

VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Bakalářská práce

bakalářský studijní obor  
**Teleinformatika**

**Student:** David Seifert

**ID:** 106765

**Ročník:** 3

**Akademický rok:** 2009/2010

## NÁZEV TÉMATU:

**Waveletová komprese obrazů pomocí distribuovaných výpočtů v Matlabu**

## POKYNY PRO VYPRACOVÁNÍ:

Nastudujte základy diskrétní waveletové transformace a základní princip komprese obrazů. Navrhněte flexibilní algoritmus pro rozdělení rozsáhlého obrazu na menší bloky včetně metodiky pro distribuci a zpětný sběr výsledků. Tento algoritmus naprogramujte, otestujte a vytvořte přívětivé a efektivní grafické uživatelské rozhraní.

Srovnajte objektivní kvalitu obrazu (PSNR, SSIM) s metodou, kdy se celý obraz zpracovává najednou.

## DOPORUČENÁ LITERATURA:

- [1] Wickerhauser M. V.: Adapted Wavelet Analysis from Theory to Software, IEEE Press, 1994 A K Peters, Ltd.
- [2] Rajmic, P.: Využití waveletové transformace a matematické statistiky pro separaci signálu a šumu (Exploitation of the wavelet transform and mathematical statistics for separation signals and noise, in Czech), PhD Thesis, Brno university of Technology, Brno, 2004.
- [3] Taubman, D.: JPEG2000: Image Compression Fundamentals, Standards and Practice. Springer, 2001. ISBN 978-0792375197

**Termín zadání:** 29.1.2010

**Termín odevzdání:** 2.6.2010

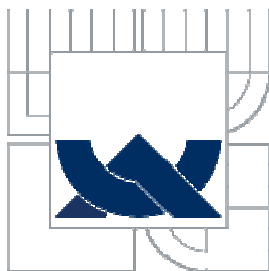
**Vedoucí práce:** Mgr. Pavel Rajmic, Ph.D.

**prof. Ing. Kamil Vrba, CSc.**

*Předseda oborové rady*

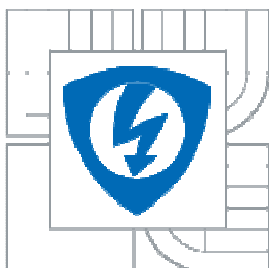
## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

**BRNO UNIVERSITY OF TECHNOLOGY**



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**

**ÚSTAV TELEKOMUNIKACÍ**

**FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION**

**DEPARTMENT OF TELECOMMUNICATIONS**

# **WAVELETOVÁ KOMPRESSE OBRAZŮ POMOCÍ DISTRIBUOVANÝCH VÝPOČTŮ V MATLABU**

**WAVELET-TYPE COMPRESSION OF IMAGES USING DISTRIBUTED COMPUTING IN MATLAB**

**BAKALÁŘSKÁ PRÁCE**

**bachelor's thesis**

**AUTOR PRÁCE**

**AUTHOR**

**DAVID SEIFERT**

## **Anotace**

Bakalářská práce pojednává o možnosti waveletové komprese v programu Matlab. V úvodu je pouze uveden malý přehled historie Waveletovy analýzy a její možnosti. Dále pak následuje kapitola o obecné Waveletové transformaci, základní poznatky o ní. Je zde také uveden Mallátův pyramidový algoritmus. Následuje kapitola o DWT a její využití pro zpracování u obrázkové komprese. Hlavní kapitolu práce tvoří realizace komprese v Matlabu. Je zde uveden důvod dělení obrázku na bloky. Jsou zde také zmíněny 2 základní metody zpracování pomocí Matlabu, a to distributivně a paralelně. V další části kapitoly následuje popis samotného programu. V poslední části kapitoly je uvedeno použití tohoto programu s různě volenými parametry programu.

**Klíčová slova:** Waveletova komprese, DWT, Mallatův algoritmus, Matlab, blokové dělení

## **Abstrakt**

The thesis deals with the possibility of wavelet compression in Matlab. In the beginning it only made me review the history of Wavelet analysis and its possibilities. Then followed a general kapitola wavelet transform, basic knowledge about it. It also identified Mallátův pyramidal algorithm. The following chapter on the DWT and its use for processing the image compression. Chapter One consists of the implementation of compression in Matlab. Here is the reason for dividing the image into blocks. There is also mention 2 basic methods of processing using Matlab, and parallel and distributed. In another part of the chapter followed by a description of the program itself. In the last part of the chapter is set to use this program with different parameters of the elective program.

**Keywords:** Wavelet compression, DWT, Mallatův algorithm, Matlab, block division



Bibliografická citace mé práce:

SEIFERT, D. *Waveletová komprese obrazů pomocí distribuovaných výpočtů v Matlabu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 26 s. 1 příloha. Vedoucí bakalářské práce Mgr. Pavel Rajmic, Ph.D.

## Prohlášení

Prohlašuji, že svoji bakalářskou práci na téma „Waveletova komprese obrazů pomocí distributivních výpočtů v Matlabu“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále

prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne.....

Podpis autora.....

## **Poděkování**

Děkuji vedoucímu práce panu Mgr. Pavlu Rajmicovi, Ph.D. za velmi užitečnou metodickou pomoc a cenné rady při zpracování bakalářské práce.

V Brně dne .....

Podpis autora .....

# Obsah

Úvod	8
1. Waveletova transformace, MRA	9
1.1. Waveletova transformace a DWT	9
1.2. MRA	10
1.3. Měřítková a waveletova rovnice	11
1.4. DWTW a Mallátův pyramidový algoritmus	12
2. Algoritmus DWT u obrázků	14
2.1. DWT u obrázků	14
2.2. Využití DWT pro kompresi	15
3. Realizace komprese pomocí Matlabu	18
3.1. Blokové dělení	18
3.2. Popis programu po blocích	18
3.3. Příklad komprese	21
4. Závěr	25
Seznam obrázků a tabulek	26





## Úvod

Waveletova analýza patří dnes mezi rychle se rozvíjející metody v řadě oborů. Její teoretické počátky sahají někam do počátku 80. let minulého století. Jedná se tedy poměrně o novou disciplínu, která začala být využívána v technické praxi až počátkem 90. let minulého století. O to ovšem progresivněji. Waveletova analýza se progresivně rozvíjí z důvodu velmi dobrých výsledků v mnoha oborech. Mezi ně patří i zpracování obrazu.

Jednou z možností, jak ji využít je právě pro kompresi obrazu. Tato bakalářská práce pojednává o kompresi obrazu pomocí „paralelních“ výpočtů v Matlabu. Nejprve je zde probrán teoretický úvod Waveletové transformace a následně její diskrétní podoba (DWT). Potom se text věnuje v hrubém základu principu komprese obrazu využívající DWT. Následně přichází hlavní náplň této práce a to paralelní zpracování obrazu pomocí Matlabu. Zde je rozebrán princip vytvořeného programu,

# 1. Waveletova transformace, MRA

Kapitola obsahuje základní poznatky o waveletové analýze. Dále pojednává o vlastnostech multirezoluční analýzy. Je zde také zmíněn Mallátův pyramidový algoritmus.

## 1.1 Waveletova transformace

Integrovaná Waveletova transformace je definována na tzv. Hilbertově prostoru (stejně jako waveletová analýza)  $L^2(\mathbb{R})$ , což je prostor, který obsahuje funkce s konečnou energií[1]. Na tomto prostoru jsou definovány i konvergentní řady. Je definována pomocí vztahu[2]

$$CWT_f(a, b) = |a|^{-1/2} \int_{-\infty}^{\infty} f(t) \overline{\psi\left(\frac{t-b}{a}\right)} dt. \quad (1.1)$$

kde  $a \in \mathbb{R}$  bez nuly,  $b$  představuje translaci (posunutí v čase) a  $a$  představuje změnu měřítka. Koeficient  $a$  je zde pro zachování energie při změně měřítka. Takže pokud například dojde ke smrštění vlnky, energie se projeví například zvětšením amplitudy při daném měřítku. Vzorcem [2]

$$DWT_f(j, k) = \left\langle f(t) \psi_{j,k} \right\rangle = \int_{-\infty}^{\infty} f(t) \overline{\psi_{j,k}} dt. \quad (1.2)$$

kde proměnné  $j, k \in \mathbb{Z}$ , definujeme diskretní WT. Je to tedy případ, když bychom do vzorce CWT dosadili za  $a = 2^j$  a za  $b = 2^j k$ , jedná se o spočetně mnoho funkcí[2]. Výraz  $\left\langle f(t) \psi_{j,k} \right\rangle$  vlastně představuje skalární součin pro spojitě funkce[1].

Existují i funkce takové, že  $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$  jsou úplnou ortonormální bází prostoru  $L^2(\mathbb{R})$ . Je-li tomu tak, potom lze funkce  $f(t)$  vyjádřit součtem waveletové řady[2]

$$f(t) = \sum_{j,k \in \mathbb{Z}} \left\langle f(t) \psi_{j,k} \right\rangle \psi_{j,k}(t). \quad (1.3)$$

Pokud navíc mají vlastnost kompaktnosti(báze), potom jsou definovány funkce  $\psi$  na ohraničeném intervalu[2]. Znamená to potom lokální příspěvek funkce, prakticky tedy takový wavelet působí lokálně.

## 1.2 Multirozlišovací analýza

Multirozlišovací analýza (MRA) na prostoru  $L^2(\mathbb{R})$  je definována jako „sada“ podprostorů, kde čím vyšší bude  $j$ , tím déle bude působit v časové oblasti a naopak bude mít užší spektrum v kmitočtové oblasti a naopak. Platí

$$\dots \subset V^{(2)} \subset V^{(1)} \subset V^{(0)} \subset V^{(-1)} \subset V^{(-2)} \dots \quad (1.4)$$

Pro multirozlišovací analýzu platí 5 základních vlastností, podrobněji jsou popsány v literatuře[1]. Zde je uvádím z této literatury pouze hrubě:

- 1)  $\bigcup_{j \in \mathbb{Z}} V^{(j)} = L^2(\mathbb{R})$ . Tato vlastnost říká, že každá funkce  $f \in L^2(\mathbb{R})$  je vlastně limitou ve smyslu toho prostoru svého promítnutí v něm, na postupně se zvyšujícím se rozlišení prostoru. Tato konvergence je právě pouze ve smyslu prostoru  $L^2(\mathbb{R})$ .
- 2)  $\bigcap_{j \in \mathbb{Z}} V^{(j)} = \{0\}$ . Tato vlastnost znamená, že každé  $f \in L^2(\mathbb{R})$  má nenulové rozlišení takové, že jeho promítnutí konverguje k 0, tak jako  $j$  míří k  $\infty$ , kde konvergence je znovu ve smyslu prostoru  $L^2(\mathbb{R})$ .
- 3)  $x(t) \in V^{(0)}$  platí pouze v případě, že zároveň také  $x(2^{-j}t) \in V^{(j)}$ . Takže dilatací signálu z rozlišovacího prostoru  $V^{(0)}$  pomocí faktoru  $2^j$  získáme signál s nižším rozlišení prostoru  $V^{(m)}$ .
- 4)  $x(t) \in V^{(0)}$  pouze za předpokladu, že zároveň také platí  $x(t-n) \in V^{(0)}$  pro všechna  $n \in \mathbb{Z}$ . Kombinací této vlastnosti s výše uvedenými, vidíme, že posunutí signálu v rozlišovacím prostoru  $V^{(j)}$  pomocí všech celočíselných násobků  $2^j$  nemění jeho rozlišení.
- 5) Hovoří o existenci ortonormální báze funkcí  $\{\varphi_k\}_{k \in \mathbb{Z}}$ . Zároveň je pro MRA popsán ortogonální doplněk prostoru  $V^{(j)}$  v prostoru  $V^{(j-1)}$  a to je:  $W^{(j)} \perp V^{(j)}$  a  $W^{(j)} \oplus V^{(j)} = V^{(j-1)}$ .

### 1.3 Měřítková a waveletová rovnice

Důležitou roli hraje myšlenka[1], že prostor  $\mathcal{V}^{(0)} \subset \mathcal{V}^{(-1)}$ , potom měřítková funkce  $\varphi(t)$  může být vyjádřena jako lineární kombinace funkcí  $\varphi_k^{-1}(t) = \sqrt{2}\varphi(2t - k)$ , která vyměřuje prostor  $\mathcal{V}^{(-1)}$ . Potom můžeme psát měřítkovou rovnici

$$\varphi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} h_k \varphi(2t - k). \quad (1.5)$$

Navíc, pokud definujeme pravidlo vztahu mezi souřadnicemi  $h_k$  a  $g_k$  můžeme napsat obdobně waveletovu rovnici

$$\psi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} g_k \varphi(2t - k). \quad (1.6)$$

Funkce  $\psi(t)$  a  $\varphi(t)$  jde tak vyjádřit pomocí lineární kombinace dvakrát smrštěné měřítkové funkce.

Navíc pro dané prostory jednotlivých měřítek platí vztahy[2]

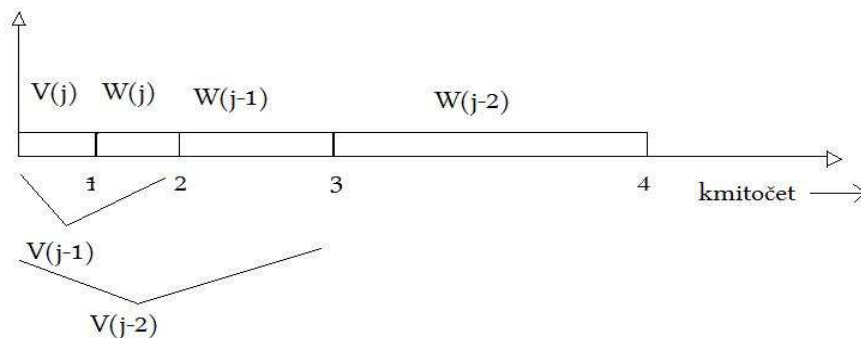
$$W^{(j)} = \left\{ w(t) \middle| w(t) = \sum_{k=-\infty}^{\infty} s_{j,k} \psi_{j,k}(t), s_{j,k} \in C \right\}, j \in Z. \quad (1.7)$$

$$\begin{aligned} \mathcal{V}^{(j)} &= \left\{ x(t) \middle| x(t) = \sum_{i=-\infty}^{j+1} \sum_{k=-\infty}^{\infty} s_{i,k} \psi_{j,k}(t), s_{i,k} \in C \right\} = \\ &= \left\{ x(t) \middle| x(t) = \sum_{k=-\infty}^{\infty} s_{j,k} \varphi_{j,k}(t), s_{j,k} \in C \right\}, j \in Z. \end{aligned} \quad (1.8)$$

Potom se dá každou funkci  $f \in L^2(\mathbb{R})$  rozložit podle MRA analýzy na součet navzájem ortogonálních funkcí

$$f(t) = \sum_{l \in Z} a_{0l} \varphi(t - l) + \sum_{k, j \in Z, j \leq 0} b_{j,k} \psi_{j,k}(t) \quad (1.9)$$

kde  $a_{0l}, b_{j,k}$  náležící do  $C$  jsou souřadnicemi  $f$  v báзовých prostorech.



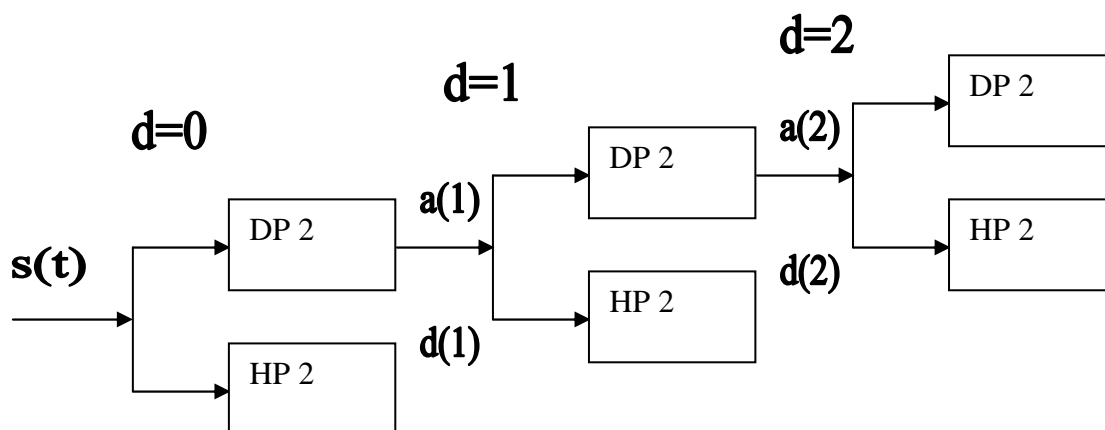
Obr. 1.1. Znáznorňuje rozsah jednotlivých prostorů  $W$  a  $V$  na kmitočtové ose

## 1.4 DWTW a Mallatův pyramidový algoritmus

DWTW je vlastně konečná diskretní waveletová transformace. Byla odvozena z důvodu konečné délky signálů. Lze ji vyjádřit pomocí ortogonální matice  $W$ [2]. Důležitá je ovšem realizace. Ta se provádí pomocí banky filtrů. Banka filtrů je v podstatě spojení více filtrů za sebou. Skládá se ze dvou částí-analytické a syntézní[3]. V analytické části se provádí analýza signálu pomocí kvadraturních zdrcadlových filtrů-jedná se o dolní a horní propust. Výsledkem jsou dva signály – ten z té dolní propusti – tzv. aproximační složka a signál z horní propusti – tzv. Detailní složka. Toto označení vzniklo z předpokladu, že každý signál má základní informace uložené v nízkofrekvenčním pásmu. Vysokofrekvenční představuje jeho detailní informaci. V tomto třídění lze pokračovat i dále. Akorát dojde k ještě větší komprimaci signálu. Není nutné zachovávat všechny vzorky vznikajících signálů za jednotlivými filtry z první části neboť filtrací se zmenšuje velikost kmitočtového pásma. Tím můžeme dosáhnout postupného snižování vzorkovacího kmitočtu. Vynecháním třeba každé sudé složky provádíme tzv. decimaci subsignálů s činitelem 2[3]. Podobně na syntézní straně se pro změnu provede zase interpolace tj. vložení stejného počtu vzorků (vkládány nuly) do subsignálů. Potom se naopak vzorkovací kmitočet zase zvyšuje.

Pokud ovšem pokud provádíme vlnkovou analýzu je algoritmus pyramidový(zpracovávají se signály pouze z dolních propustí). Tento algoritmus mezi vlnkovými funkcemi a bankou filtrů poprvé popsal Stéphane Mallat[3]. Přesto je potřeba

uchovávat i všechny výstupy z horních propustí. Dosáhne se daleko jemnějšího vícenásobného rozlišení. Postup Mallátova algoritmu zachycuje obr. 1.2..



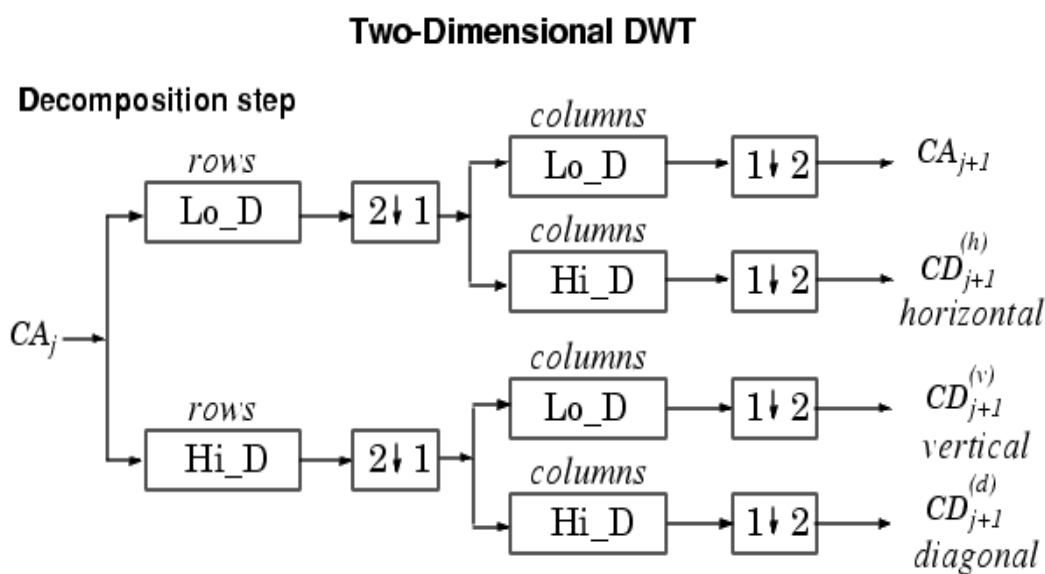
Obr. 1.2. Mallátův algoritmus

## 2. Algoritmus DWT u obrázků

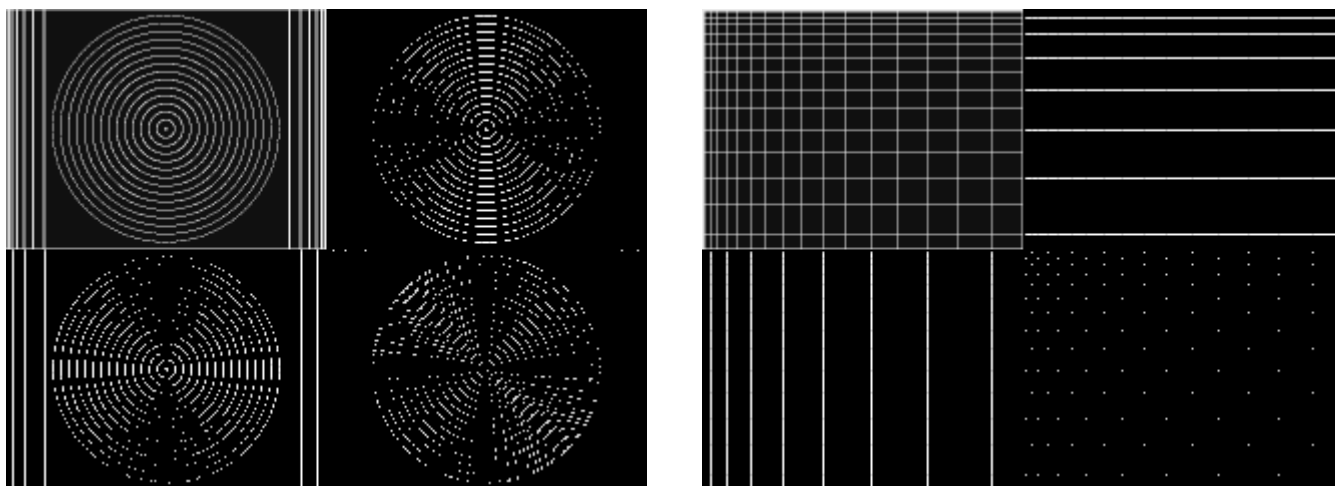
Tato kapitola obsahuje princip dekompozice u obrázků, tedy 2-D signálů. Následně je zde uveden v základech princip komprese.

### 2.1 DWT u obrázků

U 2-D signálu probíhají 2 kroky u jednoho stupně dekompozice[4] viz. obr. 2.1.. Při prvním jsou nejprve filtrovány řádky matice obrázku a následuje decimace sloupců. Při druhém jsou filtrovány sloupce matice obrázku a potom nastává decimace řádků. Ve výsledku dostaneme 4 obrázky, které při sečtení svých velikostí dávají dohromady velikost obrázku, který byl podroben dekompozici. Význam jednotlivých obrázků po dekompozici je dobře patrný z obr. 2.2.[5].

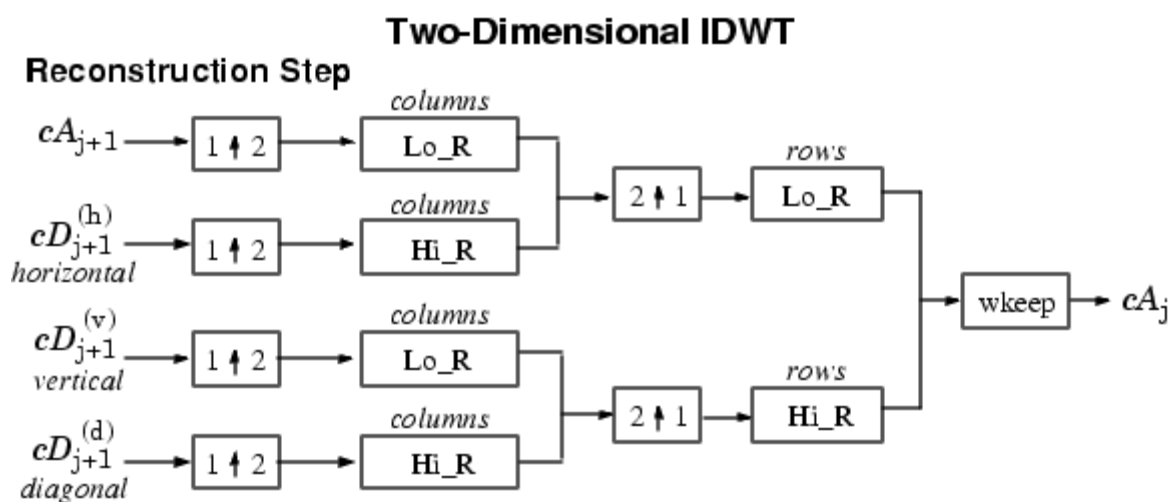


Obr. 2.1. Ukazuje jeden dekompoziční krok 2D signálové analýzy pomocí DWT



Obr. 2.2. Znáznorňuje 2 různé obrázky po provedení jednoho kroku dekompozici

Podobně se postupuje i při inverzi – rekonstrukci. Vezmou se uchované detailní koeficienty a aproximační, které představují o 1 stupeň vyšší hloubky dekompozice než je tomu u výsledného zrekonstruovaného obrázku. Jednotlivé kroky jsou skutečně oproti dekompozici zrcadlově obrácené. Tento postup zachycuje obr. 2.3.[4].



Obr. 2.3. Popis jednoho rekonstrukčního kroku 2D signálové analýzy pomocí DWT

## 2.2 Waveletova komprese obrazu

Jak už z názvu vyplývá tato komprese je založena na Waveletové transformaci – resp. na její podobě v DWT. Waveletova komprese patří mezi ztrátové komprese čili už při její dekompresi nikdy nezískám původní obrázek. Ovšem právě tohoto jevu tato komprese



využívá. Důležitým faktem je, že základ obrázku leží v aproximační složce. Tedy složce, kterou, jak bylo uvedeno výše, dostanu přes 2 DP. Zatímco detailní koeficienty zobrazují ne až tak podstatné rysy obrázku. A právě proto kompresi dosáhnou vynecháním – resp. vynulováním detailních koeficientů.

Kompresní blok se v podstatě skládá z těchto částí viz. obr. 2.4.[4]:

- z bloku realizující DWT
- z práhování koeficientů DWT a jejich následného kvantování
- z kódování, po něm následuje přenos nebo uložení

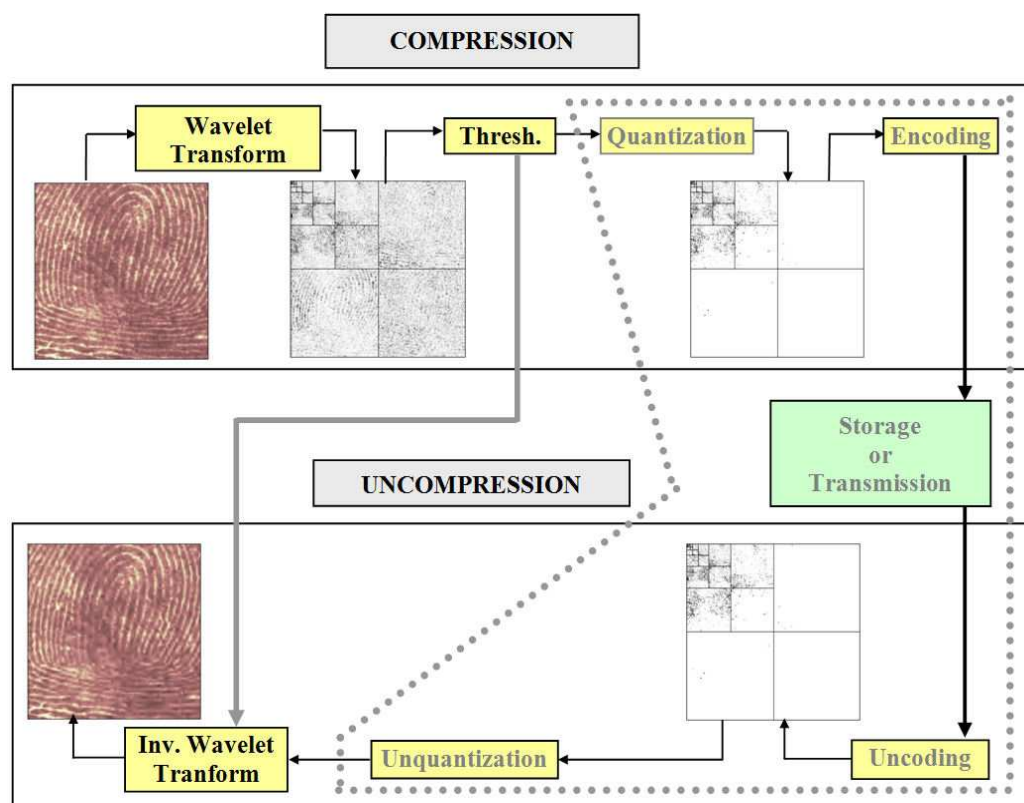
O tom, jak budou tyto bloky využívány rozhoduje zvolená kompresní metoda. První 3 kompresní metody realizují v jednom kroku kaskádně za sebou operace práhování koeficientů(globálně nebo na určité úrovni) a kódování podle kvantizace. Pevné nebo Huffmanovo kódování lze použít pro kvantování v závislosti na metodě. Těmito metodami jsou:

- globální práhování koeficientů a pevné kódování
- globální práhování koeficientů a Huffmanovo kódování
- subpásmové práhování koeficientů a Huffmanovo kódování

Tyto metody jsou často nazývané jako Metody práhování koeficientů(CTM).

Existují i propracovanější metody, které kombinují waveletovou dekompozici a kvantizaci.

Tento princip je základem tzv. progresivních metod(PTSM).



Obr. 2. 4. Znázornění kompresního a dekompresního řetězce

## 3. Realizace v Matlabu

### 3.1. Blokové dělení

Důvodem proč se realizuje blokové dělení obrázku je rychlejší zpracování výpočtů souvisejících s Waveletovou kompresí. Na řešení programu se totiž v tomto případě nepodílí pouze jeden ale více procesorových jednotek, které v případě rozumného dělení do bloků umožní rychlejší dosažení výsledku. Dělení do bloku má samozřejmě smysl právě za předpokladu rozsáhlého obrázku, tedy obrázku o masivní velikosti. Zároveň nemá smysl posílat paralelně sítě příliš malé bločky.

[4]V Matlabu v podstatě existují 2 možnosti „souvislých“ výpočtů. Jedné se říká paralelní zpracování dat a druhé distributivní. U paralelního zpracování řeší všichni pracovníci jednu úlohu, dokud tato úloha není hotova nepřechází se k další úloze. Paralelní pracovníci o sobě ví, tzn. ví i jak vypadá celková struktura zpracovávaných dat. U distributivního zpracovávání dat o sobě pracovníci neví. Každá úloha je přidělena jednomu pracovníkovi. Každý jiný pracovník o těch dalších dat, které nezpracovává, neví. Nečeká se tedy až nějaký pracovník dodělá svůj úkol, plánovač pouze přiděluje těm, kteří již mají hotovo. Nevýhodou je, že není možnost nějak programovat plánovač, který řídí právě tento management pracovníků. Existuje více druhů plánovačů v závislosti na tom, jestli organizujeme výpočty pouze na jednu PC nebo je posíláme na více jednotek. I Matlab podporuje několik plánovačů.

V této práci se uvažovalo pro více procesorových jednotek v jednom počítači. Bylo sice tedy užito distributivních příkazů, ale místního plánovače. Takže data nejsou posílána na cluster.

### 3.2. Popis programu

Uživatel má možnost volit několik vstupních parametrů pro obrázek. Jedná se o typ vlny, hloubku decimace, rozměry bloku, na který bude obrázek dělen, tj. k jsou řádky a l jsou sloupce a hodnotu práhování. Dále musí napsat přesný název obrázku, o který se jedná a jeho typ, to je např. JPEG.

V programu se nejprve vyskytují proměnné m a n. Ty udávají velikost obrázku, který je načítán, toho je dosaženo pomocí příkazu size. Dále se zde vytvoří v paměti místo pro budoucí zrekonstruovanou matici Xr a to pomocí příkazu zeros. Proměnné a a b jsou využity ve dvou smyčkách, které sledují jednotlivé bloky, na které bude obrázek rozdělen. To je třeba pro a=2 a b=1 se jedná o první bloček na druhém řádku. Pak se zde nachází proměnná pan.

Tato proměnná souvisí s jedním cyklem, který má v sobě 4 periody. Je tedy na místě už potřeba říci, že program realizuje výpočet se 4 pracovníky. To je bohužel pro tuto práci neměnné. Tyto 4 cykly jsou právě závislé na počtu pracovníků. Proměnná tedy „hlídá“ aby byly předány vždy 4 bločky. Matice B1 až B4 jsou matice, se kterými pracují daný pracovníci. V programu je B1 vymezen pro pracovníka 1, B2 pro druhého apod. Potom se zde nachází pomocná matice Q. Její velikost je dána právě počtem pracovníků a proměnnými a a b. Je zde z toho faktu, že po objetí konkrétního případu ve smyčkách se proměnné a a b nikde neukládají a tak možnost zjištění, kde se nalézal daný bloček pro informaci právě zrekonstruované matice, by chybělo. Z tohoto důvodu jed zde právě matice Q, ukládají se totiž právě do této matice a po vybrání výsledku z pracovníku(po jednom cyklu) se informace z této matice vyberou zase zpět a program tak ví, kam dané bločky patří. Dále se zde nachází 4 vektory pod označením var1 až var4, ty slouží pro prakticky podobný důvod, jako matice Q, s tím rozdílem, že jejich informace nese v sobě „druh“ bločku. Při procesu dělení obrázku na bločky mohou totiž nastat 4 případy toho, jak obecně bude daný bloček vypadat. Buď bude mít klasický rozměr daný parametry k a l nebo se bude jednat o poslední blok v řádku. Ten je tedy zkrácen ve sloupci. Třetí možnost je, že se bude jednat o bločky v posledním řádku, ty jsou zkráceny právě o rozměr v řádku. Poslední možností je nastání obou předchozích případů, to je v případě, že nastal poslední bloček obrázku. Potom se zde nachází první objekt s názvem sched. Jak už trochu název vypovídá jedná se o plánovač. Příkazem findResource vytvořím plánovač a jeho typ, který bude veden pod názvem objektu sched. Objekt je zde obecně něco, čemu matlab přiřazuje určité vlastnosti a parametry. Následně zde máme objekt pjob. Jedná se o práci, kterou bude daný plánovač přerozdělovat vytvořeným pracovníkům(dále). Každá práce může v sobě zahrnovat nespočet úloh, které jsou přerozdělovány, jak bylo řečeno výše, podle toho zda se jedná o „paralelní“ výpočty nebo „distributivní“. Vytvoření práce značí příkaz createJob. Pak se zde nachází již samotné smyčky s proměnnými a a b pro rozdělení do menších bločků. V nich jsou řešeny právě případy, jaký typ bločku nastal a podle toho dále zpracovávány. Zde se právě vyskytuje proměnná pan, která hlídá, aby nastaly pouze 4 cykly- jeden bloček pro jednoho pracovníka. Potom jsou zde „souřadnicové“ rovnice, které popisují, kde se nachází jednotlivé bločky. Dále je volána funkce rozřazení-fhandle.

Funkce fhandle vyřizuje právě přiřazení rozdělených bločků do matic B1 až B4. Dále zpracovává proměnnou pan, matici Q a proměnné a a b. Pokud proměnná pan nevychází z na výstupu z tohoto pomocného programu rovna jedné, pak bude ještě jeden cyklus pokračovat. Není totiž úplné přiřazení. Ukládají se zde právě proměnné a a b do matice Q. Následně je

pan proměnná zvýšena o jedničku. Potom funkce vrací výstup a program pokračuje dále, jakmile tato část programu je oběhnuta čtyřikrát (je tedy proměnná pan rovna jedné), následuje další část programu, která má za úkol vytvořit úlohy pomocí příkazu `createTask`. Toto vytvoření přiřazuje objektu práce objekty úloh. Jsou vedeny pod názvy `t1` až `t4`. Dále se zde v příkazu vyskytuje pojem jádro, to je další funkce, která bude volána pro zpracování daných bločků.

Funkce jádro obsahuje dekompozici, práhování a následnou rekonstrukci pomocí zbylých koeficientů. Takže funkce jádro vlastně provádí samotnou kompresi. Po ní je už vrácen bloček, který je zrekonstruován a bude čekat na přiřazení. Funkce `createTaks` ještě obsahuje informaci o počtu vrácených struktur. A jsou zde ještě parametry, které jsou předávány jednotlivým pracovníkům. Dále je zde příkaz `submit`, ten zařizuje, že dané úlohy jsou dány do fronty plánovače, resp. daná práce je dána do fronty plánovače. Následně přejímají řízení pracovníci. Aby program čekal do té doby než budou zpracovány výsledky, je zde využito příkazu `waitForState`. Ten zařídí, aby se program nepohnul do té doby, než nebudou pracovníci úplně hotovy. Až se tak stane bude program puštěn dále. Následně je potřeba výsledky nějak sebrat od plánovače. K tomu se používá příkaz `getAllOutputArguments`. Ten zaručí, že budou vybrány výsledky. Ty jsou pod názvem `result`. Jedná se právě o tu jednu strukturu, která je na výstupu. Následně je potřeba vymazat v paměti tyto ukončené práce, protože výsledky už máme, resp. právě ukončenou práci. Příkaz, ale zajistí vymazání i předchozích dokončených prací. K tomu se používá příkaz `destroy`. Nakonci tohoto odstavce programu následuje zase nové vytvoření práce.

Nabízí se otázka, proč je nutné vytvářet nový objekt práce, když by stačilo přiřadit dané práci nové úlohy. Bohužel tato myšlenka není možná. Objekty mají i své takzvané statusy. Jakmile status práce dojde do statusu `finished`, nedá se s ním udělat nic jiného, než jej vymazat.

Následně je ještě zapotřebí získané výsledky nějak zpracovat. Proto se zde nachází ještě smyčka `for` s proměnnou `v`. V nabývá 4 hodnot, protože výsledky jsou také čtyři. Zde se právě využívá matice `Q` pro zjištění, kam daný bloček patřil. Následně jsou vypočteny „souřadnice“ pomocí souřadnicových rovnic. Ty udávají odkud kam se daný bloček táhne. Ve smyčce pak dojde k přiřazení na dané místo v rekonstruované matici a to je tedy stejné, jako v té původní obrázkové matici. Nakonec smyčky je ještě potřeba vynulovat vektory `var1` až `var4`, jinak by mohlo dojít k omylu ohledně tvaru bločku, při dalším cyklu.

Až je vyhotoven celý obrázek, je daná práce opět smazána. A následně je obrázek zobrazen.

### 3.3. Příklad komprese

Byl vybrán náhodný obrázek, který je šedobílý, tedy je ve stupních šedi a má tak všechny 3 2D matice stejné-obsahuje stejné hodnoty. Vzhled tohoto obrázku znázorňuje obr. 3.1.a). Pokud je jako hodnota práhování zadána nula, pak logicky k žádné kompresi nedochází a jsou při rekonstrukci využity všechny koeficienty. Obrázek by vypadal naprosto stejně, jako ten výchozí. Pro lepší vidění je stejný obrázek přiblížen 3 krát(viz. obr. 3.1.b).



a)



b)

Obr. 3.1. Obrázek, který byl vybrán pro příklad komprese

Budeme uvažovat hloubku dekompozice 2 a bločky budou mít až na krajní velikosti 15 x 15 pixelů. Je to pouze z důvodu přehlednosti. Použitou vlnkou bude Daubechies řádu čtyři – tj. db4. Nyní ovšem práhovací hodnota bude 200. Nyní už velikost paměti, kterou obrázek zabírá bude menší, protože došlo k vynulování množství koeficientů. Přesto při ptačí perspektivě(tedy pohledu na celý obrázek) není komprese až tak značná, jak je vidět na obr. 3.2.a). Ovšem při pohledu na obrázek 3 přiblížený jsou již vidět volené bloky rozměru 15 x 15 pixelů a komprese je pro tuto velikost již značná.



a)



b)

Obr 3.2. Zvolena práhová hodnota 200

Pokud budeme dále zvyšovat hodnotu práhu situace už bude obdobná jak to zachycuje Obr. 3.3. V tomto případě byla zvolena práhová hodnota 5000.



a)



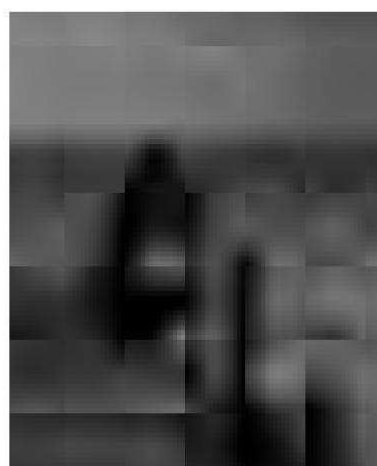
b)

Obr. 3.3. Zvolena práhová hodnota 5000

Nyní už zkusíme změnit hloubku dekompozice a to na úroveň 3. Stane se tak pro práhovou hodnotu 5000. Tento případ je vidět na obr. 3.4. Zároveň provedeme případ pro stejnou hodnotu práhování, ale s hloubkou dekompozice 4. Ten je vyobrazen na obr.3.5..Je vidět, že obrázky se znatelně změnili a prakticky použitelný už není ani celkový pohled. Zvláště u hloubky dekompozice 4 jsou již bloky 15x15 pixelů vidět na celkovém obrázku. Tento jev je dán tím, že matlab musel uchovávat větší vektor detailních koeficientů. Stejně tak jich musel více ze své paměti vyřadit po práhování. Proto již není schopen lepšího výsledku po kvalitní stránce.



a)



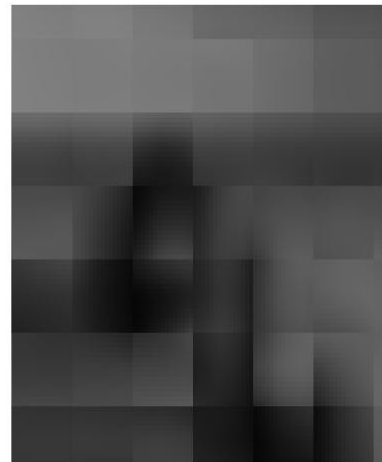
b)

Obrázek 3.4. Zvolena práhová hodnota 5000 a hloubka dekompozice 3





a)



b)

Obrázek 3.5. Zvolena práhovací hodnota 5000 a hloubka dekompozice 3

## **4. Závěr**

Byl předveden konkrétní případ realizace waveletové komprese obrazů. Nevýhodou programu samotného je, že zpracovává pouze černobílé obrázky a nelze měnit 4 pracovníky, které jsou pevně dány.

## SEZNAM OBRÁZKŮ

- Obrázek 1.1. -Znázorňuje rozsah jednotlivých prostorů  $W$  a  $V$  na kmitočtové ose
- Obrázek 1.2. -Znázornění Mallátova algoritmu
- Obrázek 2.1. -Ukazuje jeden dekompoziční krok 2D signálové analýzy pomocí DWT
- Obrázek 2.2. -Znázorňuje 2 různé obrázky po provedení jednoho kroku dekompozici
- Obrázek 2.3. -Popis jednoho rekonstrukčního kroku 2D signálové analýzy pomocí DWT
- Obrázek 2.4. - Znázornění kompresního a dekompresního řetězce
- Obrázek 3.1. - Obrázek, který byl vybrán pro příklad komprese
- Obrázek 3.2. - Zvolena práhová hodnota 200
- Obrázek 3.3. - Zvolena práhová hodnota 5000
- Obrázek 3.4. - Zvolena práhová hodnota 5000 a hloubka dekompozice 3

## Literatura

- [1] Taubman, D.: *JPEG2000: Image Compression Fundamentals, Standards and Practice.*, Springer, 2001. ISBN 978-0792375197
- [2] Rajmic, P.: *Využití waveletové transformace a matematické statistiky pro separaci signálu a šumu* (Exploitation of the wavelet transform and mathematical statistics for separation signals and noise, in Czech), PhD Thesis, Brno university of Technology, Brno, 2004
- [3] Smékal, Z., Vích, R.: *Číslíkové filtry*, Academia, 2000. ISBN 80-200-0761-X
- [4] Misiti, M. Misiti, Y., *Wavelet Toolbox 4: User's Guide*, The MathWorks, 1997-2009
- [5] URI: <http://goro.czweb.org/>

## A PŘÍLOHA

V příloze je dodán samotný program pro matlab.