

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

SEGMENTOVANÁ VLNKOVÁ TRANSFORMACE OBRAZU

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MICHAL KUČERA

BRNO 2010



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

SEGMENTOVANÁ VLNKOVÁ TRANSFORMACE OBRAZU

SEGMENTED WAVELET TRANSFORM OF IMAGES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MICHAL KUČERA

VEDOUČÍ PRÁCE

SUPERVISOR

Ing. ZDENĚK PRŮŠA

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Michal Kučera

ID: 78536

Ročník: 2

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Segmentovaná vlnková transformace obrazu

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte principy separabilní vlnkové transformace obrazu a seznámte se s různými druhy prodlužování na okrajích signálů. Zhodnoťte vliv typu prodlužování a vhodnost zařazení úplného prodloužení (pro všechny úrovně) před vlastní transformací. Dále nastudujte algoritmus SegWT pro dopřednou a zpětnou vlnkovou transformaci a zobecněte jej pro použití na obrazových datech. Kvantifikujte redundanci výpočtu vzhledem ke klasickému algoritmu. Navržený algoritmus implementujte v Matlabu.

DOPORUČENÁ LITERATURA:

- [1] VARGIC, R.: Wavelety a banky filtrů. Skripta, Slovenská technická univerzita, Bratislava. 2003, 135s
- [2] RAJMÍČ, P.: Využití waveletové transformace a matematické statistiky pro separaci signálu a šumu, Doktorská dizertační práce, Vysoké učení technické v Brně, Brno, 2004.
- [3] MALLAT, S.: A Wavelet Tour of Signal Processing. 2nd edition, Academic Press, 1999. ISBN 0-12-466606-X

Termín zadání: 29.1.2010

Termín odevzdání: 26.5.2010

Vedoucí práce: Ing. Zdeněk Průša

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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.

ABSTRAKT

Tato diplomová práce se zabývá úpravou klasického algoritmu výpočtu diskretní waveletové transformace s diskretním časem tak, aby bylo možné rozdělit vstupní obraz na segmenty a ty nezávisle zpracovávat. Je představena a implementována segmentovaná waveletová transformace s diskretním časem, díky které docházíme ke stejným koeficientům, jako když je zpracováván celý obraz najednou. V kombinaci s inverzní segmentovanou waveletovou transformací je umožněno kompletní nezávislé zpracování segmentu obrazu tj. dekompozice do určené hloubky, zpracování koeficientů a následná rekonstrukce bloku obrazu.

Je zřejmé, že při segmentaci budou vznikat nadbytečné výpočty. Redundance stoupá především s narůstající hloubkou dekompozice a délkou filtru, naopak se zvýšením rozměru segmentu nadbytečnost klesá. K tomu dochází z důvodu, že pracujeme s menším počtem segmentů, které je třeba prodlužovat, a právě prodloužení sousedními vzorky je zde zdrojem výpočtů navíc.

Práce obsahuje také popis dvou různých přístupů k prodloužení signálu na jeho okrajích: úplné prodloužení pro všechny úrovně dekompozice před vlastní transformací a klasický přístup, kdy je signál prodlužován během transformace v každém jejím kroku. V provedených testech mají obě metody totožné výsledky po rekonstrukci obrazu, liší se pouze ve vypočtených waveletových koeficientech.

Dochází též k porovnání čtyř druhů prodloužení signálu na okrajích pomocí MSE a PSNR. Výsledky všech přístupů jsou podobné, přesto ale lze favorizovat prodloužení konstantou a symetrické prodloužení, která dosáhla nižší MSE, oproti prodloužení polynomem prvního řádu a prodloužení nulami.

KLÍČOVÁ SLOVA

waveletová transformace obrazu, inverzní transformace, segmentace, prodloužení signálu, redundance

ABSTRACT

This master's thesis is dealing with modification of classical discrete time wavelet transform algorithm to be able to split input image into several segments which could be independently processed. Segmented discrete time wavelet transform is introduced and implemented and the tests shows that it produces the similar values of wavelet coefficients as the classical approach. Complete independent image segment processing is allowed thanks to inverse segmented discrete time wavelet transform.

It's obvious that redundancy calculations appear when the image is processed segment by segment compared to transform the image at once. The redundancy rises with increasing decomposition depth and filter length, however it falls when the segment dimensions are increased. It happens because it is worked with smaller amount of segments which needs to extend. The extension with neighbor samples is the only source of redundancy.

The thesis also contains description of two different approaches to signal border extension: firstly complete extension for all decomposition levels before the transformation itself and secondly classical method when the signal is extended in each decomposition step. Both approaches have the same results of image reconstruction, but it distinguishes in calculated wavelet coefficients.

Four kinds of signal border extension are compared by using MSE and PSNR. The results of all approaches are similar, however the smooth padding of order 0 and symmetric-padding could be slightly favored against smooth padding of order 1 and zero-padding which shows higher MSE.

KEYWORDS

wavelet transform of image, inverse transform, segmentation, signal border extension, redundancy

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Segmentovaná vlnková transformace obrazu“ jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové 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.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Zdeňku Průšovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování práce.

V Brně dne

.....

(podpis autora)

OBSAH

Úvod	11
1 Waveletová transformace	13
1.1 Spojitá waveletová transformace	13
1.2 Diskrétní waveletová transformace	15
1.3 Algoritmus výpočtu diskrétní waveletové transformace s diskrétním časem	15
1.4 Dvourozměrná waveletová transformace	18
2 Zpracování okrajů signálu	20
3 Segmentovaná waveletová transformace	23
3.1 Segmentace a prodlužování signálu	23
3.2 Dvourozměrná segmentovaná waveletová transformace	24
3.3 Inverzní segmentovaná waveletová transformace	27
4 Implementace algoritmů	30
4.1 Upravený Mallatův pyramidový algoritmus	30
4.2 Algoritmus SegDTWT	30
4.3 Algoritmus SegIDTWT	32
5 Popis programu	34
5.1 Ověření funkčnosti	34
5.2 Měření odchylek	36
5.3 Nadbytečnost	37
6 Vliv typu prodloužení	40
7 Závěr	45
Literatura	47
Seznam symbolů, veličin a zkratk	49
Seznam příloh	50
A Obsah přiloženého CD	51

SEZNAM OBRÁZKŮ

1.1	Příklady mateřských waveletů podle Daubechiesové	14
1.2	Waveletová dekompozice o hloubce 3	16
1.3	Jeden krok waveletové rekonstrukce	17
1.4	Jeden krok dvourozměrné waveletové dekompozice	18
1.5	Praktická ukázka rozložení obrazu pomocí dvourozměrné DTWT . . .	19
2.1	Půlbodové a celobodové symetrické prodloužení	21
2.2	Testovací obraz a jeho prodloužení na okrajích v programu MATLAB	21
3.1	Segmentace jednorozměrného signálu a prodloužení segmentů	24
3.2	Segmentace a prodloužení dvourozměrného signálu	26
3.3	Průběh skládání segmentů při SegIDTWT	28
5.1	Grafické uživatelské rozhraní programu segdtwt	34
5.2	Graf závislosti nadbytečnosti výpočtu SegDTWT na maximální délce prodloužení	38
6.1	Grafy průměrné MSE a PSNR v závislosti na typu prodloužení	41

SEZNAM TABULEK

3.1	Závislost délky levého a pravého prodloužení na délce segmentu . . .	25
5.1	Ověření funkčnosti programu – odchylky waveletových koeficientů . .	35
5.2	Závislosti nadbytečnosti výpočtu SegDTWT na délce maximálního prodloužení	38
6.1	Hodnoty MSE a PSNR pro různé druhy prodloužení – sea.bmp	42
6.2	Hodnoty MSE a PSNR pro různé druhy prodloužení – bridge.bmp . .	43
6.3	Hodnoty MSE a PSNR pro různé druhy prodloužení – building.bmp .	44

ÚVOD

Přínos waveletové (česky vlnkové) transformace je pro mnoho odvětví lidské činnosti jistě neodmyslitelný. Jako způsob zpracování signálů nachází uplatnění mimo jiné i při odstraňování šumu ze signálu, detekci objektů a hran v obraze a samozřejmě v kompresi signálů.

Kompresi obrazu založenou na waveletové transformaci umožňuje například formát JPEG2000, od kterého se původně očekávalo, že nahradí dnes stále zdaleka nejrozšířenější JPEG, který používá k analýze signálu kosínovou transformaci. Hlavní výhodou novějšího formátu je výrazně lepší vizuální kvalita obrazu při stejném objemu dat, jež je způsobena na pohled méně rušivými kompresními artefakty. Waveletová transformace zpracovává obraz jako celek a i při silném prahování koeficientů zachovává základní informace o signálu, to je velké jednolitě plochy, objekty, tvary. Naopak detaily jsou ztraceny, což se na pohled jeví jako celkové rozmazání obrazu, ale pro člověka stále není složité určit, na co se dívá.

Opačný přístup je aplikován při použití formátu JPG, kdy se obraz nejprve rozdělí na malé bloky, které jsou zpracovávány samostatně. Při silném prahování je z každého bloku zachována pouze stejnosměrná složka a celý blok tím pádem má jednolitou barvu. Proto typickým kompresním artefaktem pro JPG je „rozkostičkování“, ze kterého se hůře odhaduje, co na obraze původně bylo.

S odlišným přístupem jde ruku v ruce i výpočetní náročnost převodu signálu do jiné reprezentace. Kosínová transformace malých bloků, které mohou být počítány jeden po druhém, není náročná ani na paměťové prostředky, ani výpočetní výkon. Naopak u waveletové transformace počítáme s celým obrazem najednou, což např. u dnes běžného kompaktního digitálního fotoaparátu je nutné zpracovat i více než 10 milionů pixelů. Zde jsme narazili na důvod mizivého rozšíření formátu JPEG2000. Výpočetní náročnost, díky které by výrazně rostla spotřeba energie, v kombinaci s obrovským rozvojem flashových pamětí, které dnes dosahují proti minulosti obrovských kapacit za přijatelné ceny, nedovolily proniknout tomuto formátu do běžné spotřební elektroniky.

Hlavní nevýhodou tedy je zpracování celého obrazu najednou v krocích, které na sebe přesně navazují. Myšlenka provést waveletovou transformaci po segmentech, by tuto nevýhodu mohla částečně eliminovat. Současným trendem ve výpočetní technice už dávno není honba za vysokými frekvencemi jediného procesoru, na začátku jedenadvacátého století se čím dál více soustředíme na možnosti paralelního zpracování dat. Frekvence čipů zůstává víceméně stejná, zvyšuje se ale počet jader a na začátku roku 2010 jsme se dočkali uvedení šestijádrových procesorů i pro běžné pracovní stanice od obou hlavních výrobců. Navíc se objevují technologie jako Hyperthreading, která umožňuje např. právě šestijádrovému procesoru zpracovávat na-

jednou až 12 výpočetních vláken. Dalším nově se utvářejícím odvětvím je provádění složitých výpočtů na vysoce paralelizovaných čipech grafických karet.

Aplikace segmentované waveletové transformace je tedy zřejmá. Rozdělme obraz na bloky a nechme každý blok zpracovávat jedním jádrem vícejádrového procesoru. Úspora času při výpočtu bude výrazná. Nemůžeme ale předpokládat, že při použití čtyřjádra dostaneme výpočet čtyřikrát rychleji. Rozdělení na segmenty bohužel přináší poměrně značnou redundanci výpočtů, která se zvyšuje s klesající velikostí zpracovávaného segmentu.

Tato práce ukáže, že je možné waveletovou transformaci obrazu segmentovat, aniž by výsledek byl odlišný od standardního způsobu zpracování a nalézt algoritmus, který toto provede. Zaměřujeme se tedy především na paralelní zpracování, nikoli na zpracování dat v reálném čase, což je další z možných využití segmentované waveletové transformace. Waveletová reprezentace obrazu pak může sloužit ke kompresi či separaci šumu z obrazu nebo k dalším účelům.

1 WAVELETOVÁ TRANSFORMACE

Wavelety neboli vlnky lze použít k rozkladu a opětovnému složení signálu. V této kapitole jsou uvedeny definice a vlastnosti spojité waveletové transformace, diskrétní waveletové transformace a diskrétní waveletové transformace s diskrétním časem.

1.1 Spojitá waveletová transformace

Spojitá waveletová transformace – CWT [2] funkce $f(t)$ je popsána vztahem

$$S_{\text{CWT}}(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt, \quad (1.1)$$

kde $a < 0, b \in \mathbb{R}$. Hodnoty spektra $S_{\text{CWT}}(a, b)$ jsou dány korelačním integrálem mezi analyzovaným signálem $f(t)$ a bázovou funkcí $\left(\frac{1}{\sqrt{a}}\right) \psi(t)$, což je konkrétní vlnka. Mateřská vlnka $\psi(t)$ je stejná pro všechny vektory (a, b) , ale skutečný tvar konkrétní vlnky závisí na obou parametrech. Parametr a můžeme označit jako měřítko, způsobuje tedy dilataci – vlnka je roztažena a -krát, přičemž dělení \sqrt{a} zajišťuje zachování energie vlnky. Parametr b určuje časový posun (translaci), tj. posunutí vlnky na časové ose.

Funkce $\psi(t)$ musí splňovat podmínku přípustnosti [1]

$$\int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty, \quad (1.2)$$

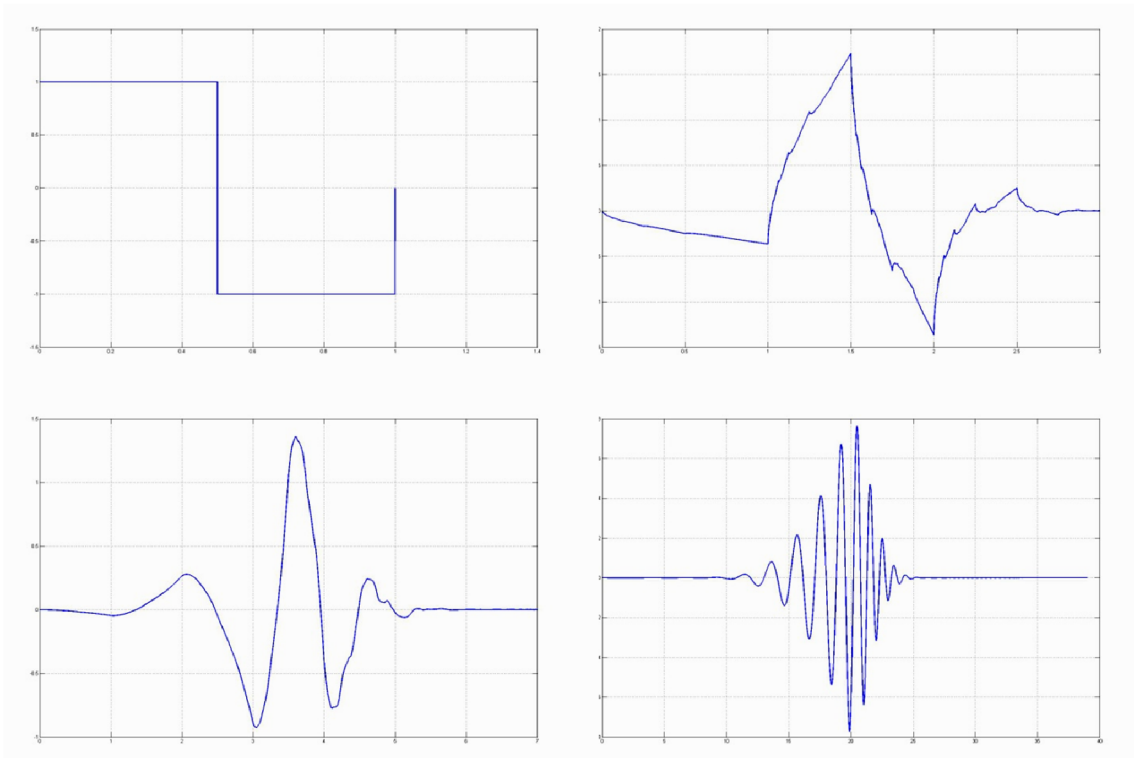
kde $\hat{\psi}(\omega)$ představuje integrální Fourierovu transformaci funkce $\psi(t)$ definovanou pro funkci $f(t)$ jako

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt. \quad (1.3)$$

Podmínka přípustnosti je postačující podmínkou pro existenci inverzní integrální waveletové transformace. Lze dokázat, že díky podmínce přípustnosti musí rovněž platit

$$\int_{-\infty}^{\infty} \psi(t) dt = 0, \quad (1.4)$$

tj. podmínka nulové střední hodnoty. Wavelety tedy nemají stejnosměrnou složku a mohou mít nenulovou hodnotu jen na konečném časovém intervalu, případně nenulové hodnoty mimo tento konečný interval musí být zanedbatelné. Pouze takové vlnky jsou wavelety s kompaktním nosičem. Ukázkou mateřských vlnek vidíme na obr. 1.1.



Obr. 1.1: Příklady mateřských waveletů podle Daubechiesové. Jde o wavelety řádu 1, 2, 4 a 20. (Db1, Db2, Db4, Db20)

1.2 Diskrétní waveletová transformace

Lze rozlišit tři druhy waveletových transformací [12]. Spojitou vlnkovou transformací, u níž jsou spojité jak změna měřítka a posunutí, tak i vstupní signál a vlnkové funkce, diskrétní vlnkovou transformací (Discrete Wavelet Transform, diskrétní waveletová transformace (DWT)), kdy zůstávají spojité vlnkové funkce a vstupní signál, ale změna měřítka a posuny jsou prováděny diskrétně a diskrétní vlnková transformace s diskrétním časem (Discrete Time Wavelet Transform, diskrétní waveletová transformace s diskrétním časem (DTWT)), u které vlnkové funkce i vstupní signál jsou diskrétní signály a změna měřítka a posuny jsou prováděny také diskrétně.

Při použití waveletové transformace k vyjádření obrazových dat pracujeme s konečným diskrétním signálem, a proto využijeme diskrétní waveletové transformace s diskrétním časem. Parametry a a b se mění diskrétně tak, že

$$a = 2^j \tag{1.5}$$

$$b = k \cdot 2^j = k \cdot a \tag{1.6}$$

pro $j, k \in \mathbb{Z}$ a $j \geq 1$.

Měřítka a je vzorkováno v dyadické posloupnosti, zatímco časový posuv rovnoměrně. Lze ukázat, že vzorky spektra v těchto bodech nesou úplnou informaci o původním signálu, který tak může být přesně rekonstruován [2].

Dyadická diskrétní waveletová transformace s diskrétním časem [12] je definovaná takto

$$S_{\text{DTWT}}(j, k) = \frac{1}{\sqrt{2^a}} \sum_{n=-\infty}^{\infty} f(n) \psi * (2^{-j}n - k). \tag{1.7}$$

DTWT je unitární transformace, a tudíž ji můžeme vyjádřit pomocí ortogonální matice \mathbf{W} řádu $n \times n$. Máme-li vektor délky n $y = (y_1, \dots, y_n)^T$, jeho waveletovou transformaci [8] $d = (d_1, \dots, d_n)^T$ získáme když

$$\mathbf{d} = \mathbf{W}y. \tag{1.8}$$

Díky tomu, že matice \mathbf{W} je ortogonální, inverzní diskrétní waveletová transformace může být vyjádřena jako

$$\mathbf{W}^{-1}\mathbf{d} = \mathbf{W}^T\mathbf{d} = y. \tag{1.9}$$

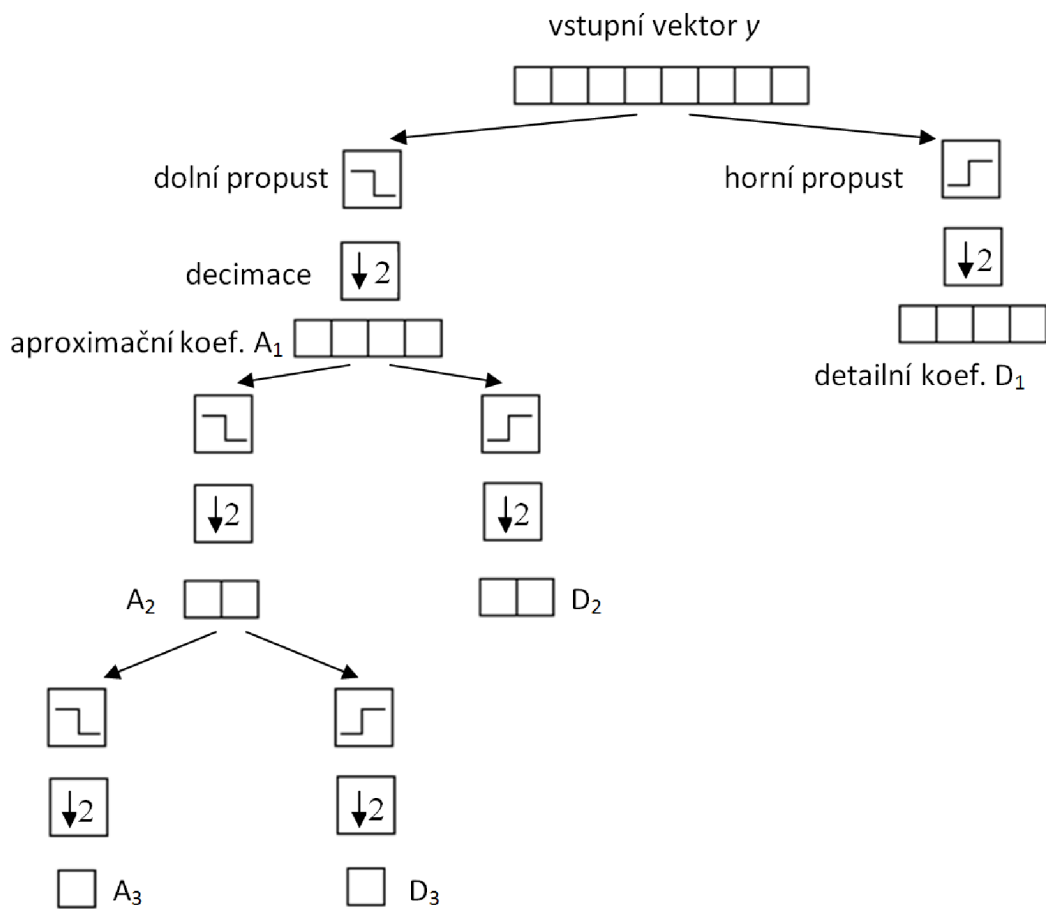
1.3 Algoritmus výpočtu diskrétní waveletové transformace s diskrétním časem

Vzhledem k výpočetní náročnosti násobení vstupního vektoru y délky n ortogonální maticí \mathbf{W} řádu $n \times n$ je v praxi výhodnější použít pyramidový rekurzivní algoritmus

S. Mallata [4], využívající k analýze a syntéze signálu banku filtrů s vlastností perfektní rekonstrukce – aliasing vzniklý při dekompozici se při rekonstrukci vyruší [15].

Filtrace se provádí pomocí čtyř filtrů – syntetizující dolní propustí (DP) a horní propustí (HP) a rekonstrukční DP a HP. Jde vždy o speciální kvadraturní zrcadlové filtry, jejichž moduly kmitočtových charakteristik jsou navzájem zrcadlově obrácené [12] podle $f/f_{vz} = 1/4$.

Vektor vstupu y filtrujeme pomocí filtru typu dolní propust a zároveň filtrem typu horní propust. Obě výsledné posloupnosti decimujeme (vypustíme každý druhý vzorek), a tudíž získáváme 2 nové posloupnosti s přibližně poloviční velikostí původního vstupního vektoru y [14], viz obr. 1.2.



Obr. 1.2: Waveletová dekompozice signálu y pomocí algoritmu S. Mallata o hloubce 3.

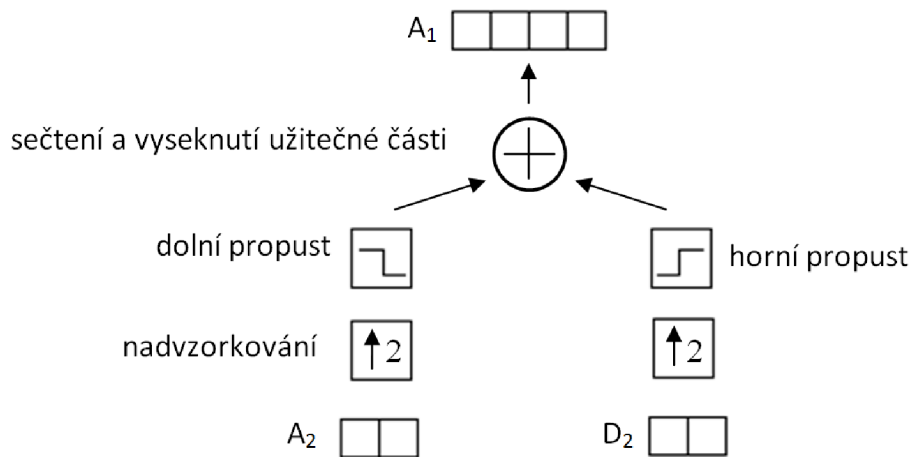
Koeficienty získané filtrem typu horní propust nazýváme detailními waveletovými koeficienty; aproximačními waveletovými koeficienty nazýváme koeficienty získané filtrem dolní propust. Nyní jsme provedli jeden krok waveletové dekompozice. Další krok provedeme obdobně pouze s rozdílem, že namísto vstupního vektoru

y přivádíme na vstup obou filtrů vektor aproximačních koeficientů s tím, že vektor detailních koeficientů zachováme. Počet kroků nazýváme hloubkou dekompozice. Z předchozího vyplývá, že maximální hloubka dekompozice pro osmičlenný vstupní vektor je 3. Obecně pro vstupní vektor s délkou n platí [8], že maximální hloubka dekompozice je

$$J_{\max} \leq \log_2 n. \quad (1.10)$$

Mallatovým algoritmem můžeme vstupní signál rozložit na detailní a aproximační koeficienty, přičemž nedochází k žádné ztrátě informace. Proces dekompozice je možné zastavit v kterékoli vhodné úrovni, např. k (a je tak umožněna tzv. neúplná diskretní waveletová transformace), což je další výhodou realizace waveletové transformace pomocí zrcadlově kvadraturních filtrů.

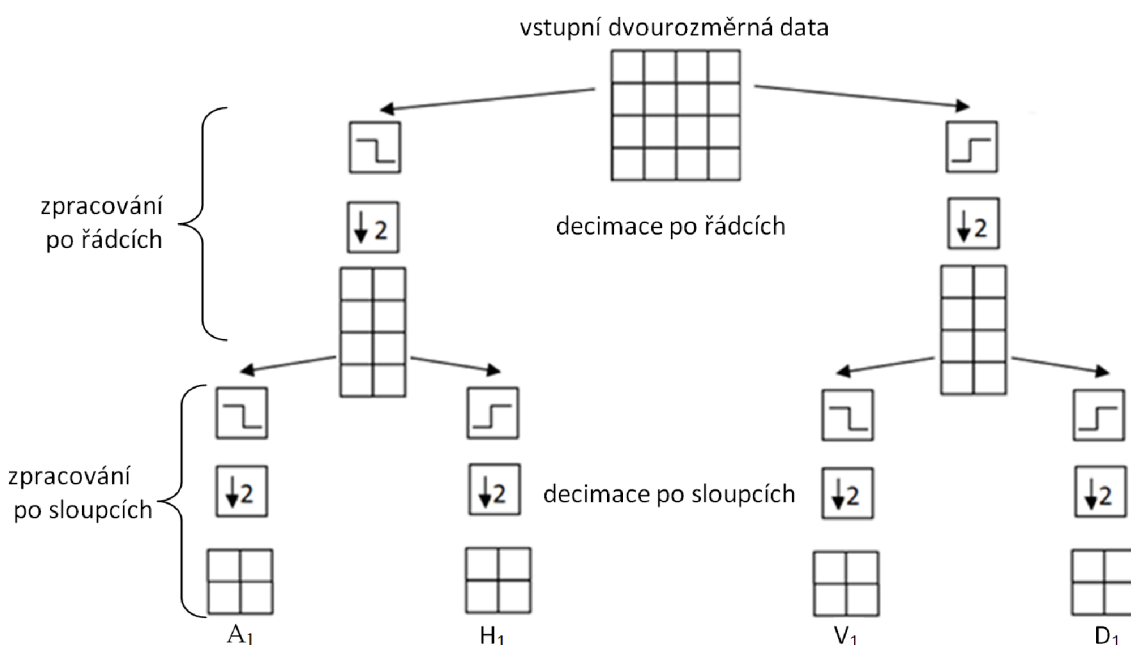
Signál je pak popsán svým aproximačním vektorem A_k a všemi vektory detailů D_k až D_1 , a není tedy problém z vypočtené dekompozice vytvořit zpět původní signál postupem přesně opačným. Ukážeme, že na kterékoli úrovni dekompozice k lze získat aproximační koeficienty vyšší úrovně $k-1$ z koeficientů A_k a D_k . Oba vektory jsou nejprve nadzvorkovány – čili do vektorů aproximačních a detailních koeficientů vkládáme nuly a zvyšujeme tedy jejich délku na dvojnásobek. Následně filtrujeme filtry horní a dolní propust, které jsou inverzní k původním dekompozičním filtrům, načerž výsledky sečteme a vysekne pouze potřebnou prostřední část, viz obr. 1.3. Postup opakujeme tolikrát, kolik je hloubka dekompozice, až dostaneme originální signál.



Obr. 1.3: Jeden krok waveletové rekonstrukce. Aproximační a detailní koeficienty jsou nadzvorkovány, filtrovány přes dolní a horní propust (inverzní k původním dekompozičním filtrům) a následně sečteny. K získání původních aproximačních koeficientů vyšší úrovně je již třeba jen vyseknout užitečnou část součtu.

1.4 Dvourozměrná waveletová transformace

Cílem této práce jsou možnosti zpracování obrazu a obraz jako takový je vektor dvojrůzoměrný. Proto musíme použít dvourozměrnou diskretní waveletovou transformaci s diskretním časem [3, 5]. Pro dekompozici použijeme podobný rekurzivní pyramidový algoritmus jako výše, data ale budeme filtrovat zvlášť po řádcích a zvlášť po sloupcích. Je zřejmé, že jedním krokem algoritmu se vzniknou čtyři druhy koeficientů. Situaci ilustruje obr. 1.4.



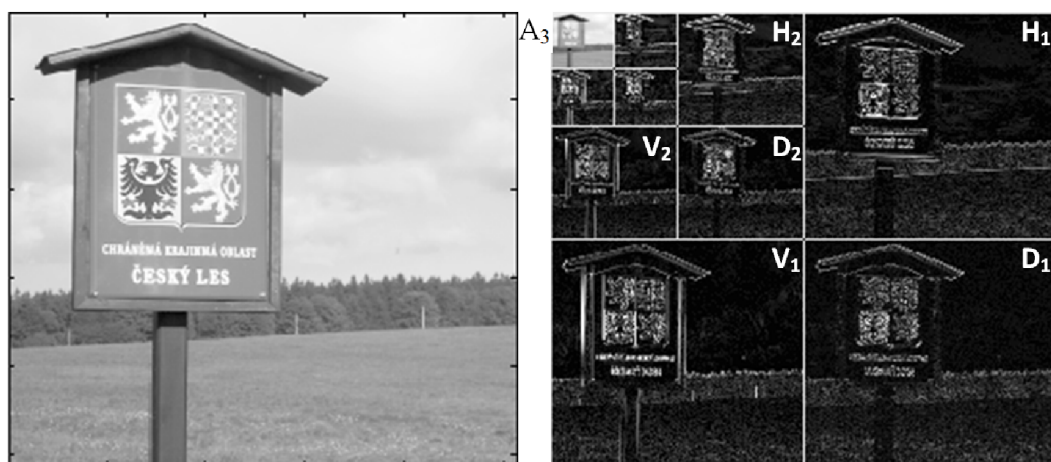
Obr. 1.4: Jeden krok dvourozměrné waveletové dekompozice. Vznikají čtyři druhy koeficientů, každý se čtvrtinou prvků vzhledem ke vstupním datům.

Obrazová data na vstupu projdou filtry DP a HP, načež jsou decimovány, tj. při zpracování po řádcích je vynechán každý druhý prvek v řádku. Výsledek opět prochází filtry DP a HP, tentokrát se ale zpracovává po sloupcích, čímž po decimaci vypadává polovina prvků ve sloupcích. Z jednoho kroku dekompozice vychází koeficienty A_1 – aproximační, H_1 – horizontální, V_1 – vertikální, D_1 – diagonální. Další stupeň dekompozice bychom prováděli obdobně, jen místo vstupního obrázku bychom použili na vstupu aproximační koeficienty posledního spočteného kroku. Horizontální, vertikální a diagonální koeficienty uchováváme.

Na obr. 1.5 vidíme praktickou ukázkou dekompozice obrazu do hloubky 3. V nejvyšší úrovni se nachází signál A_3 , jenž byl třikrát filtrován filtry typu dolní propust. Z toho vyplývá, že jde o signálovou složku obsahující hlavně hodnoty nízkých frekvencí z původního signálu, nese tedy největší energii, čili informaci. Tak jako

u všech signálů, nízkofrekvenční složky nesou základní informace o signálu. Vysokofrekvenční složky signálu poté uchovávají detailní informaci signálu, což mohou být u obrazových dat např. informace o detailech a hranách v obrazu.

Rekonstrukci obrazu z waveletové dekompozice je možné provést inverzním postupem vzhledem k dekompozici. Decimace je nahrazena interpolací, tedy vložením nuly na místo každého původně vynechaného prvku v řádku nebo ve sloupci. Místo dekompozičních filtrů DP a HP přicházejí filtry rekonstrukční, které jsou k původním inverzní. Následuje součet a vyseknutí střední části signálu.



Obr. 1.5: Praktická ukázka rozložení obrazu pomocí dvourozměrné diskrétní waveletové transformace. Vlevo je originální obrázek, vpravo jeho dekompozice do hloubky 3. Koeficienty třetí úrovně rozkladu mimo A_3 nejsou z důvodu přehlednosti vyznačeny. Provedeno v programu MATLAB.

2 ZPRACOVÁNÍ OKRAJŮ SIGNÁLU

Základem waveletové transformace prováděné pomocí Mallatova pyramidového algoritmu je konvoluce signálu délky n s filtrem délky m . Výsledek konvoluce má pak délku $m + n - 1$. Problém ale nastává při použití DTWT, kdy pracujeme s konečným signálem, ale algoritmus je navržen primárně pro práci se signálem nekonečné délky [10]. To způsobuje, že okraje signálu nemusí být zpracovány správně. Aby bylo možné toto minimalizovat, bylo by nutné znát vždy ještě určitý počet dalších vzorků za okrajem signálu. Protože to ale není povětšinou možné (ovšem v SegDTWT u vnitřních bloků používáme prodloužení vlastním signálem z předchozího bloku, viz kapitola 4), nezbyvá než nějakým způsobem hodnoty za okrajem extrapolovat. K tomu existuje několik metod [15, 6], které zobrazuje obr. 2.2.

Velkou skupinou metod je doplnění signálu na základě odhadu pomocí matematických funkcí. Souhrnnou nevýhodou těchto přístupů je, že se v každém kroku waveletové dekompozice vytvoří o $m - 2$ více koeficientů než kolik je vstupních vzorků signálu. To je ale vyváжено schopností perfektní rekonstrukce [10].

Doplnění nulami je nejjednodušším odhadem – za hranicemi vstupního signálu mají vzorky nulovou hodnotu, což samozřejmě vnáší do signálu nespojitosti.

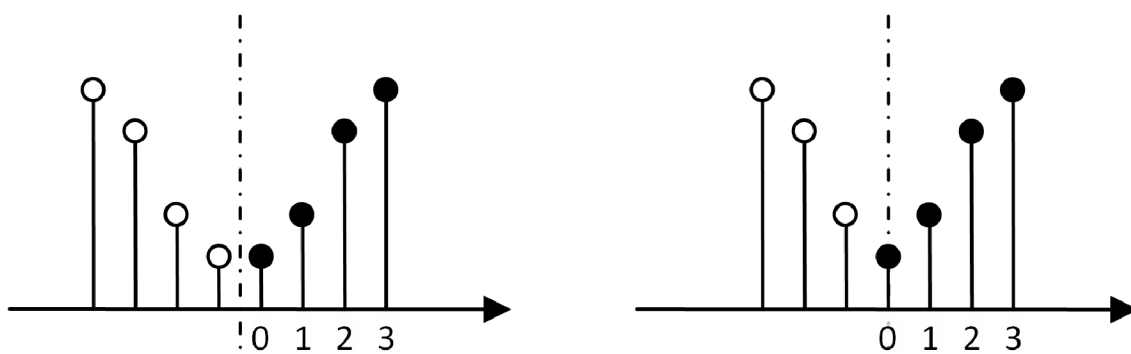
Symetrické prodloužení signálu je založeno na osově souměrnosti. Signál je rozšířen několika vzorky symetricky vůči ose, kterou tvoří začátek či konec signálu. Tato metoda je hojně používána při zpracování obrazu, přestože přidává nespojitost první derivace signálu.

Existují 2 typy symetrického prodloužení [13]: půlbodové (half-point – H) a celobodové (whole-point – W). Liší se přístupem k „zrcadlení“, kdy celobodové prodloužení zachovává rovnost $x(t) = x(-t)$ ve spojitém čase, což znamená že hodnota v $x(0)$ není jako jediná opakována. U půlbodového prodloužení je tomu naopak. Hodnota $x(0)$ opakována je, a tudíž platí $x(-1) = x(0)$. Osa symetrie prochází bodem $x = -\frac{1}{2}$ narozdíl od celobodového prodloužení, kde prochází $x = 0$. Obr. 2.1 jednoduše znázorňuje obě možnosti.

U zpracování waveletové transformace je třeba vybrat půlbodové prodloužení při použití filtru se sudou délkou a celobodové při použití filtru s lichou délkou.

Prodloužení polynomem extrapoluje průběh signálu za okrajem pomocí polynomu k -tého stupně a zachovává spojitost k -té derivace. Prodloužení polynomem multého stupně znamená doplnění konstantou, čehož vyplývá, že i doplnění nulami je speciálním případem tohoto prodloužení.

Periodické rozšíření signálu je známé i z Fourierovy analýzy a již nespadá do předchozí skupiny metod. Uvažujeme, že zpracováváný signál je jednou periodou nekonečného periodického signálu, a tudíž ho musíme na obou stranách rozšířit o vzorky signálu z opačného konce. Výhodou této metody je, že pokud je délka



Obr. 2.1: Půlbodové (vlevo) a celobodové (vpravo) symetrické prodloužení - čerchovaná čára symbolizuje osu symetrie.



Obr. 2.2: Testovací obraz a jeho prodloužení na okrajích v programu MATLAB: a) doplnění nulami, b) symetrické prodloužení, c) prodloužení polynomem, d) periodizace

signálu sudá, je počet koeficientů vzniklých jedním krokem DTWT stejný, jako počet vzorků původního signálu. Jako nevýhoda se naopak objevuje fakt, že metoda vnáší do signálu body nespojitosti.

Další možností je použití okrajových filtrů k prodloužení signálu. Pro úplnost je nutné ještě uvést, že mezi důležité postupy patří ještě useknutí, při kterém jsou zahozeny okrajové waveletové koeficienty, aby jejich počet korespondoval s počtem vzorků signálu. Tato metoda neumožňuje perfektní rekonstrukci. Některé ze zmíněných metod názorně ukazuje obr. 2.2.

3 SEGMENTOVANÁ WAVELETOVÁ TRANSFORMACE

V této části je probírán algoritmus segmentované waveletové transformace [8], který vytvořil Mgr. Pavel Rajmic, Ph.D. Dále bude uvedena úprava algoritmu pro dvojrozměrné signály a to jak ve variantě úplného prodloužení okrajů signálu před vlastní transformací, tak i pro prodlužování v jednotlivých krocích waveletové dekompozice. Nechybí ani popis výpočtu dvourozměrné inverzní waveletové transformace.

3.1 Segmentace a prodlužování signálu

Základní myšlenou je, že vstupní signál je rozdělen na určitý počet segmentů délky s a každý z těchto segmentů je zpracováván zcela samostatně. Po zpracování všech částí dojde opět ke spojení waveletových koeficientů z jednotlivých segmentů, což umožní mít kompletní signál nyní ale ve waveletové reprezentaci. Rozdíl oproti klasické DTWT je tedy pouze v procesu výpočtu, nikoli ve výsledku.

Aby waveletové koeficienty DTWT a SegDTWT souhlasily, je třeba zacházet s každým segmentem, jako s celým signálem – je nutné jeho prodloužení. Bylo by ovšem chybou pokoušet se extrapolovat vzorky za hranicí segmentu, když je můžeme přímo vzít ze segmentu sousedního. Pouze okrajové segmenty prodloužíme jednou z metod probraných v kapitole 3.

Celková délka prodloužení v k -té úrovni dekompozice je definována jako

$$r(k) = (2^k - 1)(m - 1), \quad (3.1)$$

kde m je délka dekompozičního waveletového filtru [11]. Vzorec 3.1 vyhází z faktu, že pro výpočet jednoho waveletového koeficientu je třeba v k -té úrovni dekompozice

$$p(k) = (2^k - 1)(m - 1) + 1 \quad (3.2)$$

vzorků původního signálu, což vyplývá z nutné konvoluce signálu s waveletovým filtrem, kdy potřebujeme právě m signálových vzorků k získání jednoho waveletového koeficientu první úrovně dekompozice. Například abychom dostali první koeficient, musíme na okraj signálu přidat $p(k) - 1$ extrapolovaných vzorků; to odpovídá délce prodloužení $r(k)$.

U vnitřních segmentů můžeme rozdělit celkové prodloužení na dvě části. Na pravé, kdy přidáváme vzorky k pravému okraji segmentu, a levé, u kterého se segment prodlužuje doleva. Jejich délka nemusí být stejná a nabízí se více možností jak s jejich délkou pracovat. V této práci vycházíme z původního návrhu pro zpracování signálu v reálném čase, kde byla snaha o maximalizaci levého prodloužení

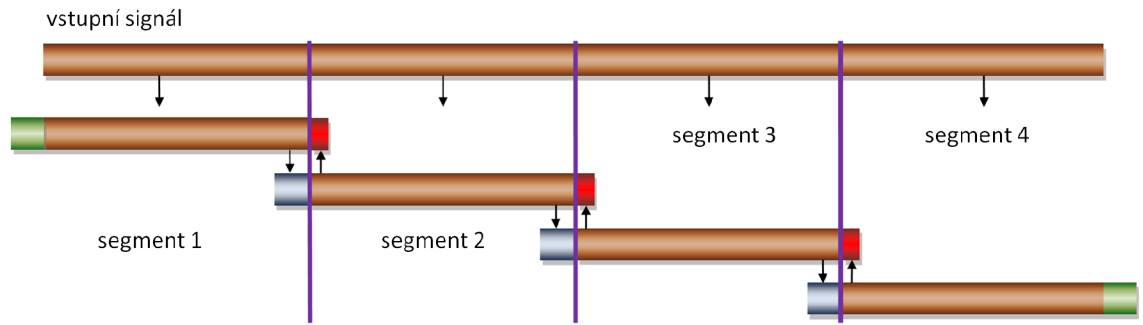
(berou se vzorky z již známého segmentu) a minimalizace pravého prodloužení, ke kterému jsou třeba vzorky dalšího, ještě neznámého segmentu. Minimální možné pravé prodloužení n -tého segmentu se spočítá jako

$$R_{\min}(n) = 2^J \left\lceil \frac{ns}{2^J} \right\rceil - ns, \quad (3.3)$$

kde J je požadovaná (maximální) hloubka dekompozice a $\lceil x \rceil$ značí zaokrouhlení x na nejbližší celé číslo nahoru. Maximální možné levé prodloužení následujícího $(n+1)$ -ního segmentu je pak

$$L_{\max}(n+1) = r(J) - R_{\min}(n). \quad (3.4)$$

Ze vzorce 3.3 jasně vyplývá, že pokud je délka segmentu s dělitelná 2^J , $R_{\min} = 0$ pro všechny segmenty. Pokud tedy zvolíme vhodnou délku segmentu, stačí vnitřní segmenty prodloužit pouze zleva o $r(J)$ vzorků. Rozdělení signálu na segmenty a jejich prodloužení ilustruje obr. 3.1 a délky prodloužení pro různé délky segmentů ukazuje tab. 3.1.



Obr. 3.1: Segmentace jednorozměrného signálu a prodloužení segmentů. Modrá barva značí prodloužení vzorky předchozího segmentu zleva, červená prodloužení vzorky následujícího segmentu zprava. Zelené je prodloužení okrajů některou z metod zmíněných v kapitole 3.

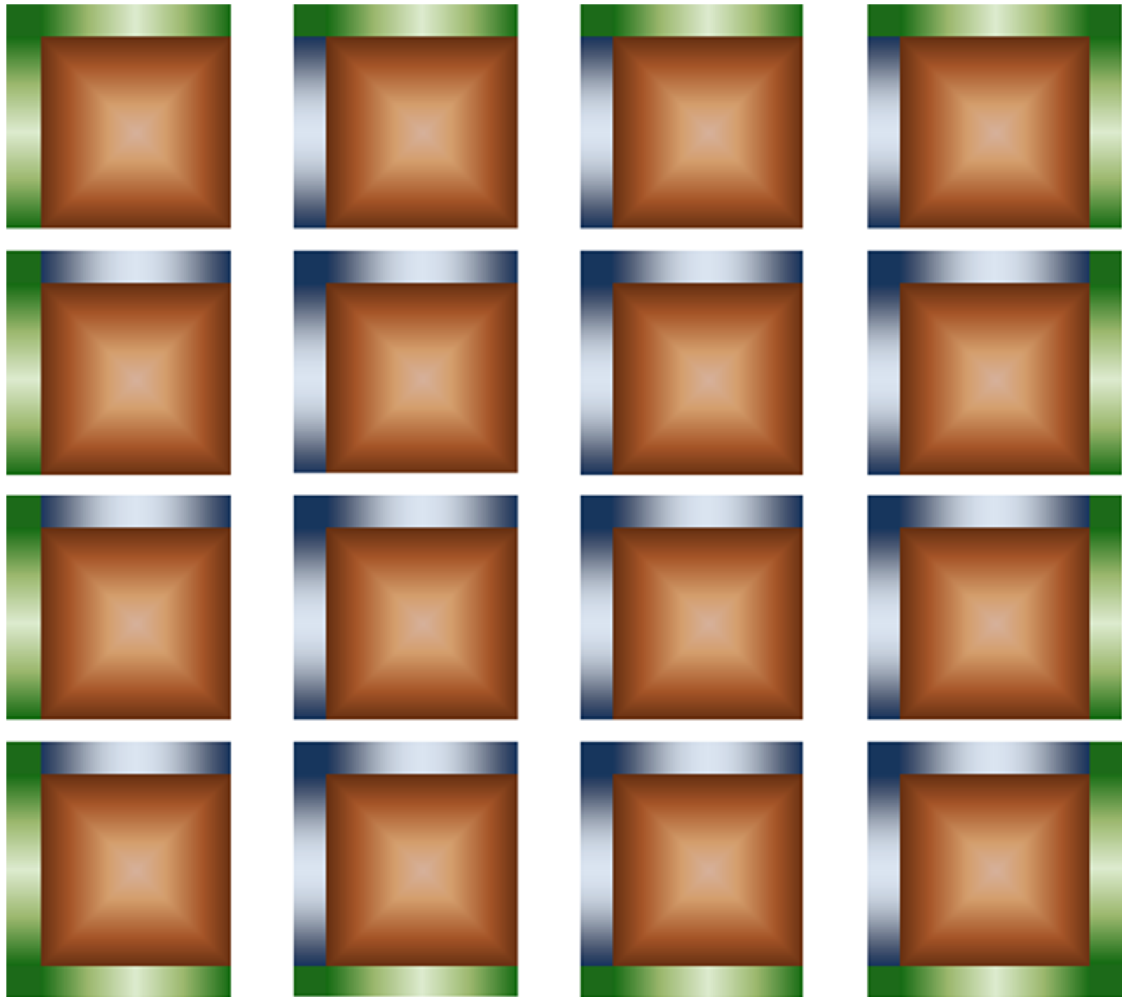
3.2 Dvourozměrná segmentovaná waveletová transformace

Protože byl algoritmus výpočtu segmentované waveletové transformace důkladně popsán v [8], v této části bude ukázána již jeho úprava pro dvojrozměrný signál. V rámci zjednodušení budeme uvažovat, že oba dva rozměry signálu jsou stejné a jedná se tedy o čtvercový obraz. Díky tomu mohou být i segmenty čtvercové.

Tab. 3.1: Závislost délky levého a pravého prodloužení na délce segmentu. Hloubka dekompozice je 3 a délka filtru 8.

délka segm.		pořadí segmentu											
		1	2	3	4	5	6	7	8	9	10	11	12
256	L_{\max}	49	49	49	49	49	49	49	49	49	49	49	49
	R_{\min}	0	0	0	0	0	0	0	0	0	0	0	0
257	L_{\max}	49	42	43	44	45	46	47	48	49	42	43	44
	R_{\min}	7	6	5	4	3	2	1	0	7	6	5	4
258	L_{\max}	49	43	45	47	49	43	45	47	49	43	45	47
	R_{\min}	6	4	2	0	6	4	2	0	6	4	2	0
259	L_{\max}	49	44	47	42	45	48	43	46	49	44	47	42
	R_{\min}	5	2	7	4	1	6	3	0	5	2	7	4
260	L_{\max}	49	45	49	45	49	45	49	45	49	45	49	45
	R_{\min}	4	0	4	0	4	0	4	0	4	0	4	0
261	L_{\max}	49	46	43	48	45	42	47	44	49	46	43	48
	R_{\min}	3	6	1	4	7	2	5	0	3	6	1	4
262	L_{\max}	49	47	45	43	49	47	45	43	49	47	45	43
	R_{\min}	2	4	6	0	2	4	6	0	2	4	6	0
263	L_{\max}	49	48	47	46	45	44	43	42	49	48	47	46
	R_{\min}	1	2	3	4	5	6	7	0	1	2	3	4
264	L_{\max}	49	49	49	49	49	49	49	49	49	49	49	49
	R_{\min}	0	0	0	0	0	0	0	0	0	0	0	0
265	L_{\max}	49	42	43	44	45	46	47	48	49	42	43	44
	R_{\min}	7	6	5	4	3	2	1	0	7	6	5	4
266	L_{\max}	49	43	45	47	49	43	45	47	49	43	45	47
	R_{\min}	6	4	2	0	6	4	2	0	6	4	2	0
267	L_{\max}	49	44	47	42	45	48	43	46	49	44	47	42
	R_{\min}	5	2	7	4	1	6	3	0	5	2	7	4
268	L_{\max}	49	45	49	45	49	45	49	45	49	45	49	45
	R_{\min}	4	0	4	0	4	0	4	0	4	0	4	0

Krajní segmenty je samozřejmě třeba prodloužit dle zvoleného způsobu prodlužování do všech stran, jak ukazuje obr. 3.2.



Obr. 3.2: Segmentace a prodloužení dvourozměrného signálu. Hnědou barvou je vyznačen původní obraz rozdělený do 16-ti segmentů. Zelená označuje prodloužení na okraji signálu dle kapitoly 3 a modrá barva symbolizuje prodloužení pomocí vzorků sousedního segmentu. Pro lepší orientaci je zvolen speciální případ, kdy délka segmentu je dělitelná 2^J , a tudíž jsou všechna pravá a dolní prodloužení rovna nule.

Oproti jednorozměrnému případu se liší především v prodlužování, které je z důvodu separability waveletové transformace (zpracování nejprve po řádcích, pak po sloupcích) nutno provádět nejen na levém a pravém okraji, ale i na okraji horním a dolním. Zavádíme tedy horní prodloužení $H_{\max} = L_{\max}$ a dolní prodloužení $D_{\min} = R_{\min}$. Rovnosti platí pouze pro segmenty obrazu se stejnou vzdáleností od okrajů, jinak je třeba vždy počítat, jak daleko je segment od horního či levého okraje pro vertikální či horizontální prodloužení. Například druhý segment ve třetím řádku

má H_{\max} a D_{\min} počítáno jako v pořadí třetí segment a L_{\max} a R_{\min} jako druhý v pořadí.

Nyní podrobně popíšeme zacházení se segmenty. Po načtení segmentu, pokud jde o okrajový segment, jej musíme na těchto okrajích prodloužit o $r(J)$ vzorků podle zvolené metody prodlužování. Pokud jde o vnitřní segment, je na levé a horní straně prodloužen o L_{\max} a H_{\max} , případně i na pravé a dolní straně o R_{\min} a D_{\min} , vzorky ze sousedních segmentů. U některých bloků dochází i ke kombinaci, kdy na jedné straně je prodloužen extrapolovanými vzorky a na jiné vzorky ze sousedního segmentu; vždy záleží na původním umístění v obrazu. Například poslední segment (tj. pravý dolní roh obrazu) je prodloužen nahoře a vlevo vzorky sousedních segmentů a zároveň na jeho pravé a dolní straně dochází k prodloužení na dle zvolené metody prodlužování.

Dále provedeme dekompozici konkrétního segmentu Mallatovým pyramidovým algoritmem (bez jakéhokoli dalšího prodlužování). Výsledkem nám je sada waveletových koeficientů, ovšem některé ještě musíme upravit. Jde o všechny horizontální, vertikální a diagonální koeficienty mimo těch z posledního dekompozičního kroku. Je třeba vypočítat délku prodloužení $r(J - k)$ v konkrétní úrovni dekompozice k podle vzorce 3.1 a o ní zkrátit waveletové koeficienty zleva a shora. Jestliže byly prodlouženy okrajové segmenty vpravo a dole je nutné je zkrátit také.

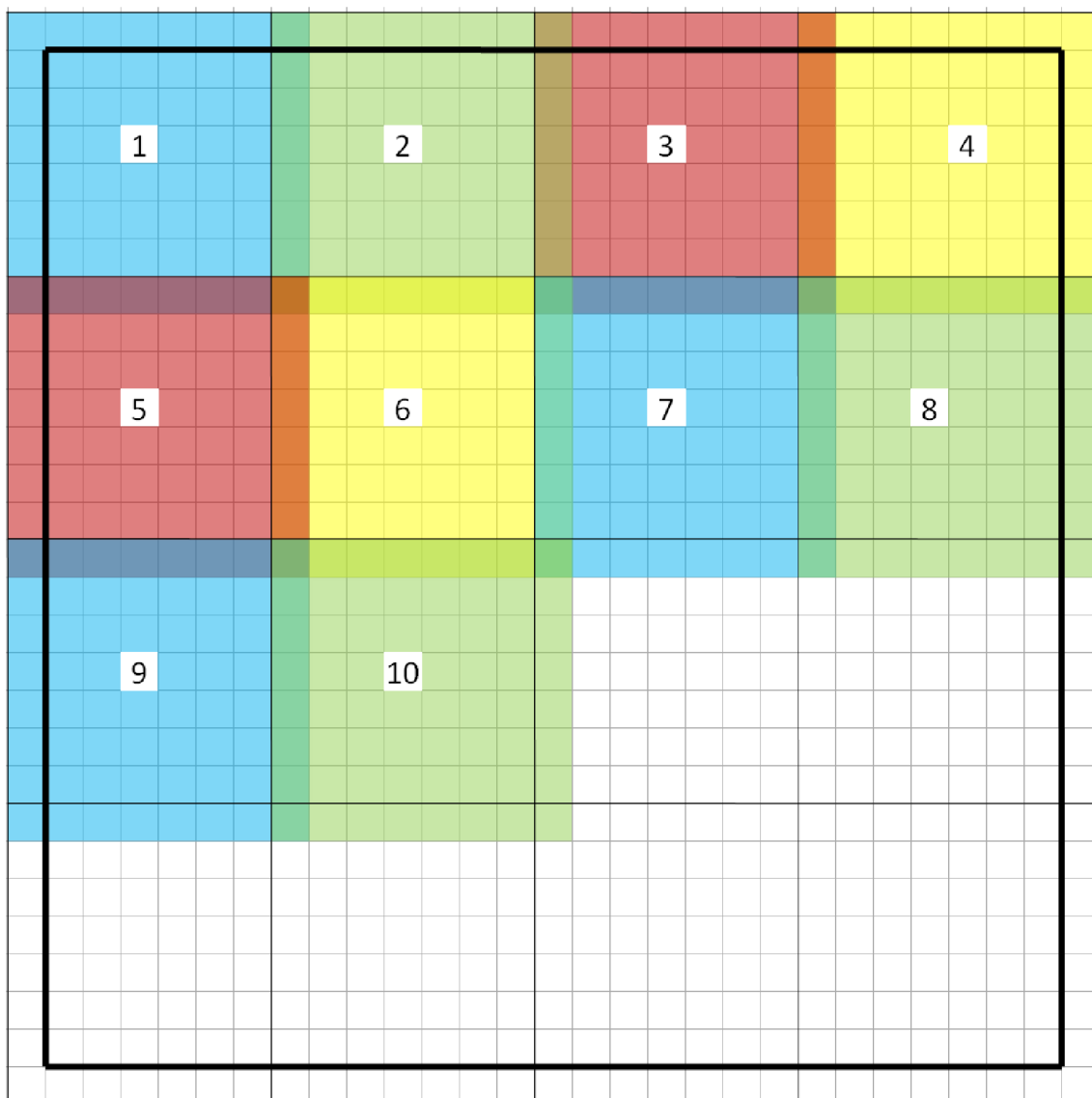
Waveletové koeficienty konkrétního segmentu vložíme na správné místo do matic koeficientů celého obrazu a pokračujeme dalším segmentem, dokud nejsou zpracovány všechny. Podrobný popis algoritmu je uveden v kapitole 4.

V případě, že nebudeme zařazovat úplné prodloužení před vlastní transformaci, je nutné u okrajových segmentů v každém kroku dekompozice provést zvolené prodloužení na příslušném okraji o $m - 1$ vzorků.

3.3 Inverzní segmentovaná waveletová transformace

Výstupem algoritmu segmentované waveletové transformace je sada waveletových koeficientů, které lze dle potřeby upravovat. Pokud jsme každý blok waveletových koeficientů vložili na správné místo v maticích koeficientů celého obrazu, je nutné tyto matice opět rozdělit na segmenty. Jestliže cílem výpočtu SegDTWT je pouze nějaká úprava obrazu (např. odšumování) je jednodušší waveletové koeficienty každého jednotlivého segmentu ihned zpracovat a použít jako vstup pro inverzní segmentovanou waveletovou transformaci. Dojde tak k vypuštění kroku skládání a řazení bloků na správné místo v maticích koeficientů celého obrazu a po úpravě jejich opětovné dělení na segmenty.

Všechny vstupní detailní koeficienty (horizontální, vertikální a diagonální) mimo těch v maximální hloubce dekompozice musíme nejprve prodloužit zleva a shora o $r(J - k)$ nulových vzorků – tedy nezávisle na typu prodlužování zvoleném při dekompozici signálu. Dále provedeme standardní waveletovou rekonstrukci podle algoritmu S. Mallata pouze vynecháme jakékoliv ořezávání po provedeném sečtení [9]. To způsobí, že výsledný segment obrazu má větší rozměry, než měl ten původní. To ale řeší způsob skládání segmentů, které ukazuje obr. 3.3 a bude podrobněji popsáno dále.



Obr. 3.3: Průběh skládání rekonstruovaných segmentů při SegIDTWT do původního obrazu. Ty jsou postupně přidávány v pořadí dle čísel – poslední byl vložen segment číslo 10. Jsou vidět místa překryvů, kde se hodnoty pixelů sčítají. Pro lepší orientaci byla přidána mřížka a tlustou čarou je vyznačen závěrečný ořez okrajů.

Je tedy nutné při vkládání rekonstruovaného segmentu na správné místo v původním obrazu dodržet přesně místo, odkud byl původně vyříznut, z čehož vyplývá, že se segmenty budou překrývat. V místě překryvu je nutné hodnoty vzorků více segmentů sečíst dohromady. Podobným přístupem se stejným výsledkem je použití metody přičtení přesahu vždy ke každému bloku rekonstruovaných vzorků. Na závěr je třeba výsledný obraz na stranách oříznout.

Výsledkem je složený obraz, který bude mít v případě, že nebyly prováděny úpravy waveletových koeficientů, pouze zanedbatelnou odchylku od původního obrazu vstupujícího do algoritmu SegDTWT. Podrobný popis algoritmu bude následovat v další kapitole věnující se popisu práce programu, který byl jako součást této práce vytvořen.

4 IMPLEMENTACE ALGORITMŮ

Nyní se dostáváme k praktické realizaci uvedených dvoudimenzionálních algoritmů SegDTWT a SegIDTWT, která je provedena v prostředí MATLAB. To je již díky integrovanému Wavelet Toolboxu připraveno na práci s waveletovou transformací, a umožní tak jednoduše ověřit funkčnost nového algoritmu.

Jako součást této práce byl vytvořen program `segdtwt.m`, který díky grafickému uživatelskému rozhraní GUI nabízí mimo jiné přehledné porovnání výsledků SegDTWT a standardní DTWT u obrazových dat. Nejdůležitější částí programu je skript `segmentation.m`, který provádí zmíněné algoritmy a jejich porovnání. Další části se budou týkat popisu práce toho skriptu.

4.1 Upravený Mallatův pyramidový algoritmus

Při popisech SegDTWT se odvoláváme na Mallatův algoritmus [4], který bylo nutné z důvodu zahrnutí úplného prodloužení před vlastní transformací lehce upravit, a to vypuštěním právě kroku prodlužování.

1. Načteme vstupní signál.
2. Provedeme konvoluci vstupního signálu s waveletovými filtry DP nebo HP po řádcích.
3. Vyskneme střední část konvoluce odebráme $m - 1$ vzorků vlevo a vpravo.
4. Výsledky upravíme pomocí sudé decimace po řádcích.
5. Filtrujeme signály waveletovými DP nebo HP po sloupcích
6. Vyskneme střední část konvoluce odebráme $m - 1$ vzorků nahoře a dole.
7. Výsledky upravíme pomocí sudé decimace po sloupcích.
8. Detailní koeficienty uložíme, aproximační přivedeme na vstup dalšího cyklu začínajícího bodem 2, dokud není dosažena požadovaná hloubka dekompozice.

4.2 Algoritmus SegDTWT

Pro algoritmus jsou dány tyto vstupní parametry: délka segmentu s , dekompoziční waveletové filtry, maximální hloubka dekompozice J , typ prodloužení na okrajích. Z nich je vypočtena maximální délka prodloužení $r(J)$ podle vzorce 3.1, počet segmentů v řádku či sloupci a délka posledního segmentu. Ta v případě, že rozměry obrazu nejsou dělitelné délkou segmentu, bude nižší než daný rozměr segmentu. Pro zjednodušení je skript naprogramován pro zpracování pouze čtvercových obrazů,

a tudíž i segmenty mají tvar čtverce. Délkou segmentu tedy rozumíme výšku nebo šířku bloku.

Zpracovávány jsou pouze černobílé obrazy, pokud je na vstupu obraz barevný, je automaticky převeden na stupně šedé. Kdybychom chtěli zpracovávat obrazy v barvách, bylo by nutné provádět SegDTWT pro každou barevnou složku – dle zvoleného barevného modelu – zvlášť.

V programu jsou implementovány dva přístupy k prodlužování signálu na okrajích. První využívá úplné prodloužení před vlastní transformací (podobně jako u prodloužení segmentů sousedními vzorky pro neokrajové segmenty) a druhý provádí prodloužení na okrajích vždy v každém kroku dekompozice tak, jak je to použito i u standardních funkcí v MATLABu.

Algoritmus SegDTWT s úplným prodloužením před transformací

1. Načteme segment obrazu.
2. Vypočteme pravé a dolní prodloužení R_{\min} a D_{\min} podle vzorce 3.3.
3. Pokud jde o okrajový segment, prodloužíme ho na jednom nebo více okrajích o $r(J)$ vzorků podle typu prodloužení.
4. Na stranách, které nejsou na okraji původního obrazu, prodloužíme segment o L_{\max} , H_{\max} , R_{\min} a D_{\min} vzorků sousedních segmentů.
5. Podle upraveného Mallatova algoritmu vypočítáme waveletové koeficienty.
6. Provedeme ořezání koeficientů podle polohy segmentu v původním obrazu následovně: pravý dolní roh – odstraníme na všech stranách $r(J - k)$ koeficientů, kde k je aktuální hloubka dekompozice; dolní okraj – odstraníme na všech stranách $r(J - k)$ koeficientů mimo pravého okraje; pravý okraj – odstraníme na všech stranách $r(J - k)$ koeficientů mimo dolního okraje; ostatní – odstraníme na $r(J - k)$ koeficientů vlevo a nahoře.
7. Vložíme koeficienty na správné místo v maticích koeficientů celého obrazu.
8. Vypočteme levé a horní prodloužení L_{\max} a H_{\max} podle vzorce 3.4 pro následující segment.
9. Opakujeme pro další segment od bodu 1, dokud nemáme zpracovány všechny segmenty obrazu.

Algoritmus SegDTWT s prodlužováním v každém kroku transformace

1. Načteme segment obrazu.
2. Vypočteme pravé a dolní prodloužení R_{\min} a D_{\min} podle vzorce 3.3.
3. Na stranách, které nejsou na okraji původního obrazu, prodloužíme segment o L_{\max} , H_{\max} , R_{\min} a D_{\min} vzorků sousedních segmentů.
4. Vypočítáme waveletové koeficienty podle upraveného Mallatova pyramidového algoritmu. Pokud ale jde o okrajový segment, provedeme před krokem

2. v upraveném Mallatově algoritmu v každé úrovni dekompozice prodloužení o $m - 1$ vzorků na okraji segmentu dle zvoleného způsobu prodlužování.
5. Ořízneme vypočtené koeficienty podle polohy segmentu v původním obrazu následovně: levý horní roh – bez úprav; horní okraj – odstraníme zleva $r(J - k)$ koeficientů, kde k je aktuální hloubka dekompozice; levý okraj – odstraníme shora $r(J - k)$ koeficientů, ostatní – odstraníme na $r(J - k)$ koeficientů vlevo a nahoře.
6. Vložíme koeficienty na správné místo v maticích koeficientů celého obrazu.
7. Vypočteme levé a horní prodloužení L_{\max} a H_{\max} podle vzorce 3.4 pro následující segment.
8. Opakujeme pro další segment od bodu 1, dokud nemáme zpracovány všechny segmenty obrazu.

4.3 Algoritmus SegIDTWT

Segmentovaná inverzní waveletová transformace je nezávislá na způsobu získání waveletových koeficientů, což nám umožňuje použít pouze jeden algoritmus pro rekonstrukci obrazu. Ve skriptu `segmentation.m` je inverzní transformace aplikována ihned po výpočtu a případné úpravě waveletových koeficientů daného segmentu, což umožňuje vynechání kroku vyřezávání segmentů z matic waveletových koeficientů.

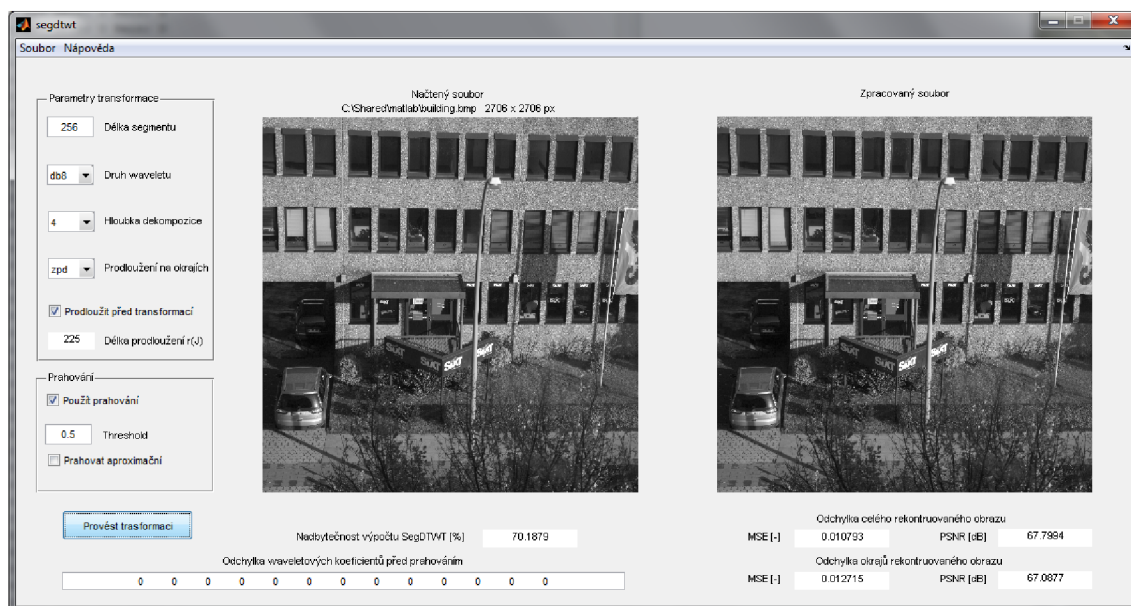
Algoritmus SegIDTWT

1. Načteme sadu waveletových koeficientů pro jeden segment obrazu
2. Prodloužíme horizontální, vertikální a diagonální waveletové koeficienty o $r(J - k)$ nulových vzorků shora a zleva, kde k je aktuální hloubka dekompozice.
3. Upravíme velikost bloku aproximačních koeficientů tak, aby odpovídala velikosti detailních koeficientů (v některých případech se rozměry liší o 1)
4. Provedeme jeden krok modifikované waveletové rekonstrukce podle S. Mallata:
 - (a) u všech koeficientů nadvzorkování po sloupcích
 - (b) konvoluce koeficientů s příslušným waveletovým rekonstrukčním filtrem po sloupcích
 - (c) nadvzorkování po řádcích
 - (d) konvoluce koeficientů s příslušným waveletovým rekonstrukčním filtrem po řádcích
 - (e) sečtení všech čtyřech výsledků dohromady
5. Opakujeme od kroku 2, dokud neprojdeme všechny úrovně dekompozice.

6. Upravíme rozměry segmentu na dvojnásobek rozměrů detailních koeficientů na první úrovni dekompozice plus $m - 1$ (v některých případech se rozměry liší o 1).
7. Vložíme rekonstruovaný segment na správné místo v obraze tak, že tam, kde dochází k překryvu segmentů, sečteme odpovídající hodnoty, viz obr. 3.3.
8. Opakujeme vše od bodu 1, dokud nedostaneme kompletní rekonstruovaný obraz.
9. Ořízneme obraz na všech stranách o $r(J)$ pixelů.

5 POPIS PROGRAMU

Na obrázku 5.1 vidíme grafické rozhraní programu **segdtwt.m**. V nabídce soubor lze načíst vstupní obraz, uložit zpracovaný soubor a ukončit program. Volíme parametry SegDTWT, což jsou délka segmentu, použitý wavelet, hloubku dekompozice a typ okrajového prodloužení. Poslední volbou je výběr algoritmu – buď s prodloužením okrajů před vlastní transformací anebo s prodloužením v každém kroku dekompozice.



Obr. 5.1: Grafické uživatelské rozhraní programu **segdtwt** po provedení SegDTWT a SegIDTWT na testovacím obraze

Program dále umožňuje použití tvrdého prahování waveletových koeficientů s nastavitelnou hodnotou prahu a volbou zda prahovat i aproximační koeficienty.

5.1 Ověření funkčnosti

Pro ověření správné funkčnosti algoritmu SegDTWT je výsledek porovnáván s běžnou DTWT v poli „odchylka waveletových koeficientů před prahováním“. Matice koeficientů SegDTWT a DTWT počítané funkcí MATLABu **wavedec2.m** jsou odečteny po prvcích a následně je vybráno v absolutní hodnotě nejvyšší číslo. Koeficienty jsou řazeny od maximální úrovně dekompozice v tomto pořadí: $A_J, H_J, V_J, D_J, H_{J-1}, V_{J-1}, D_{J-1} \dots H_1, D_1, V_1$. V případě správné funkčnosti, jsou vráceny pouze nulové hodnoty nebo velmi malá, zanedbatelná čísla.

Tab. 5.1: Ověření funkčnosti programu – odchylky waveletových koeficientů pro prodloužení před transformací i pro prodlužování v každém kroku dekompozice. Jsou použita čtyři základní prodloužení: zpd – prodloužení nulami, sp0 – prodloužení konstantou, sp1 – prodloužení polynomem prvního řádu, sym – půlbodové symetrické prodloužení.

prodloužení před transformací						
délka segmentu:256, hloubka dek.: 3, obraz: sea.bmp 1000x1000px						
prodloužení	wavelet	odchylka A_3	odchylka D_3	odchylka D_2	odchylka D_1	
zpd	db4	0	0	0	0	
	bior2.4	0	0	0	0	
	db8	0	0	0	0	
sp0	db4	58,6	38,55	27,15	0	
	bior2.4	12,56	6,74	3,337	0	
	db8	13,76	9,153	6,095	0	
sp1	db4	4484	944,9	237,8	$2,6 \cdot 10^{-13}$	
	bior2.4	319,6	106,5	21,78	$1,44 \cdot 10^{-13}$	
	db8	7234	495,1	92,55	$5,05 \cdot 10^{-13}$	
sym	db4	1163	730,8	280,1	0	
	bior2.4	1968	855,5	350	0	
	db8	1734	463	380,7	0	

prodloužení v každém kroku dekompozice						
délka segmentu:256, hloubka dek.: 3, obraz: sea.bmp 1000x1000px						
prodloužení	wavelet	odchylka A_3	odchylka D_3	odchylka D_2	odchylka D_1	
zpd	db4	0	0	0	0	
	bior2.4	0	0	0	0	
	db8	0	0	0	0	
sp0	db4	0	0	0	0	
	bior2.4	0	0	0	0	
	db8	0	0	0	0	
sp1	db4	$2,18 \cdot 10^{-11}$	$5,83 \cdot 10^{-12}$	$3,13 \cdot 10^{-13}$	$2,65 \cdot 10^{-13}$	
	bior2.4	$1,67 \cdot 10^{-10}$	$7,62 \cdot 10^{-12}$	$2,09 \cdot 10^{-12}$	$1,44 \cdot 10^{-13}$	
	db8	$6,69 \cdot 10^{-10}$	$5,42 \cdot 10^{-11}$	$4,83 \cdot 10^{-12}$	$5,05 \cdot 10^{-13}$	
sym	db4	0	0	0	0	
	bior2.4	0	0	0	0	
	db8	0	0	0	0	

Když ale použijeme úplné prodloužení před vlastní transformací, zjistíme, že se liší všechny koeficienty mimo těch v první úrovni dekompozice. Nejde ovšem o chybu, ale o vlastnost z důvodu jiného použitého algoritmu než používá MATLAB pro srovnání. Při prodloužení nulami jsou koeficienty identické při obou nastaveních, neboť není podstatné, zda přidáme na okraj nuly hned na začátku, nebo v průběhu transformace. Konvoluce delších nulových okrajů vrací opět samé nuly. U ostatních prodloužení tomu tak není, a proto se potom koeficienty liší. Dostáváme tedy jinou waveletovou reprezentaci, která ale po inverzní transformaci dosahuje opět perfektní rekonstrukce, tj. minimální odchylky originálního a zpracovaného obrazu.

5.2 Měření odchylek

Abychom mohli potvrdit funkčnost i SegIDTWT, je potřeba stanovit kritérium, podle kterého lze posuzovat rekonstruovaný obraz. Jako nejpoužívanější objektivní metoda se nabízí střední kvadratická odchylka – MSE a špičkový poměr signálu k šumu – PSNR. MSE je definována [7]

$$\text{MSE} = \frac{1}{M \cdot N} \sum_{i=1}^M \sum_{j=1}^N (\text{E}(i, j))^2, \quad (5.1)$$

kde M a N je výška resp. šířka obrazové matice a

$$\text{E}(i, j) = \text{K}(i, j) - \text{O}(i, j), \quad (5.2)$$

pro které platí, že $\text{K}(i, j)$ je komprimovaný obraz a $\text{O}(i, j)$ je originální obraz, přičemž i a j udávají aktuální souřadnice pixelu. Díky tomu že již známe MSE, lze vypočítat PSNR pomocí vztahu

$$\text{PSNR} = 10 \log \left(\frac{255^2}{\text{MSE}} \right), [\text{dB}] \quad (5.3)$$

přičemž číslo 255 udává počet možných úrovní signálu – v našem případě má černobílý vstupní i výstupní obraz maximálně 255 úrovní šedi.

Ze vztahů tudíž vyplývá, že čím více se rekonstruovaný obraz liší od originálu, tím je vyšší i MSE a zároveň klesá PSNR. V programu `segdtwt.m` jsou používána obě kritéria měření kvality. Na celý obraz jsou použity vzorce 5.1 a 5.3, pokud ale chceme počítat MSE a PSNR pouze na okrajích signálu, které mohou být ovlivněny různým druhem prodloužení, musíme vzorec MSE upravit následovně. Nejprve sečteme kvadráty všech chybových prvků $\text{E}(i, j)$, ale odečíst musíme střední část tj. všechny kvadráty chybových prvků, které mají vzdálenost od některého z okrajů větší než $r(J)$, jak ukazuje vzorec

$$E_o = \left(\sum_{i=1}^M \sum_{j=1}^N (E(i, j))^2 - \sum_{i=r(J)+1}^{M-r(J)} \sum_{j=r(J)+1}^{N-r(J)} (E(i, j))^2 \right). \quad (5.4)$$

Následně dělíme počtem prvků na okrajích a dostáváme

$$\text{MSE}_o = \frac{E_o}{M \cdot N - (M - 2r(J))(N - 2r(J))}. \quad (5.5)$$

Vzhledem k tomu, že při výpočtu PSNR používáme MSE, je vzorec pro PSNR na okrajích stejný jako 5.3, pouze za MSE dosadíme MSE na okrajích.

5.3 Nadbytečnost

Výhodou segmentace je zejména možnost zpracování každého bloku zvlášť na jiném procesoru či procesorovém jádře. Čím více paralelně pracujících výpočetních jednotek, tím rychleji lze dojít k výsledku. Ovšem z popisu algoritmu je zřejmé, že zpracování po segmentech produkuje nadbytečné výpočty. Za nadbytečné považujeme pixely, se kterými musíme počítat více než jednou, neuvažujeme ale prodloužení na okrajích, neboť to je nutné provádět i při běžné DTWT. Nadbytečnost výpočtu tak tvoří prodloužování segmentů vzorky sousedních bloků. Je tedy třeba zpracovat

$$(r(J) \cdot (c - 1) + M)^2 \quad (5.6)$$

pixelů po prodloužení vnitřních segmentů, kde c značí počet segmentů v řádku či sloupci a M je rozměr obrazu (uvážujeme čtvercové obrazy a segmenty). Počet nadbytečných pixelů je tedy

$$(r(J) \cdot (c - 1) + M)^2 - M^2. \quad (5.7)$$

Při každém výpočtu SegDTWT počítá vytvořený program nadbytečnost v procentech podle vzorce

$$\text{Red} = \frac{((r(J) \cdot (c - 1) + M)^2 - M^2) \cdot 100}{(r(J) \cdot (c - 1) + M)^2} \cdot [\%] \quad (5.8)$$

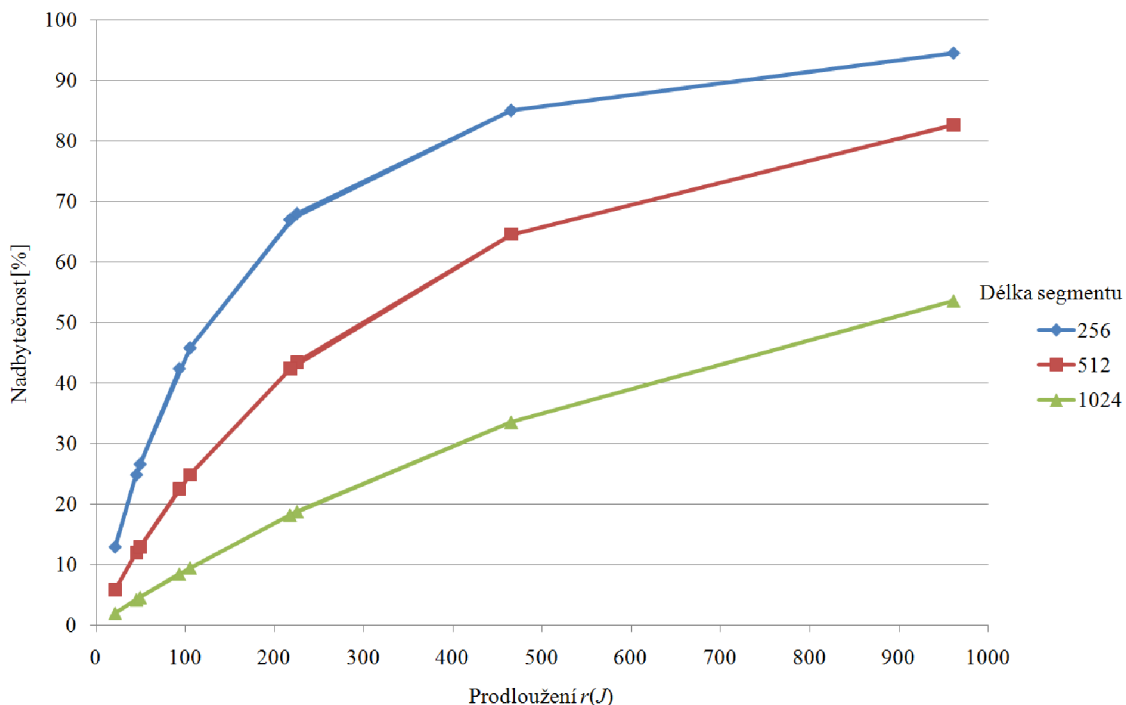
Závislost nadbytečnosti výpočtu SegDTWT na maximálním prodloužení $r(J)$ ukazuje tabulka 5.2 a graf na obr 5.2. Je z nich jasně patrné, že s přibývajícím délkou filtru a zvyšující se hloubkou dekompozice rychle narůstá nadbytečnost. Naopak při zvětšení délky segmentů redundance klesá. Obecně pro nízkou nadbytečnost výpočtu je třeba volit co největší segmenty a menší hloubku dekompozice.

Tab. 5.2: Závislosti nadbytečnosti výpočtu SegDTWT na délce maximálního prodloužení pro segmenty délky 256, 512 a 1024 vzorků

rozměr obrazu 2048				rozměr segmentu 256								
délka filtru	4			8			16			32		
hloubka dekomp.	3	4	5	3	4	5	3	4	5	3	4	5
prodloužení $r(J)$	21	45	93	49	105	217	105	225	465	217	465	961
nadbytečnost[%]	13	25	42	27	46	67	46	68	85	67	85	95

rozměr obrazu 2048				rozměr segmentu 512								
délka filtru	4			8			16			32		
hloubka dekomp.	3	4	5	3	4	5	3	4	5	3	4	5
prodloužení $r(J)$	21	45	93	49	105	217	105	225	465	217	465	961
nadbytečnost[%]	6	12	23	13	25	42	25	43	65	42	65	83

rozměr obrazu 2048				rozměr segmentu 1024								
délka filtru	4			8			16			32		
hloubka dekom	3	4	5	3	4	5	3	4	5	3	4	5
prodloužení $r(J)$	21	45	93	49	105	217	105	225	465	217	465	961
nadbytečnost[%]	2	4	9	5	10	18	10	19	34	18	34	54



Obr. 5.2: Graf závislosti nadbytečnosti výpočtu SegDTWT na maximální délce prodloužení

Při využití v praxi by bylo vhodné vzít ještě úvahu, kolik výpočetních jednotek je k dispozici, a tudíž kolik segmentů zároveň lze zpracovávat. Například při současném výpočtu 4 segmentů a 50% redundanci, dojdeme ke dvojnásobnému zrychlení kalkulace oproti standartní DTWT. To ovšem platí pouze v případě, že neuvažujeme režii, kterou reálný paralelizovaný program má. Zrychlení bylo pak bylo menší.

Je také otázkou, zda menší počet rozměrnějších segmentů nebude mít příliš velké paměťové nároky při výpočtu a nebude lepší zvýšit počet redundantních výpočtů ve prospěch objemu využívaných paměťových prostředků. Proto je třeba u každé konkrétní aplikace pečlivě zvážit vhodnou volbu rozměrů segmentu, aby bylo dosaženo co nejvyšší možné efektivity.

6 VLIV TYPU PRODLOUŽENÍ

V programu jsou použity čtyři druhy prodloužování na okrajích a dva různé přístupy k jejich zařazení do algoritmů. Aby bylo možné zhodnotit jejich vliv, byly vybrány tři testovací obrazy, každý s různou scénou a rozměry, u kterých je počítáno MSE a PSNR na okrajích (tj. na všech pixelech do vzdálenosti $r(J)$ od kteréhokoli kraje) podle vzorců 5.5 a 5.3. Tyto parametry slouží k porovnání úspěšnosti při rekonstrukci, protože, jak bylo ukázáno v kapitole 5, waveletové koeficienty se po zpracování po segmentech téměř neliší.

Podrobné nastavení programu při testování je různé pro každý z obrazů. Pro lepší možnost zhodnocení bylo v polovině měření použito prahování s nastaveným prahem 1. Program využívá tzv. tvrdé prahování, což znamená, že každý koeficient, který je v absolutní hodnotě menší než prahová hodnota, je vynulován. Volba prahovat aproximační koeficienty nebyla zaškrtnuta, jelikož z několika namátkových testů vyplynulo, že použití této volby nemá na výsledné hodnoty MSE a PSNR téměř žádný vliv.

Při volbě délky segmentu bylo přihlíženo k tomu, aby nebyl rozměr obrazu dělitelný právě délkou segmentu a aby se objevila nejen situace, kdy pravé a dolní prodloužení je nulové, ale i případ jejich využití.

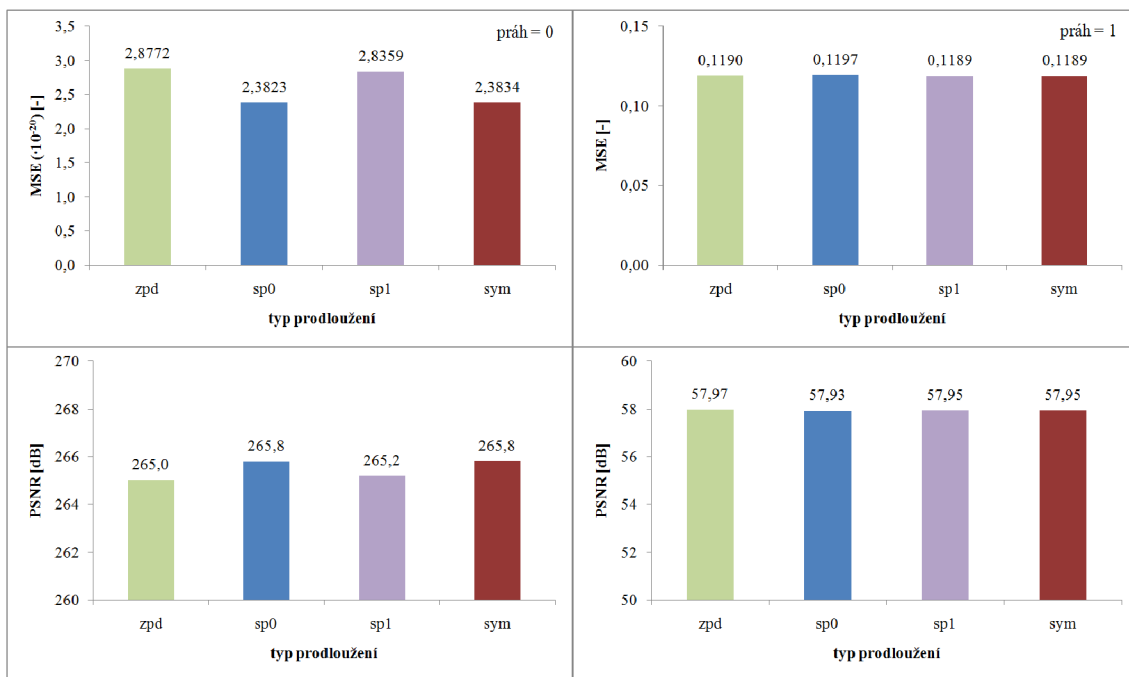
Přestože známe dva druhy symetrického prodloužení, je využita pouze možnost půlbodového symetrického prodloužení. Důvodem je fakt, že program MATLAB upravuje délku všech waveletových filtrů na sudou hodnotu a pro sudé filtry je vhodné použití půlbodového prodloužování.

Všechny testy zobrazují tabulky 6.1, 6.2 a 6.3, kde si všimněme především hodnot ΔMSE a ΔPSNR . Jde o příslušné rozdíly v absolutní hodnotě mezi přístupy k prodloužování. Z uvedených tabulek zjevně vyplývá, že přestože se waveletová reprezentace v některých případech liší, nemá toto vliv na rekonstrukci. Podle provedených měření nelze vyzdvihnout ani jeden z přístupů, protože oba produkují obrazy s jen zanedbatelnými odchylkami.

Výběr tedy závisí na tom, pro jakou aplikaci je který přístup snadněji realizovatelný. Z hlediska výpočetní náročnosti sice prodloužení před vlastní transformací vnáší určitou nadbytečnost tím, že dochází ke konvoluci s filtrem na vyšším počtu vzorků, které jsou poté u některých koeficientů stejně ořezány, ale na druhou stranu nevyžaduje v každém kroku waveletové transformace extrapolaci signálu.

Na obrázku 6.1 jsou znázorněny grafy průměrných hodnot MSE a PSNR z předchozích tabulek, aby bylo možné určit prodloužení, které nejméně ovlivňuje výsledný obraz. Především je třeba říci, že rozdíly mezi použitými prodlouženími jsou velmi

malé. Ale i přesto vychází symetrické prodloužení a prodloužení konstantou jako mírně lepší než prodloužení polynomem prvního řádu nebo jednoduché prodloužení nulami. V případě prahování koeficientů se rozdíly ještě zmenšují.



Obr. 6.1: Grafy průměrné MSE a PSNR v závislosti na typu prodloužení. Levá část obsahuje průměrné hodnoty pro zpracování obrazu bez prahování koeficientů, pravá po nastavení prahu na hodnotu 1.

Tab. 6.1: Hodnoty MSE a PSNR pro různé druhy prodloužení (zpd – prodloužení nulami, sp0 – prodloužení konstantou, sp1 – prodloužení polynomem prvního řádu, sym – půlbodové symetrické prodloužení). Je aplikováno jak úplné prodloužení před vlastní transformací, tak prodloužení v každém kroku dekompozice. Hodnoty Δ MSE a Δ PSNR jsou příslušné rozdíly v absolutní hodnotě.

délka segmentu: 256, hloubka dek.: 3, obraz: sea.bmp 1000x1000px, práh: 0							
wavel.	prodl.	prodl. před tr.		prodl. během tr.		Δ MSE	Δ PSNR
		MSE	PSNR	MSE	PSNR		
db4	zpd	$1,1 \cdot 10^{-20}$	247,9	$1,1 \cdot 10^{-20}$	247,9	0	0
	sp0	$5,8 \cdot 10^{-21}$	250,5	$5,7 \cdot 10^{-21}$	250,5	$1 \cdot 10^{-22}$	0
	sp1	$7,9 \cdot 10^{-21}$	249,2	$7,3 \cdot 10^{-21}$	249,5	$6 \cdot 10^{-22}$	0,3
	sym	$5,7 \cdot 10^{-21}$	250,5	$5,7 \cdot 10^{-21}$	250,6	0	0
bior3.5	zpd	$1,5 \cdot 10^{-26}$	306,4	$1,4 \cdot 10^{-26}$	306,4	0	0
	sp0	$1,5 \cdot 10^{-26}$	306,4	$1,5 \cdot 10^{-26}$	306,4	0	0
	sp1	$1,5 \cdot 10^{-26}$	306,4	$1,5 \cdot 10^{-26}$	306,3	0	0
	sym	$1,5 \cdot 10^{-26}$	306,3	$1,5 \cdot 10^{-26}$	306,6	0	0,2
db8	zpd	$7,9 \cdot 10^{-20}$	239,1	$7,9 \cdot 10^{-20}$	239,1	0	0
	sp0	$6,3 \cdot 10^{-20}$	240,2	$6,3 \cdot 10^{-20}$	240,2	0	0
	sp1	$7,3 \cdot 10^{-20}$	239,5	$7,2 \cdot 10^{-20}$	239,6	$1 \cdot 10^{-21}$	0,1
	sym	$6,3 \cdot 10^{-20}$	240,2	$6,3 \cdot 10^{-20}$	240,2	0	0

délka segmentu: 256, hloubka dek.: 3, obraz: sea.bmp 1000x1000px, práh: 1							
wavel.	prodl.	prodl. před tr.		prodl. během tr.		Δ MSE	Δ PSNR
		MSE	PSNR	MSE	PSNR		
db4	zpd	0,089	58,59	0,089	58,59	0	0
	sp0	0,092	58,46	0,092	58,47	0	0,1
	sp1	0,091	58,53	0,091	58,54	0	0,1
	sym	0,092	58,48	0,092	58,49	0	0,1
bior3.5	zpd	0,236	54,39	0,236	54,39	0	0
	sp0	0,235	54,42	0,235	54,42	0	0
	sp1	0,233	54,45	0,233	54,46	0	0,1
	sym	0,232	54,47	0,232	54,47	0	0
db8	zpd	0,082	58,98	0,082	58,98	0	0
	sp0	0,084	58,88	0,084	58,89	0	0,1
	sp1	0,084	58,88	0,084	58,88	0	0
	sym	0,084	58,93	0,083	58,92	0	0,1

Tab. 6.2: Hodnoty MSE a PSNR pro různé druhy prodloužení (zpd – prodloužení nulami, sp0 – prodloužení konstantou, sp1 – prodloužení polynomem prvního řádu, sym – půlbodové symetrické prodloužení). Je aplikováno jak úplné prodloužení před vlastní transformací, tak prodloužení v každém kroku dekompozice. Hodnoty Δ MSE a Δ PSNR jsou příslušné rozdíly v absolutní hodnotě.

délka segmentu: 270, hloubka dek.: 3, obraz: bridge.bmp 1375x1375px, práh: 0							
wavel.	prodl.	prodl. před tr.		prodl. během tr.		Δ MSE	Δ PSNR
		MSE	PSNR	MSE	PSNR		
db4	zpd	$8,8 \cdot 10^{-21}$	248,7	$8,8 \cdot 10^{-21}$	248,7	0	0
	sp0	$6,9 \cdot 10^{-21}$	249,7	$6,9 \cdot 10^{-21}$	249,7	0	0
	sp1	$8,1 \cdot 10^{-21}$	249,1	$7,9 \cdot 10^{-21}$	249,2	$2 \cdot 10^{-22}$	0,1
	sym	$7 \cdot 10^{-21}$	249,7	$7 \cdot 10^{-21}$	249,7	0	0
bior3.5	zpd	$5,1 \cdot 10^{-27}$	311,1	$5,1 \cdot 10^{-27}$	311,1	0	0
	sp0	$5,1 \cdot 10^{-27}$	311,1	$5,5 \cdot 10^{-27}$	311,1	$4 \cdot 10^{-28}$	0
	sp1	$5 \cdot 10^{-27}$	311,1	$5 \cdot 10^{-27}$	311,1	0	0
	sym	$5 \cdot 10^{-27}$	311,1	$5 \cdot 10^{-27}$	311,1	0	0
db8	zpd	$5,4 \cdot 10^{-20}$	240,8	$5,4 \cdot 10^{-20}$	240,8	0	0
	sp0	$4,8 \cdot 10^{-20}$	241,3	$4,8 \cdot 10^{-20}$	241,3	0	0
	sp1	$5,3 \cdot 10^{-20}$	240,9	$5,3 \cdot 10^{-20}$	240,9	0	0
	sym	$4,8 \cdot 10^{-20}$	241,3	$4,8 \cdot 10^{-20}$	241,3	0	0

délka segmentu: 270, hloubka dek.: 3, obraz: bridge.bmp 1375x1375px, práh: 1							
wavel.	prodl.	prodl. před tr.		prodl. během tr.		Δ MSE	Δ PSNR
		MSE	PSNR	MSE	PSNR		
db4	zpd	0,056	60,68	0,056	60,68	0	0
	sp0	0,057	60,57	0,057	60,57	0	0
	sp1	0,056	60,62	0,056	60,63	0	0,01
	sym	0,057	60,58	0,057	60,57	0	0,01
bior3.5	zpd	0,158	56,15	0,158	56,15	0	0
	sp0	0,158	56,14	0,158	56,14	0	0
	sp1	0,157	56,16	0,157	56,16	0	0
	sym	0,157	56,16	0,157	56,16	0	0
db8	zpd	0,054	60,82	0,054	60,82	0	0
	sp0	0,054	60,77	0,054	60,77	0	0
	sp1	0,054	60,76	0,054	60,76	0	0
	sym	0,054	60,83	0,054	60,83	0	0

Tab. 6.3: Hodnoty MSE a PSNR pro různé druhy prodloužení (zpd – prodloužení nulami, sp0 – prodloužení konstantou, sp1 – prodloužení polynomem prvního řádu, sym – půlbodové symetrické prodloužení). Je aplikováno jak úplné prodloužení před vlastní transformací, tak prodloužení v každém kroku dekompozice. Hodnoty Δ MSE a Δ PSNR jsou příslušné rozdíly v absolutní hodnotě.

délka segmentu: 540, hloubka dek.: 4, obraz: building.bmp 2706x2706px, práh: 0							
wavel.	prodl.	prodl. před tr.		prodl. během tr.		Δ MSE	Δ PSNR
		MSE	PSNR	MSE	PSNR		
db4	zpd	$1,4 \cdot 10^{-20}$	246,6	$1,4 \cdot 10^{-20}$	246,6	0	0
	sp0	$1,1 \cdot 10^{-20}$	247,9	$1,1 \cdot 10^{-20}$	247,9	0	0
	sp1	$1,5 \cdot 10^{-20}$	246,3	$1,4 \cdot 10^{-20}$	246,8	$1 \cdot 10^{-21}$	0,5
	sym	$1,1 \cdot 10^{-20}$	247,9	$1,1 \cdot 10^{-20}$	247,9	0	0
bior3.5	zpd	$1,6 \cdot 10^{-26}$	306,1	$1,6 \cdot 10^{-26}$	306,1	0	0
	sp0	$1,6 \cdot 10^{-26}$	306	$1,6 \cdot 10^{-26}$	306	0	0
	sp1	$1,6 \cdot 10^{-26}$	306	$1,6 \cdot 10^{-26}$	306	0	0
	sym	$1,6 \cdot 10^{-26}$	306	$1,6 \cdot 10^{-26}$	306	0	0
db8	zpd	$9,2 \cdot 10^{-20}$	238,5	$9,2 \cdot 10^{-20}$	238,5	0	0
	sp0	$8 \cdot 10^{-20}$	239,1	$8 \cdot 10^{-20}$	239,1	0	0
	sp1	$1 \cdot 10^{-19}$	238	$9,9 \cdot 10^{-20}$	238,2	$1 \cdot 10^{-21}$	0,2
	sym	$8 \cdot 10^{-20}$	239,1	$8 \cdot 10^{-20}$	239,1	0	0

délka segmentu: 540, hloubka dek.: 4, obraz: building.bmp 2706x2706px, práh: 1							
wavel.	prodl.	prodl. před tr.		prodl. během tr.		Δ MSE	Δ PSNR
		MSE	PSNR	MSE	PSNR		
db4	zpd	0,084	58,90	0,084	58,90	0	0
	sp0	0,085	58,90	0,085	58,90	0	0
	sp1	0,084	58,88	0,084	58,88	0	0
	sym	0,084	58,87	0,084	58,87	0	0
bior3.5	zpd	0,221	54,69	0,221	54,69	0	0
	sp0	0,221	54,69	0,221	54,69	0	0
	sp1	0,220	54,71	0,220	54,71	0	0
	sym	0,220	54,72	0,220	54,72	0	0
db8	zpd	0,091	58,56	0,091	58,56	0	0
	sp0	0,091	58,53	0,091	58,53	0	0
	sp1	0,091	58,53	0,091	58,53	0	0
	sym	0,091	58,55	0,091	58,54	0	0,1

7 ZÁVĚR

V diplomové práci segmentovaná vlnková transformace obrazu byla rozebrána teorie waveletové transformace, jejího výpočtu pro diskrétní signály včetně podrobného popisu algoritmu pro zpracování dvojrozměrných obrazových dat. Bylo ukázáno, že waveletovou transformaci lze rozdělit na segmenty, které mohou být nezávisle zpracovávány, aniž by se výsledné koeficienty lišily od případu, kdy je zpracováván celý obraz najednou. Dále byla popsána metoda výpočtu inverzní dvojrozměrné waveletové transformace a její úprava na výpočet po nezávislých segmentech.

Jako součást diplomové práce je přiložen program `segdtwt.m`, který prakticky ověřuje funkčnost algoritmů SegDTWT a SegIDTWT. Ty byly implementovány tak, aby dokázaly zpracovat čtvercový obraz o jakýchkoli rozměrech a libovolné délce segmentu. Podle zvoleného rozměru segmentu jsou vypočteny délky levého, horního, pravého a dolního prodloužení pro konkrétní segment, o něž je pak blok prodloužen vzorky sousedních segmentů. Program využívá grafického uživatelského rozhraní, kde lze volit všechny parametry, a díky podobnému značení vnitřních funkcí algoritmu jako ve Wavelet Toolboxu MATLABu by neměl být problém s jeho případným rozšířením.

Práce obsahuje také popis dvou různých přístupů k prodloužení signálu na jeho okrajích: úplné prodloužení pro všechny úrovně dekompozice před vlastní transformací a klasický přístup, kdy je signál prodlužován během transformace v každém jejím kroku. Prodlužovat signál na okrajích lze metodami nastudovanými a popsanými v kapitole 2.

Výsledky ukazují, že při použití klasického přístupu k prodlužování, produkuje algoritmus SegDTWT waveletové koeficienty pouze se zanedbatelnou, většinou však nulovou odchylkou od koeficientů počítaných pomocí DTWT. Po inverzní transformaci je střední kvadratická odchylka originálního a zpracovaného obrazu většinou v řádech 10^{-20} a PSNR se pohybuje okolo 265 dB, což značí opravdu minimální rozdíly. Pokud využijeme úplné prodloužení okrajů signálu před vlastní transformací, dostáváme sice odlišné některé waveletové koeficienty, ale po SegIDTWT jsou odchylky rekonstruovaného obrazu od originálu ve stejných řádech, jako u předchozího přístupu. Nelze tedy jednoznačně určit, která metoda je lepší, neboť výsledky jsou téměř shodné. Záleží, která implementace by byla pro konkrétní účel výhodnější.

Při porovnávání metod prodlužování okrajů, jejichž výsledky lze najít v tabulkách 6.1, 6.2 a 6.3, jsme použili výpočet MSE a PSNR pouze na okraji obrazu. Docházíme k závěru, že všechny čtyři testované druhy prodloužení dávají podobné výsledky, přesto ale lze favorizovat prodloužení konstantou a symetrické prodloužení, která dosáhla nižší MSE, proti prodloužení polynomem prvního řádu a prodloužení nulami.

Část diplomové práce je věnována výpočtu nadbytečnosti algoritmu SegDTWT oproti klasickému postupu. Redundance stoupá především s narůstající hloubkou dekompozice a délkou filtru, naopak se zvýšením rozměru segmentu nadbytečnost klesá. K tomu dochází důvodu, že pracujeme s menším počtem segmentů, které je třeba prodlužovat, a právě prodloužení sousedními vzorky je zde zdrojem výpočtů navíc. Až konkrétní aplikace může říci, zda zvolit krátké segmenty, které budou mít pravděpodobně menší paměťové nároky a vyšší nadbytečnost samotných výpočtů nebo naopak velké segmenty s nízkou procentuální redundancí a vyššími nároky na paměť.

Výhodou zpracování waveletové transformace po segmentech je také možnost neskládat po výpočtu jednoho bloku celkové matice koeficientů, ale koeficienty rovnou upravit a poslat na vstup inverzní transformaci. Pak můžeme nezávisle zpracovávat bloky obrazu, ať už jde o odšumování, kompresi nebo kterékoliv jiné využití waveletové transformace.

LITERATURA

- [1] HORA, P. *Vlastnosti spojité waveletové transformace* [online]. 2000. Dostupné z URL: <http://www.cdm.cas.cz/publications/hora/ph_turb2000.pdf>.
- [2] JAN, J. *Číslíková filtrace, analýza a restaurace signálů*. druhé upravené a rozšířené vydání. Brno: VUTIUM, 2002. 427s. ISBN 80-214-2911-9.
- [3] KYSELÝ, F. *Metody maskování chyb v prostředí Matlab*. Diplomová práce, Vysoké učení technické v Brně, Brno, 2007.
- [4] MALLAT, S. *A Wavelet Tour of Signal Processing*. 2nd edition, Academic Press, 1999. ISBN: 0-12-466606-X.
- [5] MALÝ, J. *Metoda pro kompresi obrazových signálů pomocí waveletové transformace*. 85s. Diplomová práce, Vysoké učení technické v Brně, Brno.
- [6] MATLAB, oficiální stránky [online]. c1994-2009, [cit. 2010-05-06]. Dostupné z URL: <<http://www.mathworks.com/products/matlab>>.
- [7] PRŮŠA, Z. *Efektivní nástroj pro kompresi obrazu v jazyce java*. Diplomová práce, Vysoké učení technické v Brně, Brno, 2008.
- [8] RAJMIC, P. *Využití waveletové transformace a matematické statistiky pro separaci signálu a šumu*. Doktorská dizertační práce, Vysoké učení technické v Brně, Brno, 2004.
- [9] RAJMIC, P. *Algorithms for Segmentwise Computation of Forward and Inverse Discrete-time Wavelet Transform*. Journal of Concrete and Applicable Mathematics, Special Issues on Applied Mathematics and Approximation Theory, Vol. 8, No. 3, s. 393-406. Eudoxuspress, Memphis, TN USA, 2010. ISSN 1548-5390.
- [10] RAJMIC, P.; MALÝ, J. *Boundary Effects in the Wavelet Transform of Finite Discrete Signals*. In Proceedings of the 30th International Conference Telecommunications and Signal Processing TSP-2007. Brno: VUT v Brně, 2007. s. 81-84. ISBN: 978-80-214-3445-5.
- [11] RAJMIC, P.; VLACH, J. *Real-time Audio Processing via Segmented Wavelet Transform*. In Proceedings of the 10th International Conference on Digital Audio Effects (DAFx-07). Bordeaux, University of Bordeaux 1. 2007. pp. 55-58. ISBN 978-88-901479-1-3.

- [12] SMÉKAL, Z. *Číslíkové zpracování signálů*. Skripta, Vysoké učení technické v Brně, Brno, aktualizace 2009, 202s.
- [13] STRANG G.; NGUYEN T. *Wavelets and Filter Banks*. 2nd edition, Wellesley College, 1996. ISBN: 978-0961408879.
- [14] VALENS, C. *A Really Friendly Guide to Wavelets*., [online] 1999-2004, poslední aktualizace 26.02.2004 [cit. 2010-04-20] Dostupné z URL: <<http://pagesperso-orange.fr/polyvalens/clemens/wavelets/wavelets.html>>.
- [15] VARGIC, R. *Wavelety a banky filtrov*. Skripta, Slovenská technická univerzita, Bratislava. 2003, 135s

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CWT	Continuous Wavelet Transform, spojitá waveletová transformace
DP	Dolní propust
DTWT	Discrete Time Wavelet Transform, diskrétní waveletová transformace s diskrétním časem
DWT	Discrete Wavelet Transform, diskrétní waveletová transformace
f_{vz}	Vzorkovací frekvence
GUI	Graphical user interface, grafické uživatelské rozhraní
HP	Horní propust
JPEG	Joint Photographic Experts Group, obrazový formát se ztrátovou kompresí
JPEG2000	Joint Photographic Experts Group 2000, obrazový formát založený na waveletové transformaci
MSE	Mean Square Error, střední kvadratická odchylka
PSNR	Peak Signal-To-Noise Ratio, špičkový poměr signálu k šumu
SegDTWT	Segmented Discrete Time Wavelet Transform, segmentovaná waveletová transformace s diskrétním časem
SegIDTWT	Segmented Inverse Discrete Time Wavelet Transform, inverzní segmentovaná waveletová transformace s diskrétním časem

SEZNAM PŘÍLOH

A Obsah přiloženého CD

51

A OBSAH PŘILOŽENÉHO CD

1. **adresář MATLAB** – Program **segdwt.m** a včetně testovacích obrázků.
Pro spuštění GUI: **segdwt.m**, hlavní algoritmy: **segmentation.m**
2. **xkucer57.pdf** – Elektronická verze diplomové práce
3. **readme.txt** – Obsah CD