Selesnick. Hilbert transform pairs of wavelet bases. *IEEE Signal Processing Lett.*, 8(6):170–173, 2001.

[5] I.W. Selesnick. The design of approximate hilbert transform pairs of wavelet bases. *IEEE Trans. Signal Processing*, 50(5):1144–1152, 2002.

[6] I.W. Selesnick. The double-density dual-tree discrete wavelet transform. *IEEE Trans. Signal Processing*, 52(5):1304–1314, 2004.

[7] I.W. Selesnick, R.G. Baraniuk, and N.G. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, pages 123–151, 2005.