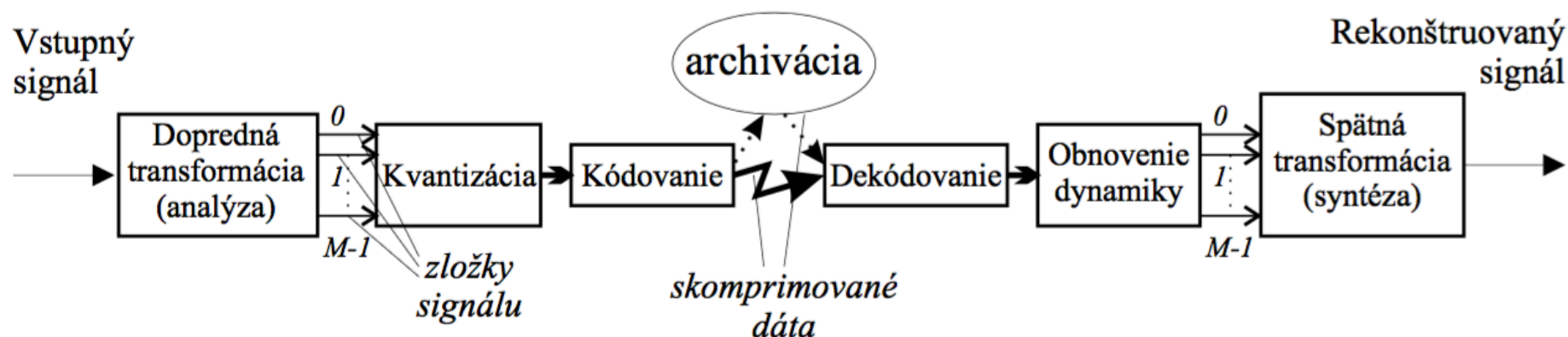


Interpolácia a banky filtrov

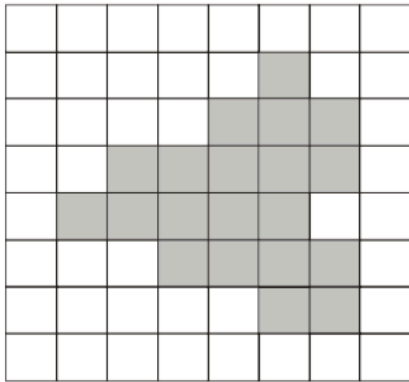
- Klasická interpolácia – ako to funguje?
 - Pomocou polynómov
 - Pomocou Splajnov (spline)
 - Polýnomy na intervaloch (piecewise polynomials)
 - Okrem prepojenia dvoch bodov musí sedieť aj derivácia v koncových bodoch
 - Kvadratické spajny, kubické splajny, ...
 - Konštrukcia splajnov – aké sú to funkcie?
- Interpolácia pri rekonštrukcii v 2 pásmových bankách filtrov
 - DP Filter $H(z)$ nazývame **interpolačný**, ak je schopný interpolovať koeficienty prestriedané nulami (najradšej ako polynómy R-tého rádu).
 - Príklady Interpolačných filtrov:
 - Haarov wavelet – konštantná interpolácia $h(n)=\{1,1\}$
 - CDF 2,2 – lineárna intarpolácia, $h(n)=\{1/2, 1, 1/2\}$,
 - 4-bodová schéma $h(n) = \{-1/16, 0, 9/16, 1, 9/16, 0, -1/16\}$
- V 2D prípade bank filtrov sa dajú zostrojiť separovateľné aj neseparovateľné prípady

Kódovanie obrazu všeobecná schéma

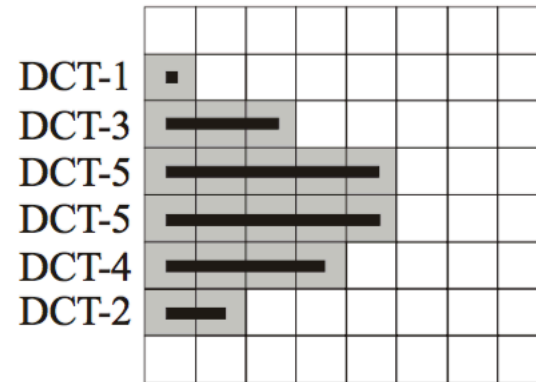


Kódovanie objektov

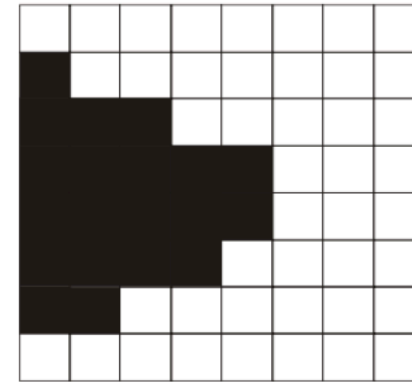
- Klasicky o objekte hovoríme ak máme jasnú hranicu objektu
- Kódovanie pomocou DWT
 - Pamatáme si hranicu
 - Pamatáme si iba tie koeficienty, ktorých odpovedá bazová funkcia, ktorá sa prekrýva s objektom
- Kódovanie pomocou DCT
 - napr. shape adaptive DCT



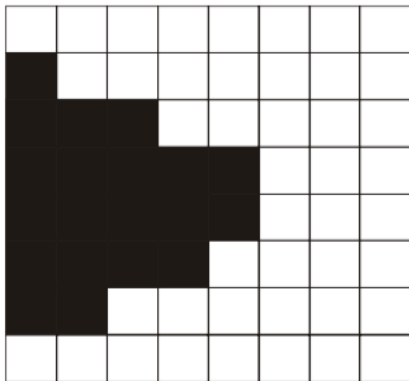
(a) Originálny segment



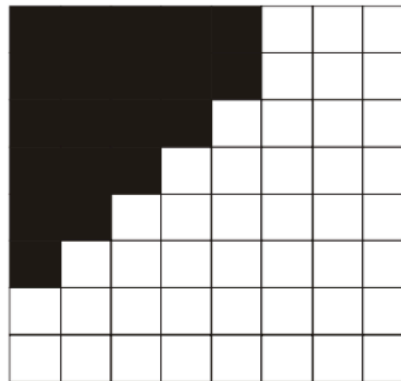
(b) Zoradenie pixelov
a horizontálna SA-DCT



(c) Uloženie 1D
koeficientov

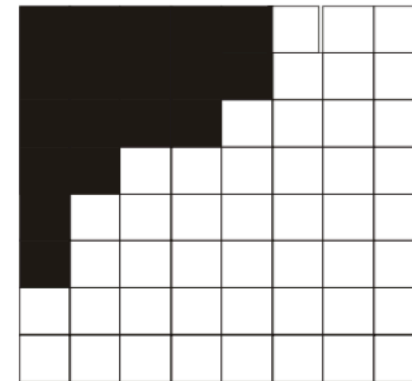


(d) Uloženie vzoriek
pre vertikálnu SA-DCT



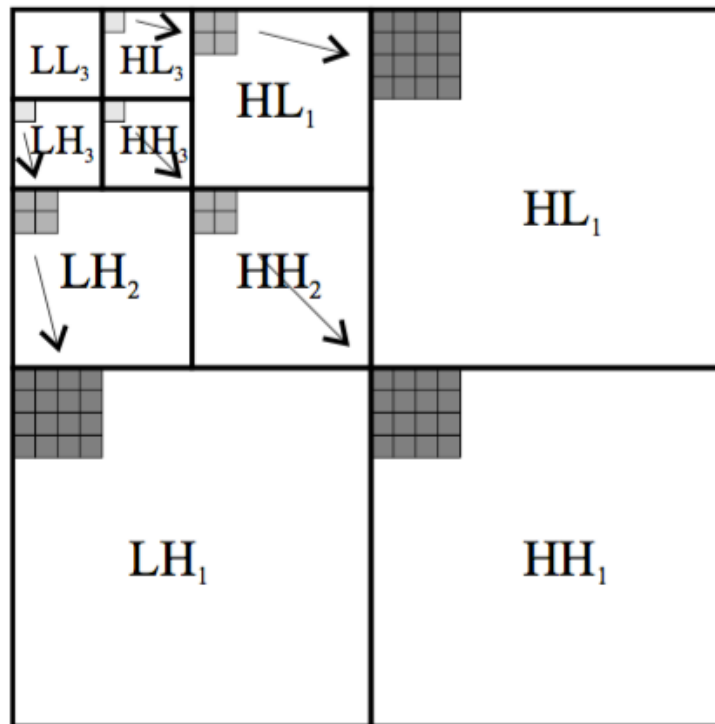
DCT-6
DCT-5
DCT-4
DCT-3
DCT-2

(e) Zoradenie 1D vzoriek
a vertikálna SA-DCT



(f) Uloženie 2D
SA-DCT koeficientov

Waveletové spektrum - závislost v magnitudách koeficientov:



Závislosť v magnitúdach koeficientov:

Túto vlastnosť spektra DWT využívajú viaceré algoritmy na kompresiu obrazu, napr. SPIHT [49] a JPEG 2000 [50]. Hierarchie s nevýznačnými koeficientmi sa nazývajú **stromy núl** [48]. Algoritmy využívajúce hierarchické závislosti sa snažia stromy núl vyhľadávať a následne ich obchádzať pri kódovaní (zakódovať iba ich polohu).

Pri kódovaní obrazu je často dôležitá schopnosť tzv. **postupného prenosu informácie**, kde vyžadujeme prednostný prenos informácie, ktorá nám najviac redukuje chybu pri rekonštrukcii. Potom hovoríme o tzv. **progresívnych kóderoch**.

Označme body pôvodného obrazu $\mathbf{p}$, spektrálne waveletové koeficienty $\mathbf{d}$ a ich rekonštruované verzie $\hat{\mathbf{p}}$, resp. $\hat{\mathbf{d}}$. Potom miera skreslenia E v zmysle strednej kvadratickej chyby (MSE) je pre ortonormálne transformácie invariantná, t. j.:

$$\frac{\sum_i \sum_j (p_{i,j} - \hat{p}_{i,j})^2}{N} = E_{mse}(\mathbf{p} - \hat{\mathbf{p}}) = E_{mse}(\mathbf{d} - \hat{\mathbf{d}}) = \frac{\sum_i \sum_j (d_{i,j} - \hat{d}_{i,j})^2}{N}, \quad (4.12)$$

kde N je počet spektrálnych koeficientov. Zo vzt'ahu 4.12 je zrejmé, že vyslanie presnej hodnoty $d_{i,j}$ do dekódera zníži MSE o $|d_{i,j}|^2/N$. Z toho vyplýva, že koeficienty s väčšou magnitúdou by mali byť prenesené prvé, lebo obsahujú viac informácií.

Tento postup sa dá ešte zefektívniť, ak sa pozrieme priamo na binárnu reprezentáciu $|d_{i,j}|^2/N$ a v zmysle predchádzajúcich úvah budeme prenášať ako prvé ich najdôležitejšie bity a až potom menej význačné bity. Ide v istom zmysle o kódovanie bitových rovín.

Uvedené princípy sú bližšie popísané v nasledujúcej časti na príklade algoritmu SPIHT. Kódovací algoritmus v štandarde JPEG 2000 pracuje na rovnakých princípoch.

SPIHT je založený na troch konceptoch:

1. čiastočné zoradenie koeficientov podľa magnitúdy s prenosom pozičnej informácie pomocou algoritmu na triedenie do množín (algoritmus je duplikovaný v dekóderi)
2. postupný prenos zoradených bitových rovín
3. využitie hierarchickej štruktúry spektra DWT

bitová rovina	m ₅		m ₄		m ₃				m ₂							
znamienko	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s
5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	→	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	→	→	→	1	1	1	1	0	0	0	0	0	0	0	0	0
2	→	→	→	→	→	→	→	1	1	1	1	1	1	1	1	1
1	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→
0	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→

Spresňujúce
bity

Obr. 4.9. Príklad koeficientov utriedených podľa magnitúdy

SPIHT-procesy

Podstatné sú najmä procesy ako sú množiny koeficientov delené a ako je informácia o význačnosti koeficientov prenášaná. Nech koeficienty sú zoradené podľa počtu bitov potrebných na binárnu reprezentáciu ich magnitúdy (pozri obr. 4.9). Dekóder potrebuje na rekonštrukciu koeficientov:

1. informáciu o zoradení (neprenáša sa, v dekóderi je tvorená duplikovaním triediaceho algoritmu)
2. čísla μ_n zodpovedajúce počtu koeficientov $d_{i,j}$ takých, že $2^n \leq |d_{i,j}| < 2^{n+1}$
3. spresňujúce bity.

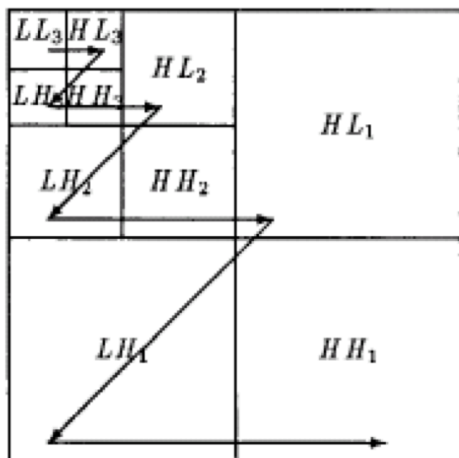
Zjednodušená implementácia progresívneho prenosu informácií je potom nasledovná:

1. pošli $n = \lfloor \log_2(\max_{(i,j)} |d_{i,j}|) \rfloor$ do dekódera
2. pošli μ_n a znamienko pre každý zodpovedajúci koeficient (triediaci krok)
3. pošli n -tý najvýznačnejší bit pre všetky koeficienty $d_{i,j}$ pre ktoré $|d_{i,j}| \geq 2^{n+1}$ (spresňujúci krok)
4. zmenši n o 1 a chod' na krok 2.

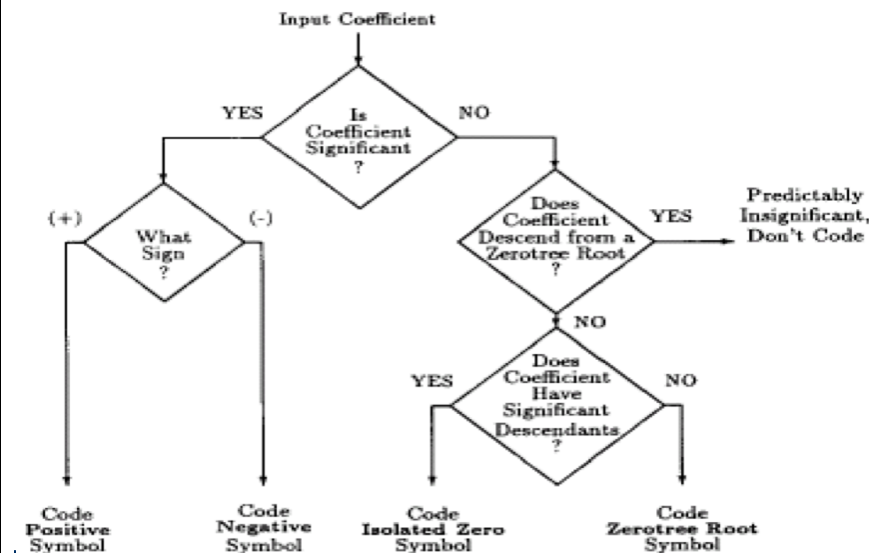
Celý algoritmus má pri kódovaní a dekódovaní rovnakú zložitosť. Príjemná vlastnosť je, že dekóder môže byť identický s kóderom, pričom namiesto „pošli“ vykonáme „načítaj“.

EZW – embedded zero tree wavelet (Shapiro 1993)

Poradie skenovania koeficientov



Kódovanie hodnôt symbolmi:



Hlavná slučka enkódera:

```

 $T \leftarrow 2^{\lfloor \log_2(\max(|Y(x,y)|)) \rfloor}$ 
repeat
    dominantPass(T)
    if image is color then
        dominantPass(T)
        dominantPass(T)
    end if
    subordinatePass(T)
     $T \leftarrow T/2$ 
until  $T = 0$ 
    
```

Dominantný prechod

```

initialize dominant list
while dominant list not empty do
    retrieve and remove coefficient from head of dominant list
    if coefficient  $\neq$  ZTR then
        scan for next coefficient
        code coefficient and add to dominant list
        if coefficient is coded as POS or NEG then
            add absolute_value(coefficient) to subordinate list
            set coefficient position in transformed image to 0
        end if
    end if
end while
    
```

Podriadený prechod

```

while not past end of subordinate list do
    get next coefficient from subordinate list
    coefficient  $\leftarrow$  coefficient -  $T$ 
    if coefficient  $\geq T/2$  then
        output a '1'
    else
        output a '0'
    end if
end while
    
```